The estimation and use of 3D information, for natural human action recognition

Simon Hadfield

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

May 2013

Abstract

The aim of this thesis, is to develop estimation and encoding techniques for 3D information, which are applicable in a range of vision tasks. Particular emphasis is given to the task of natural action recognition. This "in the wild" recognition, favours algorithms with broad generalisation capabilities, as no constraints are placed on either the actor, or the setting. This leads to huge intra-class variability, including changes in lighting, actor appearance, viewpoint and action style. Algorithms which perform well under these circumstances, are generally well suited for real world deployment, in applications such as surveillance, video indexing, and assisted living. The issues of generalisation, may be mitigated to a significant extent, by utilising 3D information, which provides invariance to most appearance based variation. In addition, 3D information can remove projective distortions and the effect of camera orientation, and provides cues for occlusion. The exploitation of these properties has become feasible in recent years. This is due to both the emergence of affordable 3D sensors, such as the Microsoft Kinect$^{\text{TM}}$, and the ongoing growth of 3D broadcast footage (including 3D TV channels, and 3D Blu-Ray).

To evaluate the impact of this 3D information, and provide a benchmark to aid future development, a large multi-view action dataset is compiled, covering 14 different action classes and comprising over an hour of high definition video. This data is obtained from 3D broadcast footage, which provides a broader range of variations, than may be feasibly produced, during staged capture in the lab. A large number of existing action recognition techniques are then implemented, and extensions formulated, to allow the encoding of 3D structural information. This leads to significantly improved generalisation, over standard spatiotemporal techniques.

As an additional information source, the estimation of 3D motion fields is also developed. Motion estimation in 3D is also referred to as "scene flow", to contrast with its image plane counterpart "Optical Flow". However, previous work on scene flow estimation, has been unsuitable for real applications, due to the computational complexity of approaches proposed in the literature. The previous state of the art techniques generally require several hours, to estimate the motion of a single frame, rendering their use with datasets of reasonable size, intractable. This in turn, has lead to the field of scene flow estimation being often viewed as an item of academic interest only. In this thesis, a new monte-carlo based approach to motion estimation is proposed, which is not only several orders of magnitude faster (and amenable to a parallelised GPU implementation), but also provides improved accuracy by avoiding over-smoothing artefacts.

The value of this particle based approach is further augmented, by a re-examination of the underlying assumptions in motion estimation theory. A deeper understanding of the behaviour of such systems is developed, for both the optical flow and scene flow estimation scenarios. In particular, existing formulations are demonstrated to either require accurate initialisation, or to favour small motions (explaining the popularity of multi-scale estimation schemes). The most enlightening analysis, however, explores the idea that functions which accurately represent real data, are not by default, suitable for the detection of estimation errors. This leads to the proposal of a more robust, non-linear estimation scheme, based on machine learning. This "Intelligent Transfer Function"

is incorporated into existing motion estimation schemes (both single and multi-view), along with support for probabilistic occlusion handling, and multi-hypothesis motion smoothing techniques.

Finally, this fast and accurate approach to 3D motion estimation, is exploited within the original task of natural action recognition. A range of schemes are explored, for effectively encoding this rich information, based on variations of the hugely popular Histogram of Oriented Gradients (HOG) and Histogram of Oriented Flow (HOF) descriptors. By utilising the actors undistorted motion field, action recognition rates are significantly improved over both the standard spatiotemporal approaches, and their 3D structural extensions. This serves to demonstrate that, due to the new more tractable formulation, in conjunction with the growth of 3D data, scene flow estimation may be a valuable tool for computer vision in the future.

Key words: Scene Flow, 3D Motion, Motion Estimation, Unregularized, Action Recognition, Interest Point, Saliency, Hollywood, Particle, 3D, Hand Tracking, Sign Language, Tracking, Scene Particles, Occlusion Estimation, Probabilistic Occlusion, Occlusion, Bilateral Filter, Histogram of Scene-flow, HOS, Motion Features, 3D Tracking, Motion Segmentation

Email: S.Hadfield@surrey.ac.uk

WWW: `http://personal.ee.surrey.ac.uk/Personal/S.Hadfield/`

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Richard Bowden. The contents of this thesis would be far less exciting without his brilliant insights, and extraordinary ability to find connections between seemingly disparate topics.

In addition I must acknowledge the many members of CVSSP at the university of Surrey, whose discussions helped to clarify in my mind both the goals, and the contributions of the work. In particular Dr. Nicolas Pugeault and Dr. Andrew Gilbert, who provided useful feedback on this thesis.

I am also grateful to my family, whose support throughout the years allowed me to reach this point without either getting a real job, or starving.

But most of all I am indebted to my fiancée Dina, who was tolerant beyond the call of duty in the face of late nights poring over code. Her love helped motivate me to always try my best.

Contents

Nomenclature

HCI	Human Computer Interaction
LBP	Local Binary Pattern
PTAMM	Parallel Tracking and Multiple Mapping
RDF	Randomised Decision Forest
HMM	Hidden Markov Model
CRF	Conditional Random Field
DTW	Dynamic Time Warping
SLAM	Simultaneous Location and Modelling
HOG	Histogram of Oriented Gradients
HOF	Histogram of Oriented Flow
HOOF	Histogram of Oriented Optical Flow
HOS	Histogram of Oriented Scene-flow
HODG	Histogram of Oriented Depth Gradients
BOW	Bag of Words
FAR	False Acceptance Rate
FRR	False Rejection Rate
AAM	Active Appearance Model
PCA	Principal Component Analysis
NRMS	Normalised Root Mean Square
AAE	Average Angular Error
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Feature

SfM	Structure from Motion
MRF	Markov Random Field
SFE	Scene Flow Estimation
DoF	Degree of Freedom
ASM	Active Shape Model
MIL	Multiple Instance Learning
KLT	Kanade-Lucas-Tomasi
SLR	Sign Language Recognition
GMM	Gaussian Mixture Model
SVM	Support Vector Machine
KF	Kalman Filter
PSO	Particle Swarm Optimisation
DCBG	Dual Cross Bilateral Grid
RMD	Relative Motion Descriptor
BoVW	Bag of Visual Words
SSD	Sum of Squared Differences
SAD	Sum of Absolute Differences
RBF	Radial Basis Function
PDF	Probability Density Function
FFBS	Filter Forward Backward Smoothing
ITF	Intelligent Transfer Function

Symbols

Introduced in chapter 3

$\boldsymbol{I}$	Input observations
$\boldsymbol{I}_a^s$	Observations from appearance sensors, at scale s
$\boldsymbol{I}_d^s$	Observations from depth sensors, at scale s
$\boldsymbol{I}_{\bigtriangledown x}, \boldsymbol{I}_{\bigtriangledown y}, \boldsymbol{I}_z, \boldsymbol{I}_{\bigtriangledown \tau}$	The derivative images, along dimension x,y,z and τ respectively
x, y	Spatial position within a 2D image
x', y'	Spatial position within a 2D image
t	Temporal position (frame number) in a sequence
τ	Temporal position (frame number) in a sequence
S	Number of image scales
s	Index for image scales
σ	The variance of a Gaussian
μ	The mean of a Gaussian
$g\left(\mu, \sigma\right)$	A Gaussian function with mean μ and variance σ
$g\left(x; \mu, \sigma\right)$	The output from a Gaussian function, for input x, with mean μ and variance σ
$\boldsymbol{H}_s$	The second moment matrix of a volume
$\boldsymbol{H}_{e-4D}$	The Hessian matrix of a volume
k	A constant, used when estimating Harris corners
$F_{4D\text{-}Ha}$	The set of 4D Harris interest points
$F_{4D\text{-}He}$	The set of 4D Hessian interest points
$F_{3.5D\text{-}Ha}$	The set of 3.5D Harris interest points

$F_{3.5D\text{-}He}$	The set of 3.5D Hessian interest points
$F_{3.5D\text{-}S}$	The set of 3.5D Separable Filter points
$F_{3D\text{-}Ha}$	The set of 3D Harris interest points
$F_{3D\text{-}S}$	The set of 3D Separable Filter points
$\lambda_{4D\text{-}Ha}$	The threshold for 4D Harris interest points
$\lambda_{4D\text{-}He}$	The threshold for 4D Hessian interest points
$\lambda_{3.5D\text{-}Ha}$	The threshold for 3.5D Harris interest points
$\lambda_{3.5D\text{-}He}$	The threshold for 3.5D Hessian interest points
$\lambda_{3.5D\text{-}S}$	The threshold for 3.5D Separable Filter points
$\lambda_{3D\text{-}Ha}$	The threshold for 3D Harris interest points
$\lambda_{3D\text{-}S}$	The threshold for 3D Separable Filter points
α	Weighting of depth versus structure information during 3.5D interest point estimation
g_{odd}	The odd half of a quadrature pair of gabor filters
g_{eve}	The even half of a quadrature pair of gabor filters
O	The number of saliency content comparisons
$\boldsymbol{I}_{Ha}$	The response volume of the Harris operator
$\boldsymbol{I}_{SF}$	The response volume of the Separable Filter operator
$\boldsymbol{I}_{int}$	The integral saliency volume created using interest point detection strengths
$\boldsymbol{I}_{rmd}$	The response volume for the RMD operator
$\boldsymbol{I}_{int\text{-}4d}$	The 4D integral saliency volume, created using interest point detection strengths and the depth stream
$\boldsymbol{I}_{rmd\text{-}4d}$	The response volume for the RMD-4D operator
$\boldsymbol{\eta}$	A 3 dimensional spatial offset
$\hat{x}, \hat{y}, \hat{\tau}$	The dimensions of a 3D cuboid
$\hat{x}, \hat{y}, \hat{z}, \hat{\tau}$	The dimensions of a 4D hyper-cuboid
$\boldsymbol{I}_{cont}(x, y, \tau)$	The saliency content of a spatio-temporal region with origin (x, y,τ).
$\boldsymbol{I}_{cont\text{-}4D}(x, y, z, \tau)$	The saliency content of a spatio-temporal region with origin (x, y,z,τ).

ρ	A single descriptor in the bag of visual words approach
$\omega\left(\boldsymbol{I}, (x, y, \tau)\right)$	A function, returning the coarsely quantized joint histogram, of values from observations $\boldsymbol{I}$ in a local region around (x, y, τ)
ρ_{4D}	A single descriptor in the bag of visual words 4D approach

Introduced in chapter 4

$\dot{x}, \dot{y}, \dot{z}$	Spatial position in 3D world co-ordinates
$v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}$	Velocity in 3D world co-ordinates
v_x, v_y	Velocity in 2D on the image plane
d	The disparity of a points projection between sensors
v_d	Out of plane velocity in terms of disparity change between frames
$\boldsymbol{r}$	3D structure point (a vector specifying $\dot{x}, \dot{y}, \dot{z}$)
$\boldsymbol{r}_t$	3D structure point at frame $\boldsymbol{r}$
$\boldsymbol{v}$	3D motion vector, specifying $v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}$
$\boldsymbol{p}\left(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}\right)$	Scene probability distribution
$\boldsymbol{p}\left(\boldsymbol{r}, \boldsymbol{v}\right)$	Prior scene probability
$\boldsymbol{p}\left(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v}\right)$	Scene likelihood
$\boldsymbol{p}_a(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v})$	Scene appearance likelihood
$\boldsymbol{p}_d(\boldsymbol{I}_d \mid \boldsymbol{r}, \boldsymbol{v})$	Scene depth likelihood
$\boldsymbol{p}_p(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})$	Scene likelihood (patch based)
$\boldsymbol{p}\left(\boldsymbol{r}_t, \boldsymbol{v}_t \mid \boldsymbol{r}_{t-1}, \boldsymbol{v}_{t-1}\right)$	Scene transition model
M	Number of appearance sensors
m	Index of an appearance sensor
L	Number of depth sensors
l	Index of a depth sensor
c	Index of a general sensor (either depth or appearance)
$\boldsymbol{\Pi}_{1\ldots M}$	Appearance sensor projection functions
$\boldsymbol{\Psi}_{1\ldots L}$	Depth sensor projection functions

$\boldsymbol{\Psi'}_{1...L}$ Depth sensor reverse projection functions

$\boldsymbol{I}^{st}_{a,m}$ Frame t from appearance sensor m, at scale s

$\boldsymbol{I}^{st}_{d,l}$ Frame t frame from depth sensor l, at scale s

$\bar{I}$ Mean observed colour of a point

e_a The kurtosis parameter for the measurement model of appearance data

e_d The kurtosis parameter for the measurement model of depth data

P The set of Scene Particles

N The cardinality of P

n Index into P

j Index into P

$\boldsymbol{P}n$ The n'th Scene Particle from the set P

P_t The set of Scene Particles at a particular time t

w_n The weight of the n'th Scene Particle

$\bar{w}_n$ The normalised weight of the n'th Scene Particle

$\Theta_{C,x,y}$ The subset of Scene Particles along the ray originating from pixel x, y in sensor c

R The number of ray resampling subsets of the particle population

$\boldsymbol{\Phi}$ Set of $\boldsymbol{r}$,$\boldsymbol{v}$ values visible within a scene

$\boldsymbol{\mu}_s$ The mean observable ($\boldsymbol{r}$,$\boldsymbol{v}$) values in the scene

$\boldsymbol{\sigma}_s$ A diagonal covariance matrix, with each dimension scaled by the range of observable ($\boldsymbol{r}$,$\boldsymbol{v}$) values in the scene

$\boldsymbol{\gamma}$ A 6 dimensional offset in ($\boldsymbol{r}$,$\boldsymbol{v}$)

δ Scaling parameter for diffusion

$\boldsymbol{I}_o$ The output image generated from a Scene Particle cloud, with 4 channels encoding $v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}$ and z for each pixel

$\boldsymbol{I}_g$ The ground truth image, containing 4 channels encoding the true $v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}$ and z at each pixel

err_{of} Error measure for image plane flow magnitude

err_{sf} Error measure for out of plane flow magnitude

err_{ae} Error measure for flow directional

err_{st} Error measure for structure reconstruction

err^n_{of} Normalised error measure for image plane flow magnitude

err^n_{sf} Normalised error measure for out of plane flow magnitude

err^n_{ae} Normalised error measure for flow directional

err^n_{st} Normalised error measure for structure reconstruction

$\boldsymbol{p}_s(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v})$ Skin region likelihood

$\boldsymbol{p}_{sk}(\bar{I})$ Skin probability for a given colour $\bar{I}$

$\boldsymbol{p}_{bg}(\bar{I})$ Background probability for a given colour $\bar{I}$

Introduced in chapter 5

$\boldsymbol{u}_C$ The 3D world co-ordinates of sensor c

$\bar{\boldsymbol{I}}_p$ The "mean" patch formed by performing pixelwise averaging on a number of patches

$\hat{\boldsymbol{\theta}}_C$ A unit vector along a ray of sensor c

u Distance along a ray

$\boldsymbol{p}^t(\boldsymbol{r}_t)$ The prior probability with velocity dimensions marginalised out

$\boldsymbol{o}$ A 2 dimensional pixel offset

Ω The set of pixel offsets A 2 dimensional pixel offset ($\boldsymbol{o}$) comprising a local image region or patch.

σ_{i2} The variance of the spatial weighting function during 2D bilateral filtering

σ_{v2} The variance of the value weighting function during 2D bilateral filtering

σ_{i3} The variance of the spatial weighting function during 3D bilateral filtering

σ_{v3} The variance of the value weighting function during 3D bilateral filtering

$\boldsymbol{p}_v(c, \boldsymbol{r}, t)$ Visibility probability for point at $\boldsymbol{r}$ in sensor c at time t

$\bar{\boldsymbol{p}}_v(\boldsymbol{r})$ Average visibility probability for point at $\boldsymbol{r}$ across all sensors and frames

$\hat{\boldsymbol{I}}_o$ The smoothed version of $\boldsymbol{I}_o$

$\hat{P}$ The smoothed set of Scene Particles

$\hat{\boldsymbol{P}}n$ The n'th Scene Particle from the smoothed set $\hat{P}$

$\hat{v}_{\dot{x}}, \hat{v}_{\dot{y}}, \hat{v}_{\dot{z}}$ Smoothed velocities in 3D world co-ordinates

Λ	The set of 3D offsets A 3 dimensional spatial offset ($\boldsymbol{\eta}$) comprising a local neighbourhood in world co-ordinates
Γ	The set of supporting pixel locations in terms of x, y,t
Ψ	The set of supporting pixel appearance values
ψ	A supporting pixel value, i.e. an element from Ψ
Ψ_g	The set of supporting pixel gradient values
$\bar{\boldsymbol{I}}_m$	The mean appearance value of a patch in sensor m
ABS	The sum absolute differences matching metric
SQ	The sum square differences matching metric
ABS_g	ABS when applied to gradient images to embody gradient constancy
SQ_g	SQ when applied to gradient images to embody gradient constancy
OFC	The "Optical Flow Constraint" matching metric
F_{dif}	The set of pixel difference features
F_{var}	The set of pixel variance features
F_{pix}	The set of raw pixel value features
F_{pm}	The set of patch mean features
F_{pv}	The set of patch variance features
F_{ph}	The set of patch histogram features
ζ	The number of trees used in an RDF classifier
ϕ	The depth of the trees used in an RDF classifier

Introduced in chapter 6

ϵ	The number of spatial cells the neighbourhood Ω is split into, when calculating histogram features
$\Omega_{\epsilon\epsilon}$	The set of 2D offsets definining cell (ϵ,ϵ)
$\omega_{\epsilon\epsilon}$	The histogram for cell (ϵ,ϵ)
ω_u	The unnormalised local histogram, created by concatenating all cell histograms
b	Index of a bin from a histogram
B	The number of orientation bins for a cell histogram

B_{azi} The number of azimuth orientation bins for a cell histogram

B_{ele} The number of elevation orientation bins for a cell histogram

$\boldsymbol{I}_{mag}$ The flow magnitude image

$\boldsymbol{I}_{ori}$ The flow orientation image

$\boldsymbol{I}_{mag\text{-}p}$ The scene flow magnitude image, created using the projected scene flow image $\boldsymbol{I}_o$

$\boldsymbol{I}_{azi\text{-}p}$ The scene flow azimuth image, created using the projected scene flow image $\boldsymbol{I}_o$

$\boldsymbol{I}_{ele\text{-}p}$ The scene flow elevation image, created using the projected scene flow image $\boldsymbol{I}_o$

$\omega^{p}_{\epsilon\epsilon}$ The projected scene flow histogram for cell (ϵ,ϵ)

$\bar{\boldsymbol{P}}_{C,x,y}$ The weighted average scene particle, from ray x, y in sensor c

$\boldsymbol{I}_{mag\text{-}3d}$ The scene flow magnitude image, created using the weighted average scene particle $\bar{\boldsymbol{P}}_{C,x,y}$

$\boldsymbol{I}_{azi\text{-}3d}$ The scene flow azimuth image, created using the weighted average scene particle $\bar{\boldsymbol{P}}_{C,x,y}$

$\boldsymbol{I}_{ele\text{-}3d}$ The scene flow elevation image, created using the weighted average scene particle $\bar{\boldsymbol{P}}_{C,x,y}$

$\omega^{3d}_{\epsilon\epsilon}$ The 3D scene flow histogram for cell (ϵ,ϵ)

$\omega^{3d\text{-}azi}_{\epsilon\epsilon}$ The histogram of 3D scene flow azimuths, for cell (ϵ,ϵ)

$\omega^{3d\text{-}ele}_{\epsilon\epsilon}$ The histogram of 3D scene flow elevations, for cell (ϵ,ϵ)

$\omega^{3d\text{-}ind}_{\epsilon\epsilon}$ The 3D scene flow histogram, with independently encoded azimuth and elevation for cell (ϵ,ϵ)

$\omega^{2d}_{\epsilon\epsilon}$ The scene flow histogram for cell (ϵ,ϵ) using a single elevation bin (approximating optical flow)

$\omega^{3d\text{-}int}_{\epsilon\epsilon}$ The 3D scene flow histogram, for cell (ϵ,ϵ) using an ITF

List of Figures

List of Tables

Declaration

The work presented in this thesis is also present in the following manuscripts:

[1] Simon Hadfield and Richard Bowden. Scene particles: Unregularized particle based scene flow estimation. *Pattern Analysis and Machine Intelligence*, Under Review, 2013.

[2] Simon Hadfield and Richard Bowden. Hollywood 3D: Recognizing actions in 3D natural scenes. In *Proceedings, Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[3] Simon Hadfield and Richard Bowden. Go with the flow: Hand trajectories in 3D via clustered scene flow. In *Proceedings, International Conference on Image Analysis and Recognition*. Springer, June25 - 27 2012.

[4] Simon Hadfield and Richard Bowden. Kinecting the dots: Particle based scene flow from depth sensors. In *Proceedings, International Conference on Computer Vision*, pages 2290 – 2295, Barcelona, Spain, 6-13 Nov 2011.

[5] Simon Hadfield and Richard Bowden. Generalised pose estimation using depth. In *Proceedings, International Workshop on Sign, Gesture, Activity*, Heraklion, Crete, September 5 – 11 2010.

Chapter 1

Introduction

The work presented in this thesis is motivated by the recent rise in 3D video data, and the need for efficient algorithms, in order for such data to be fully exploited within classic computer vision tasks. Particular focus is given to the estimation of 3D structure and motion fields, and their use in the recognition of human actions. The greatest challenge in this application area tends to arise due to the level of generalisation necessary for widespread deployment. The 3D features presented mitigate this to some extent, by offering invariance to a number of common sources of variation, such as changes in lighting or viewpoint and differences in actor appearance. The algorithms are capable of operating with any combination of appearance sensors (RGB or greyscale) and depth sensors, including appearance only configurations, and setups employing multiple depth sensors. Additionally, any camera arrangement may be utilised, providing the calibration is known (i.e. techniques are not restricted to narrow baseline or coplanar configurations).

This initial chapter highlights the motivations of the work, the research aims and the associated constraints. In addition, an overview of the thesis structure is provided, with outlines of the contributions present in the remaining chapters.

Human action recognition is one of the most broadly applicable areas of computer vision. Human Computer Interaction (HCI) techniques are important for the next generation of natural user interfaces, which may be employed in office software, as well

as home cinema and gaming. Similarly surveillance applications encompasses many indoor and outdoor scenarios, in a vast array of possible locations. Video categorisation tasks are often applied to enormous, unconstrained data repositories such as youtube. In addition to providing some of the largest application areas, these tasks also involve some of the greatest levels of variation, the same action may be performed in many different ways by different people, but still have the same semantic label.

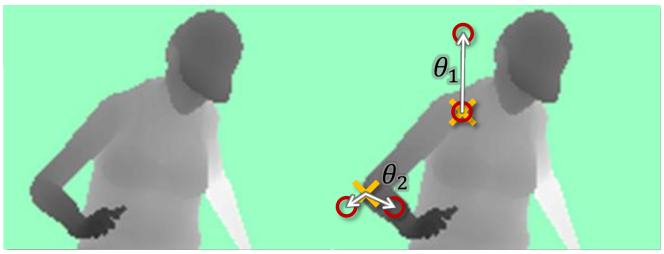
Attempting to take vision techniques out of the lab in order to solve these tasks, provides unique challenges which are rarely considered during the technique's development. The lack of control over the setting favours simple, highly invariant descriptors, as more complex features often lead to overfitting on restricted datasets. This was highlighted recently, with the release of the Microsoft Kinect^{TM}and the work of Shotton *et al.* [113], which brought computer vision into consumers living rooms (figure 1.1a). To operate in such unconstrained scenarios, structural information was employed (figure 1.1b), with simple features based on depth comparisons (figure 1.1c). This provided the many invariances discussed previously while also simplifying the segmentation of objects and the detection of occlusions. The work presented in this thesis follows a similar vein, focussing on the use of general descriptors, extracted from invariant data sources, with application to unconstrained natural datasets.

There has been extensive investment in producing such unconstrained (sometimes referred to as "in the wild") datasets for Action Recognition [73, 83, 79]. Some examples are shown in figures 1.2b,1.2d and 1.2f compared to older "staged" datasets (figures 1.2a,1.2c and 1.2e). However, in previous work, structural information (or the ability to infer and reconstruct structural information) is neglected. Hence, much of the work presented in this thesis is evaluated using a newly captured dataset which has been made available to the community to support future work.

Chapter 2 discusses the current state of the art, in the fields of action recognition and motion estimation. This mirrors the order these topics are introduced in the remainder of the thesis. In each case, the position of this thesis in relation to previous work is highlighted.

Chapter 3 explores the use of structural information for recognising unconstrained hu-

(a) Kinect in the living room



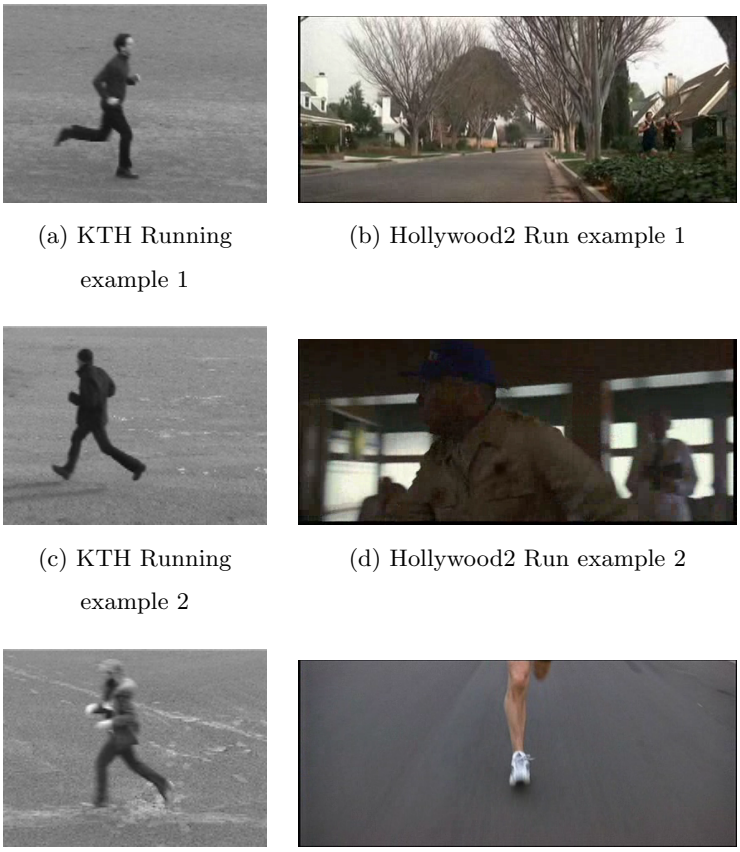
(b) An output depth map

(c) Pointwise depth comparison features

Figure 1.1: Generally applicability of Kinect. The use of depth based features (1.1c)) allows operation in extremely varied environments such as consumers living rooms (1.1a). Images taken from xbox.com and [113] respectively.

man actions. Initially the collection of the 3D dataset is discussed, including both automatic and manual annotation schemes. Next, existing state of the art techniques are explained in detail, followed by proposed extensions to enable encoding of structural information. These proposed extensions include five interest point detection schemes, and two feature descriptors. The analysis then shows that all of the proposed approaches lead to improved performance over the simple appearance based techniques examined.

Chapter 4 investigates the improvement of the motion features used in chapter 3 by incorporating structural information. Specifically, the estimation of 3D motion fields is explored (also known as scene flow fields, in contrast to 2D Optical Flow fields). Figure 1.3 shows an example of such an estimated 3D motion field. Existing scene flow estimation techniques are found to be impractically slow for the quantities of data un-



(a) KTH Running example 1

(b) Hollywood2 Run example 1

(c) KTH Running example 2

(d) Hollywood2 Run example 2

(e) KTH Running example 3

(f) Hollywood2 Run example 3

Figure 1.2: Staged vs "in the wild" Actions. Examples are from the "run" class of the staged KTH dataset (1.2a,1.2c and 1.2e)[109] and the "in the wild" Hollywood2 dataset (1.2b,1.2d and 1.2f)[83].

der consideration. In addition, the oversmoothing artefacts present in most techniques reduce accuracy in regions which tend to be most salient. Thus, a novel probabilistic approach to scene flow estimation is proposed, inspired by monte-carlo estimation techniques, and specifically particle filtering. This approach is capable of maintaining multiple motion hypotheses at each point, obviating the need for smoothness assumptions, as constraints can accumulate over time to resolve ambiguities. As discussed previously, techniques are formulated to operate with any combination of appearance and depth sensors in any configuration. When depth sensors are not present the algorithm per-

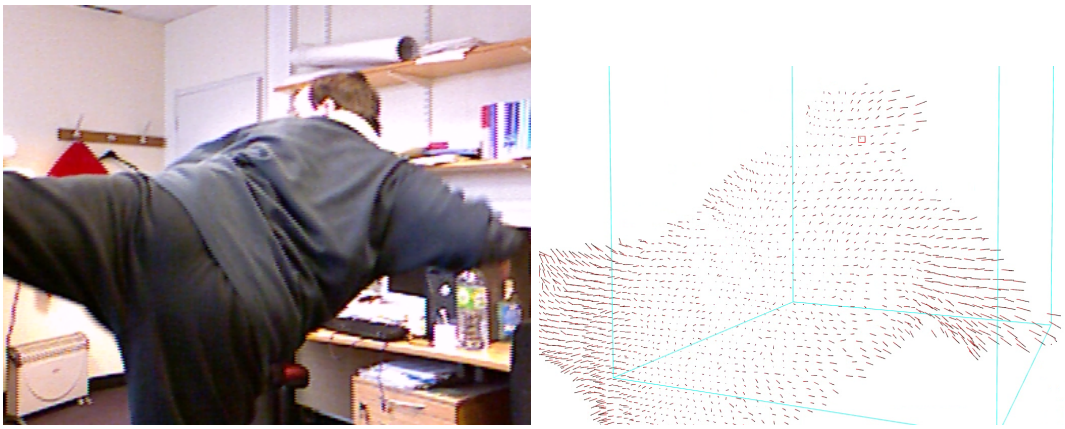


Figure 1.3: An estimated 3D motion field (right), from a sequence of a person performing a kicking action (left). Motion vectors move from the red to purple vertices.

forms simultaneous motion and structure estimation. The speed and flexibility of the algorithm, enables a vast array of new applications to exploit 3D motion information, beyond the intended use as a feature descriptor for action recognition. An example of such applications is demonstrated, where 3D hand trajectories are extracted for 3 million frames of sign language footage, by clustering the estimated motion field.

Chapter 5 improves upon the scene flow estimation system, by a re-evaluation of the assumptions underlying motion estimation. Techniques are explored to include smoothness assumptions, while maintaining the useful properties of the algorithm. In addition, the inclusion of the constant velocity assumption, and probabilistic handling of occlusions, is demonstrated. Finally an analysis of the most fundamental assumption, that of brightness constancy, is presented. This analysis brings to light a number of issues previously noted in the motion estimation literature, and explains the popularity of a number of techniques, including multi-scale iterative motion estimation. Finally, a more robust formulation is proposed, based on machine learning, and is incorporated into the motion estimation system, with excellent results.

In chapter 6, a feature descriptor is developed to encode the estimated 3D motion of a scene, as shown in figure 1.4. The descriptor is inspired by work on HOFs, but based on the richer 3D motion field. This descriptor is then incorporated into the action

Figure 1.4: The HOS descriptor on an example sequence of a waving hand. Local motions are encoded into 4 orientation bins, with the length of the line relating to the strength of the motion in that direction. The 4 images relate to different levels of out of plane motion (top left involves motions towards the camera, bottom right is motions away from the camera, the other images contain intermediate bins). Green squares are "blocks" of normalised motions.

recognition systems proposed in chapter 3. The inclusion of 3D motion features is shown to significantly improve recognition rates, over the use of image plane motion alone.

Finally, chapter 7 brings these contributions together, summarizing the value of 3D structure and motion information for unconstrained recognition tasks. Plans for future work are also presented, including improvements to the proposed action recognition schemes, and the application of scene flow estimation in other areas of computer vision.

To briefly summarise, the contributions of the work presented in this thesis are:

- A publicly available dataset for unconstrained action recognition, including structural information.

- Publicly available code for the reproduction of all baseline results on the above dataset.

- Five interest point detection schemes, employing structure information when determining saliency.
- Two feature descriptors based on the current state of the art in action recognition, and capable of encoding structural information.
- An extremely fast and accurate approach to 3D motion estimation, without the presence of over-smoothing artifacts.
- A multi-modal framework for the application of the above motion estimation system, with any combination of appearance and depth sensors.
- An approach for estimating 3D occlusion maps during motion estimation.
- A more robust brightness constancy formulation based on machine learning.
- An automatic 3D tracking scheme, based on estimated motion fields.
- A general feature descriptor, for encoding 3D motion information.

Chapter 2

Literature Review

This chapter discusses the current state of the art in a range of topics. Initially, general techniques for human action recognition are examined, with discussions of interest point detection, local feature encoding and holistic recognition. In addition existing work on the simultaneous estimation of 3D structure and motion is discussed, for both sparse and dense scenarios.

2.1 Action Recognition

Many state of the art approaches to action recognition follow a general process, consisting of 4 stages. First, the sequence is sampled at a number of spatio-temporal locations. Second local feature descriptors are extracted around each sample point. Third the collection of local features is encoded into a single holistic descriptor for the sequence. Finally this sequence descriptor is passed to a discriminative classifier, in order to select the most likely category. This approach is analogous to the highly successful bag-of-words techniques from the object recognition literature, but accounting for the additional temporal dimension.

The simplest approach to sampling is a dense sampling scheme (for example, sampling on a regular spatio-temporal grid). In this case many samples will lie in background regions, rather than on the objects involved in the action. To reduce this contamination,

many authors extend the dense sampling concept beyond a uniform grid [25, 57]. In these works, features are calculated via integration across the entire spatio-temporal volume, with the contributions at each point weighted by saliency.

In some situations, features extracted from background regions can sometimes provide useful information about the context of actions. Wang *et al.* [129] argue that densely sampled features, which incorporate both action and context information, may outperform sparse features (albeit at far higher cost). Some authors take this idea a step further, and attempt to model context directly. Han *et al.* [61] recognise objects and object parts in the scene, and create an additional feature channel, based on the number and configuration of detections. Marszalek *et al.* [83] perform a separate scene classification step, and combine this with prior knowledge about probable action contexts (for example the "Get Out Car" action is unlikely to occur indoors) demonstrating improved recognition rates.

Although dense sampling has its advantages, sparse sampling schemes are often favoured, due to the large scale of many action recognition datasets. Such sparse sampling relies on the detection of salient regions or "interest points". This serves to reduce the "noisiness" of the descriptor, assuming there is correlation between the interest point detector and the object performing the action.

2.1.1 Interest Point Detection

The literature contains a wide variety of interest point detection schemes, each based on a different concept of saliency. One of the oldest detectors still in common use is the corner detector introduced by Harris and Stephens [62], which is extremely fast to compute. The detector examines local image gradients at every pixel, and identifies positions with strong intensity changes in 2 distinct directions. These points relate to corners, where the edge directions are not restricted to lie along the x, y axes. Corners are often considered salient as the constrains of two edges serve to solve the aperture problem, allowing precise localisation and tracking. Although originally designed for detection of interest points in images, the approach has been applied to temporal sequences, by examining the 3 orthogonal planes (xy, xt and yt) in isolation [53].

The Harris detector has also been extended by Laptev and Lindeberg, to operate directly within space-time volumes [71]. In this case, detected interest points are those with strong intensity gradients along 3 different spatio-temporal directions (not restricted to x, y, t axes). Obviously the computational time is significantly increased over 2D corners, however the detected points are assumed to be more salient.

Dollar *et al.* proposed a scheme even faster than the original Harris corner, referred to as "separable linear filters" [45]. In this approach, the spatial and temporal dimensions are treated separately, using different filtering schemes. The resultant interest points correspond to spatial regions (rather than points) which, as a whole, exhibit intensity peaks, troughs and edges over time. Such points correspond to moving regions of high contrast, such as moving object boundaries.

More recently, Willems *et al.* extended the Beaudet Saliency measure [22] to the spatio-temporal domain, for the purpose of action recognition [134]. In this case, salient points are defined as locations with strong second order intensity derivatives, including both spatio-temporal blobs and saddles.

It's difficult to determine which of these definitions of saliency is the most valid for a particular task, as saliency itself is often poorly defined. Performance comparisons for 3D Action Recognition are presented in section 3.3.1 of this manuscript. Further discussion and comparisons can also be found in the review paper by Tuytelaars and Mikolajczyk [121].

Regardless of the specific interest point detection scheme employed, the scale of the salient features remains unknown. Considering interest points at various scales can provide an increased number of detections, and the scale of the salient region can inform subsequent stages of the system, such as feature extraction. The simplest and most commonly employed approach, is to run the interest point detection on a small set of discrete scales [72]. However, there has also been some work on the automatic detection of optimal scale parameters, for a previously detected interest point. Lindeberg proposed an approach for achieving this, by examining the saliency response as a function of scale parameters [78]. This idea can be applied to determine both the spatial and temporal scales of interest points [77]. However, most interest point

detectors are not scale invariant, thus positions which exhibit maxima in the saliency measure at one scale, may no longer be maxima when saliency is measured at another scale. This leads to iterative schemes, where the position and scale optimisations are repeated until convergence, for each detected interest point [35, 71]. Some approaches have been proposed to avoid this iterative scheme, which rely on scale-invariant interest point detectors such as the Harris-Laplace operator of Mikolajczyk and Schmid [85] or the Hessian points of Willems *et al* [134]. In these cases, a single criteria can be optimised, which simultaneously determines the position and scale parameters.

2.1.2 Local Feature Descriptors

When the sampling points have been determined, local features descriptors are calculated. Again, many different descriptors have been exploited in previous work. The simplest and most easily computed descriptor was proposed by Dollar *et al.* based on the histogram of either pixel or gradient values, in the spatio-temporal cuboid surrounding the interest point [45]. Schuldt *et al.* also use gradient information, however they suggest calculating gradients only at the centre of the cuboid, to form a jet descriptor [109]. Ke *et al.* [68] proposed using motion information based on optical flow, to produce volumetric features similar to Haar-features [127]. Laptev *et al.* [73] then incorporated both the appearance and motion information of previous approaches, utilising HOG [38] and HOF [39] descriptors, and found that both provided complimentary information for action recognition.

There are also a large number of spatio-temporal feature descriptors, developed by extending traditional 2D features from the field of object recognition. The spatial Scale-Invariant Feature Transform (SIFT) descriptor [80] was extended by Scovanner *et al.* to encode temporal information [110]. In a similar vein, Willems *et al.* [134] extended the Speeded Up Robust Feature (SURF) descriptor [20], and Klaser *et al.* [69] extended the HOG descriptor, into the spatio-temporal domain. This work is similar to that presented in chapter 6 of this thesis, where the HOF descriptor (an inherently spatio-temporal descriptor) is extended into 4D, utilising 3 dimensional flow fields.

Some authors have explored using the interest point detections themselves as a feature descriptor. Oshin *et al.* proposes a descriptor based on the interest point detection strengths within two spatio-temporal sub-volumes [93]. This is similar to the spatio-temporal Haar-like features of Ke *et al.* [68], but based on relative saliency distributions rather than motion. Similarly Gilbert *et al.* demonstrate excellent performance, mining spatio-temporal configurations of interest points [53].

2.1.3 Sequence Encoding

The next stage is to accumulate the local features from every sample position, in order to form a holistic descriptor of the entire sequence. If the local features used are discrete, they can be accumulated directly into a histogram, as in the work by Oshin *et al.* [94]. For the more common case of continuously valued features, the codebook and bag-of-words approach (popularised in object recognition) is generally employed. This approach attempts to cluster the samples in feature-space, extracting a small subset of exemplar features, called the codebook. Sequences are then encoded, by matching each sample to the closest exemplar of the codebook, and producing a histogram of occurrences for each exemplar. Performing this accumulation spatially and temporally implies that the holistic descriptor is invariant to a range of deformations, such as spatial and temporal translations, stretching, reflections etc. Although this is valuable for generalisation, it also discards much of the relational information, such as the spatial configuration and temporal ordering of features. Laptev *et al.* attempt to mitigate this by splitting the spatio-temporal volume into sub-blocks, creating a descriptor for each sub-block, and concatenating them to create the sequence descriptor [72]. Sapienza *et al.* follow a similar vein, encoding individual sub-sequences, however rather than concatenating to create a single descriptor, they employ Multiple Instance Learning (MIL) [105]. This accounts for some parts of the sequence being irrelevant, for example before and after the action.

The purpose of encoding features into a sequence level descriptor, is to allow subsequent machine learning techniques to categorise the sequence in its entirety. However, approaches to action recognition have been proposed, which avoid machine learning

altogether. In these cases, no holistic sequence descriptor is required, instead a voting scheme is employed, with each sample providing one vote. An advantage of such techniques, is that they are able to handle videos containing more than one action, by detecting multiple peaks in the voting space, which is not possible using a holistic descriptor of the sequence. A second advantage is that actions may be localised, both spatially and temporally, by examining the samples which contribute during voting. Gilbert *et al.* propose one such technique, employing data-mining to learn frequent descriptors for each action class [54]. A weighted voting scheme is then used, with the weight of each patterns vote determined by the distinctiveness of that pattern. Uemura *et al.* proposed an alternative tree-based scheme for fast voting, using tracked SIFT features [58]. They also demonstrated a motion compensation scheme, to remove noise from the features tracks, caused by camera motion.

A third possibility for recognition, is to examine each frame in isolation, then classify the video based on the sequence of frames. An example of this approach, is to equate each frame with a state in a Hidden Markov Model (HMM). By building one HMM per class, recognition can be performed for a test sequence, by determining which HMM is most likely to have produced the observed state sequence [26]. This has the advantage that it accounts for temporal ordering, however it can be very difficult to determine the correct structure of the HMM for a given task. Additionally, in order to provide good generalisation, a significant amount of training data is required.

2.2 3D Motion Estimation

Motion features have proved valuable for action recognition, which is unsurprising given the temporal nature of the task. One of the most obvious deficiencies of such features, is their definition in terms of image plane motion, while the action to be recognised is performed within the 3D scene. This limitation is understandable, as a historic artifact, but given the prevalance of commercial depth sensors, and the emergence of 3D broadcast footage, it should now be possible to use more natural and rich 3D motion features.

The earliest work estimating the 3D motion of a scene from video, comes from the

field of Structure from Motion (SfM). This work was primarily concerned with the use a monocular moving camera, exploiting the parallax effect in order to estimate the 3D structure of a static, rigid, scene [36]. However, these techniques are equally applicable to moving rigid scenes, observed by a static camera (indeed these phenomena are duals of each other) [17]. Later work relaxed the rigidity assumption, using non-rigid factorization [119] in conjunction with non-linear optimization [42] and Markov Random Fields (MRFs) [130]. In general, these SfM approaches estimate dense reconstructions, while the related field of Simultaneous Location and Modelling (SLAM) has focused on sparse reconstruction by tracking feature points. In SLAM systems, the tracked features often relate to the same interest points used for action recognition (discussed in section 2.1.1 above), including Harris corners [21], the Shi-Tomasi detector [112] used by Davison *et al.* [40, 41] and FAST corners [102] used by various authors [70, 31]. SLAM techniques generally perform the 3D estimation, using either Kalman filter variants or Bundle Adjustment optimisation. These techniques were contrasted by Strasdat *et al.* who demonstrated that optimisation techniques are more efficient, for all but very low accuracy systems [117]. In recent years, the distinction between SfM and SLAM has become more blurred, as the focus for SLAM moves towards dense reconstructions, utilising modern hardware [88, 89, 90].

The field of Scene Flow Estimation (SFE) was introduced by Vedula *et al.* [124] and also attempts to reconstruct a scene's 3D structure and motion. However, SFE differs from SfM and SLAM techniques, by removing the assumption of a rigid (or piecewise rigid) scene, instead reconstruction is possible even for fully deformable scenes. In order to make this task tractable, SFE techniques focus on the aggregation of information from multiple sensors, rather than employing the parallax within a monocular system. Most SFE algorithms have focused on multi-view appearance data, however Spies *et al.* proposed a technique based on depth sensors, termed "Range Flow" [115], which was later combined with appearance information by Lukins and Fisher [82] and further developed by Schuchert *et al.* [107, 108]. The work presented in chapter 4 is applicable in all these situations, with any number of appearance and/or depth sensors. Similar to the previous discussion of SfM and SLAM, approaches to SFE exhibit a dichotomy, between sparse and dense estimation.

2.2.1 Sparse Scene Flow Estimation

Local scene flow approaches are closely related to the field of surface tracking [29, 116], particularly when used in conjunction with an underlying structure mesh. One particular difference between "scene flow" and "surface tracking" stems from their respective temporal behaviours. In general, scene flow estimates the motion for different points at every frame, based on which parts of the scene align with the pixel centres. There is no assumption in scene flow, that the points being tracked now correspond to points tracked previously. In contrast surface tracking is based on detecting localisable areas of the scene, and following those areas throughout the sequence, regardless of their sub-pixel positions. This difference implies that surface tracking techniques are susceptible to drift, where the surface points being tracked at the end of the sequence may not correspond to the initially tracked points. This makes surface tracking algorithms less reliable for estimating motion fields (scene flow), particularly in longer sequences. In contrast scene flow algorithms do not suffer from drift, however they are also unable to provide the long term trajectories necessary for mesh deformations.

Some of the earliest work in this field was done by Carceroni and Kutulakos, where a collection of "surfels" were tracked in a multi-view sequence, comprised of seven appearance sensors [30]. The surfels relate to small, deformable surface patches, with associated reflectance parameters. The original formulation required the configuration of all light sources to be known with respect to the cameras, however Devernay *et al.* extended the technique to perform 3D point tracking in unknown scenes [44].

These surfel tracks provide a small number of very reliable 3D trajectories, however this is not suitable for some motion estimation applications. Specifically, no motion estimate is available for locations to which no surfels project, and any general statistics derived from the motion estimate will be inherently biased. To address this Nuemann and Aloimonos proposed a technique [87] using space carving [111] to initialise an object mesh, and then tracking the mesh between frames using brightness constancy. This allows a motion estimate to be produced at any point on the mesh via interpolation, with the restriction that the motion of points respect the restrictions of the mesh. Furukawa and Ponce build on this idea, by tracking each mesh vertex individually,

then deforming the previous mesh to fit the new vertices [52]. This approach was further developed by Courchay *et al.* using a variational framework for vertex tracking [37]. However, these approaches required an initial mesh to be provided at the first frame.

2.2.2 Dense Scene Flow Estimation

A dense scene flow field can be equated to an optical flow field, with the addition of out of plane motion at every pixel. Indeed, the initial work on SFE by Vedula *et al.* focused on the combination of optical flow from multiple viewpoints, in order to estimate a mutually consistent 3D flow field [124, 126]. Similar to early work on SfM, dense SFE attempts to estimate the motion of all points within a scene, without necessarily finding exact correspondences between one frame and the next.

The aperture problem is a well documented issue in optical flow estimation. In brief, when an edge point undergoes motion within a video, only the component of the motion perpendicular to the edge can be retrieved. Any motion parallel to the edge cannot be determined, assuming the edge covers the aperture (the local patch under consideration). This is why corner points are often considered salient, as the two edge constraints at these locations allow motion to be uniquely determined. Unsurprisingly due to the similarity between the tasks, similar issues occur in SFE algorithms. Spies *et al.* demonstrate that for points lying on a planar surface, motion is only determined in 1 dimension, and in two directions when the point lies on an edge formed by two planes [115]. The full 3 dimensional scene flow is only uniquely determined for corners, at the intersection of 3 planes, or for regions with a unique non-planar geometry.

Obviously this means that the estimation of scene flow over an entire scene is highly ambiguous, with the vast majority of points being under-constrained. As a result, authors generally resort to the assumption of smoothness. By assuming that the motion field is smooth, constraints may be combined from neighbouring points, in order to determine their motion. Similar schemes have already been proposed for optical flow estimation, and generally fall into two categories. The first is local patch based schemes, inspired by the work of Lucas and Kanade [81], where the motion at each pixel is

estimated using constraints from a subset of neighbouring pixels. The second method uses variational optimisation, inspired by the work of Horn and Schunck [63], where the motion field is optimised globally for the image, including a regularisation term to penalise the total variance of the estimate. For SFE tasks, the latter approach is the most common, as it has the useful property of "filling in" motion in untextured regions (where local approaches fail), by smoothly interpolating from the boundaries, so as to minimise total variation. A range of optimisation schemes are then employed to obtain an estimate of the final motion field, including Gradient Descent [51], Euler-Lagrange [19] and Dynamic Programming [55]. The energy function is obviously highly non-convex, due to the dimensionality of the task and the unstable data matching term. In an attempt to reduce the effect of local minima, coarse to fine strategies are often employed, although the estimate is still unlikely to be globally optimal.

A wide range of smoothness terms have been proposed. The simplest, used by Huguet *et al.* is based on the spatial gradient of the motion field, within the image plane [64]. More recently Basha *et al.* proposed a 3D point cloud formulation, where smoothness assumptions are more valid [19] (a smooth motion field in 3D can project to an un-smooth one in the image plane). Unfortunately, object boundaries produce discontinuities in the motion field, which do not fit with the underlying assumption of smoothness. In these regions, the erroneous assumption is counterproductive, and the combination of constraints from both sides of the discontinuity, leads to "blurred" estimates. These are referred to as oversmoothing artefacts, and it was noted by Basha *et al.* that they are the primary source of errors for most modern SFE techniques.

A wide range of approaches have been proposed for mitigating these over-smoothing artefacts. One of the simplest approaches, employed by Wedel *et al.* is to reduce the weight of the smoothness term at each point, based on the local intensity gradient [133, 131]. This follows from the assumption that object boundaries will produce strong edges in the image, however this may lead to difficulties estimating the motion of highly textured regions.

Zhang and Kambhamettu [135] instead extend the Lucas & Kanade optical flow approach to SFE. In this case, there is no explicit smoothness term, instead motion is

estimated for each local patch, so as to match it to patch's in other viewpoints and frames. Smoothness is still implicit in the approach, as there will be a large overlap in the neighbourhoods of 2 close pixels, however the extent of oversmoothing artifacts is limited by the size of the neighbourhood. Isard and MacCormick employed a similar neighbourhood matching scheme, in a probabilistic framework using belief propagation [67]. In later work Zhang and Kambhamettu combined the idea with a segmentation scheme, such that consistent motions were determined within segments, without propagating constraints between segments [136]. The problem with matching local neighbourhoods (and segments) in this manner, is that for a given region in the scene, the size and shape of its projection is unlikely to be consistent between viewpoints, or over time. Thus, matching similarly shaped regions, implies the assumption that the patch under observation is equi-distant from all viewpoints, with the same orientation relative to all image planes. An approach to loosening this assumption was introduced in later work by Zhang and Kambhamettu, by fitting the parameters of an affine motion model to each segment [137]. A similar approach was followed by Li and Sclaroff, but using a range of different motion models to deform the patch [75, 74].

The effect of structure on reprojection between viewpoints was exploited, by Pons *et al.* at the global level [98]. By reprojecting each image, to every viewpoint, via the current structural estimate, a matching score was developed. This was employed in a variational framework, to iteratively update the estimated structure and motion [97, 99]. The accuracy of the reprojections in this case, depends on accurate knowledge of the camera setup. Indeed, this is the case with most approaches that relax the fronto-parallel assumption, however Valgaerts *et al.* recently proposed the inclusion of stereo geometry parameters into optimisation, such that extrinsic camera parameters and scene flow were estimated simultaneously [122].

Recently, a number of techniques for dense SFE have been proposed, which break with the traditional Lucas-Kanade and Horn-Schunck approaches. Rabe *et al.* perform a traditional variational scene flow estimation, but integrate temporal information into the optimisation, by applying Kalman filtering at each pixel [100]. Cech *et al.* avoid optimisation entirely, instead estimating a small number of high accuracy matches, and using a region growing approach to propagate the motion [32]. Although this eliminates

the explicit regularisation term, region growing is still constrained, so as to produce smooth motion fields. In addition, estimated motion fields are semi-dense, depending on the distribution of seeds.

There has also been a smaller body of work, based on extending space carving and voxel colourisation, for scene flow estimation tasks. Unlike the previously described approaches, smoothness assumptions may be avoided, mitigating over-smoothing artifacts. The idea was introduced by Vedula *et al.* who proposed a 6D space, covering 3D structure and motion, which was carved out based on photo-consistency [125]. In later work, this was simplified to standard 3D shape carving, combined with optical flow estimation in each viewpoint [126]. This "brute force" exploration of possible solutions was costly, and generally relied on a heavily quantised space, with an associated loss of fidelity in the estimate. To reduce this, Ruttle *et al.* included a number of heuristic constraints, including background subtraction and optical flow thresholding [103]. More recently Basha *et al.* developed a voxel based approach, scalable to large numbers of cameras, in which hard structure selection was deferred until after motion had been estimated [18].

These voxel based methods bear the greatest similarity to the work presented in chapter 4 of this thesis, which explores the original continuous space (without voxel quantisation) using a collection of discrete samples. These multiple-hypotheses also obviate the need for smoothness assumptions, allowing the aperture problem to be solved via accumulation of constraints over time, rather than combining constraints spatially. This is similar to the "deferred estimation" of Basha *et al.* , but allows the deferred choices to also improve the estimation of future frames.

Additional information can be found in the recent book of Wedel and Cremers [132], which provides a useful introduction to scene flow estimation, with particular focus on variational schemes and driver assistance applications.

Chapter 3

Action Recognition Using 3D Structure

Performing action recognition tasks in a natural setting is extremely difficult. The relevant areas of the scene are unclear, the appearance and orientation of the subject is unknown, and even the definition of the action is loose (the same high level semantic label, is often assigned to a wide range of actions). These factors lead to a great deal of intra-class variation, which must be compensated for, in order to make the task tractable. This may by achieved in part, by making use of structural information, which highlights occlusions and has inherent invariance to both actor appearance and lighting changes. This approach has been previously demonstrated in other vision tasks, which suffer from such high intra-class variations (for example the general pose estimation task, discussed briefly in Appendix A).

In this chapter, the need for a natural 3D action dataset is addressed. A large number of existing action recognition techniques, are then extended to account for the newly available structural information (including 5 saliency measures and 2 feature descriptors). Finally an in depth analysis is performed, not only comparing all combinations of new techniques, but also exploring their behaviour as the density (and noisiness) of the features increases.

3.1 3D Action Data Collection

In recent years, commercially available 3D displays have been introduced into the market, leading to a sharp rise in commercially available 3D content, including films available on 3D BluRay^{TM}and 3D broadcast footage. In addition, the availability of the Microsoft Kinect^{TM}has greatly simplified the capture of new 3D data. Despite this, work on action recognition has focussed on spatiotemporal sequences (i.e. two spatial dimensions and one temporal dimension). The small number of datasets produced thus far such as the ChaLearn gesture dataset [14] and Human Daily Actions [34] have been extremely limited (see the following section) and in some cases not publicly available [76]. Consequently it is currently unknown how best to exploit such information for action recognition tasks. Thus it was necessary to collate a new dataset for the purposes of this thesis. The dataset was made available to the community at CVPR 2013 [60], along with all source code necessary to reproduce the baseline results from this chapter.

3.1.1 Unstaged Action Data

In the past, most action recognition datasets such as KTH [109] and Weizmann [25], utilised staged data. These datasets are generally considered "solved" with many techniques reporting performance of 95% or more. As a result, recent action recognition datasets have been concerned with action recognition "in the wild", dealing with unconstrained variations in location, lighting, camera angle, actor and action style. It is extremely difficult to incorporate this level of variation in staged data, captured within the lab, and methods designed and trained for such datasets often do not translate well to more natural scenarios such as surveillance or video categorization. From an applications point of view, staged capture is even less desirable due to the limited operating range of sensors such as the Kinect, and the inability to capture accurately, in direct sunlight or outdoor settings. Due to these difficulties, alternative means for obtaining data have been explored, including videos obtained from youtube [79] and clips extracted from commercially available films ("Hollywood" [72] and "Hollywood 2" [83]).

3.1.2 Automatic Extraction

The large size of the Hollywood 2 dataset was due to the automated extraction process employed by Marszalek *et al.* [83]. Such techniques are the cheapest means of data collection. Their approach relies on the extraction of time stamped subtitles from the film. These subtitles are then aligned with film scripts, which contain no timestamps, but describe the behaviour of the actors in addition to dialogue. The alignment of the two produces a script, with weak timestamp labels throughout. By training an Support Vector Machine (SVM) to detect action instructions within the script, the timestamps can be used to automatically extract the clip.

Unfortunately, most 3D films are too recent to have publicly available transcriptions. As an alternative, the use of the "audio description" track was explored. Audio Description is a secondary audio track, which is present in some films and is intended for the blind community. The audio contains a spoken description of the scene contents and the behaviours of the actors.

By employing speech recognition software, the spoken description was converted into textual form, suitable for the automatic extraction of actions. Table 3.1 shows the number of actions extracted from a film, using this technique in conjunction with a range of speech recognition APIs, compared to manual extraction. The performance is poor, mainly due to failures in the transcription process. Most of these dictation tools rely on a training period in order to adapt to the users voice. In addition there are difficulties with conversations involving multiple users. Due of these issues, a manual extraction approach was employed, as in the original Hollywood dataset [72]. This has the additional advantage that actions could be more accurately segmented from their carrier movies.

3.1.3 Manual Extraction

A significant fraction of the available 3D films, were constructed from the original 2D recordings via post-processing techniques, such as rotoscoping. Any depth data extracted from these films would be less rich, with no depth variations within objects.

Speech Recognition	True Positive Rate	Total False Positives
Naturally Speaking	0%	5
Microsoft API	1.1%	7
Android API	3.2%	5
Apple API	8.2%	8

Table 3.1: Performance of automatic action extraction. The fraction of correct actions, automatically extracted using a range of speech recognition APIs, compared to manual annotation. Also shown is the total number of false detections for the film.

This leads to scenes which are fundamentally artificial, created for effect only, and resembling a collection of cardboard cut-outs. Additionally, films generated entirely through CGI, such as "Monsters Inc." are unlikely to provide transferable information for real human actions. For the dataset, clips were only used from films captured using commercial camera rigs such as James Camerons Fusion Camera System^{TM}, or products from 3ality Technica. These technologies produce 3D consumer content from real stereo cameras, and thus the results can be reconstructed into true 3D depth maps.

Imposing this restriction, the amount of acceptable footage was greatly reduced, leaving only 14 films: Resident Evil: Afterlife (RE), Drive Angry (DA), Fright Night (FN), My Bloody Valentine (MBV), Pirates Of the Caribbean: On Stranger Tides (PotC), Sanctum (S), Spy Kids: All The Time In The World (SK), Step Up 3D (SU), Tron: Legacy (T), Avatar (A), The Three Musketeers (TM), Final Destination 5 (FD), A Very Harold and Kumar Christmas (H&K) and Underworld: Awakening (U). The resultant dataset is referred to as "Hollywood 3D" and contains over 650 video sequences, covering 14 different action classes. This puts the dataset on a similar level to the original Hollywood dataset, with an increased number of action classes. In addition, a set of sequences containing none of the chosen actions was randomly extracted as negative data, ensuring no overlap with positive sequences.

Each of the films was split between the train and test sets, on a per action basis. This means that each action is tested on actors and settings that were not seen for that action during training (although that actor or scene may have been observed during

the training of a different action). Table 3.2 shows the training and test split of the data across films and classes, while table 3.3 lists the resulting number of training and test clips for each action. In addition example images from the dataset can be seen in figure 3.1. Due to the manual extraction procedure, all action sequences available in the films were considered. However, only sequences judged to be of sufficient quality, were included within the dataset. Included sequences were at least 10 frames (roughly half a second) in length, and had a field of view sufficient to contain the majority of the action (i.e. videos of *Run* actions where only the actors head is visible, were not included). Shot-cuts were not included within the data, instead the sequences before and after the shot-cut were used as separate examples, if they were of sufficient length. The use of manual extraction also allowed these sequences to be temporally localized down to the frame level, removing irrelevant data before and after the action. On average the resulting sequences were 2.5 seconds long.

Each sample from the dataset comprises a video from both the left and right viewpoint, both of which are 1920 by 1080 resolution and 24 frames per second. This amounts to over an hour of video data. In addition to the left and right videos, a reconstructed depth video (discussed in section 3.1.4) is provided for each sample, with the same resolution and frame rate. If necessary, it is possible to simulate a hybrid dataset, such as would be obtained from a Kinect, by ignoring the right viewpoint videos, and taking only the left viewpoint and the depth. However, such a dataset would posses a higher spatial resolution, and a coarser depth quantisation, than for true Kinect data. Any artefacts introduced into the depth data via post processing of the footage, are assumed to be negligible, as any significant changes would cause the actions to appear unnatural to the audience of the films.

Certain actions are more common in some films, and as such, are more prevalent in the dataset. If the task of action recognition is to be approached from a multi-class point of view, this imbalance may lead to biasing of the results. Thus, a "balanced" version of the dataset is also created, by duplicating randomly chosen samples from each class, until all have equal representation. This balanced dataset is used for the experiments in this thesis.

Film	*Run*	*Punch*	*Kick*	*Shoot*	*Eat*	*Drive*	*UsePhone*	*Kiss*	*Hug*	*StandUp*	*SitDown*	*Swim*	*Dance*
RE	11	1	5	41						4	1	6	
DA		4	1	25		47	3	2		2			1
FN	3				2		7	12	1	1	3		1
MBV	13	3				3	9	3	2	1			
PotC	6	1	3		1			2				7	
S	2	2			5		1	1	4	8	8	14	
SK	3	1	2		6	4		1			2		
SU	4		2			1		6	4	1			44
T	3		5		2	15	1		1	6	2		
A	9	2		7	1	10				2	2		
TM	6		1	7				6		4			2
FD	7	1	2		3		12	6	2	5	8	2	
H&K	3	4	1	6	2	13	1	1	1	3	1		3
U	7			11		5	7		2	6		4	

Table 3.2: Train and Test split for Films. Abbreviated film names are used for reasons of space. Empty entries indicate that no actions of that type were present within the film. Blue entries were part of the train set, and red entries part of the test set.

Film	*Run*	*Punch*	*Kick*	*Shoot*	*Eat*	*Drive*	*UsePhone*	*Kiss*	*Hug*	*StandUp*	*SitDown*	*Swim*	*Dance*	*NoAction*	Total
Train Samples	38	10	11	47	11	51	21	20	9	22	14	16	45	44	359
Test Samples	39	9	11	50	11	47	20	20	8	21	13	17	7	34	307

Table 3.3: Number Train and Test clips for each action in the dataset

Figure 3.1: Example frames from the Hollywood 3D dataset. Both the left view and depth streams are displayed. From top to bottom, examples are taken from the action classes *Hug*, *Kick*, *Kiss* and *Drive* and *Eat*. Darker regions of the depth images are closer to the camera.

3.1.4 Depth Data Generation

In order to examine the value of 3D information in action recognition, stereo reconstruction must be performed on the left and right viewpoint videos extracted in section 3.1.3, in order to obtain a structural estimate for the sequence. Due to the size of the Hollywood 3D dataset, many state of the art reconstruction algorithms would prove impractical. However, the GPU accelerated approach of Richardt *et al.* [101], allowed depth data to be generated in a tractable time-frame.

This scheme (referred to as Dual Cross Bilateral Grid (DCBG)) is inspired by bilateral filtering, which is a discontinuity preserving smoothing scheme, discussed in greater detail in section 5.1. To generate Hollywood 3D, the temporal variant was employed, which favours consistency of the reconstruction between frames, so as to remove flickering in the resulting sequences. Unfortunately, this scheme has high memory overheads and due to memory limitations on the GPU, results were restricted to 96 disparity levels. Using these settings the system was able to operate close to real time, and process the dataset in a matter of hours.

3.2 Holistic Action Recognition Using Structure

In order to determine the value of 3D information for action recognition, a range of traditional spatiotemporal action recognition techniques, are extended to account for structural information. Figure 3.2 shows the program flow traditionally used in such systems, beginning with the detection of salient spatiotemporal points, followed by the extraction of local feature descriptors, and finally the encoding of the sequence in a single holistic descriptor, which is then classified.

These approaches are less flexible than some alternatives (such as an iterative extension of the pose estimation system of appendix A), as it is difficult to perform online estimation, or to estimate the location of actions (both temporally and spatially). To balance this, the holistic accumulation provides improved robustness to translations and scaling, while the use of interest points leads to less noisy features.

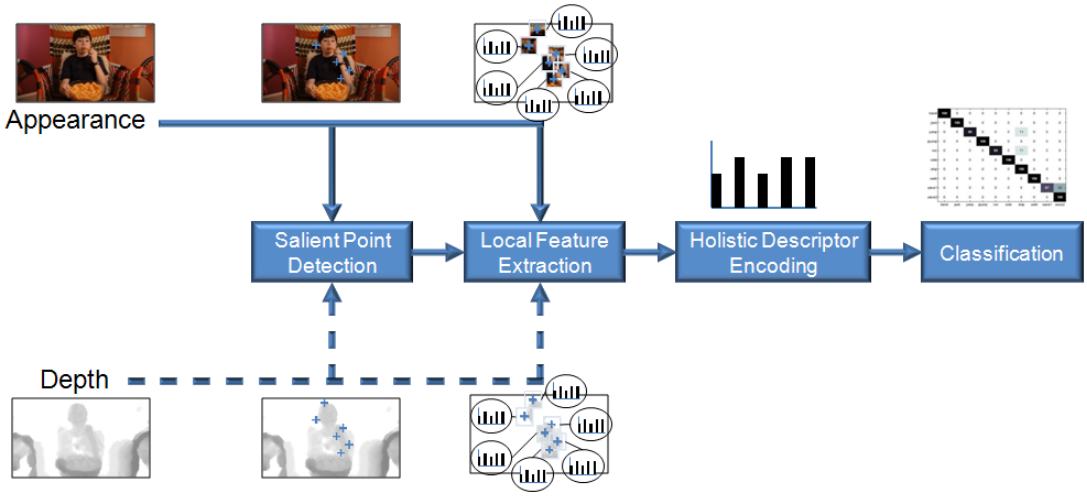


Figure 3.2: Traditional action recognition pipeline, including depth inputs. A number of salient points are detected and local feature descriptors are extracted for each point. These local descriptors are then accumulated into a holistic descriptor of the entire video which is used for classification.

The remainder of this chapter discusses 5 extensions to traditional saliency detectors, incorporating 3D information. In addition 2 popular feature descriptors for action recognition are discussed, along with their extensions to encode structural features.

3.2.1 Interest Point Detection in 4D

Performing dense sampling (as in the pose system of appendix A) generally leads to a significant number of features unrelated to the action under consideration, and containing little to no discriminative potential. In a temporal system this may be exacerbated, by irrelevant frames before and after the action. By estimating salient regions, irrelevant features may be removed. The saliency measures discussed below were originally formulated for static images, and were later extended to operate in the 3D (spatiotemporal) domain. Initially, a similar naive approach was followed, to extend the techniques from 3D to 4D (incorporating depth).

The specific spatiotemporal saliency measures discussed are the Harris Corner extension by Laptev and Lindeberg [71], the Hessian points approach of Willems *et al.* [134] and

the Separable Filters technique of Dollar *et al.* [45]. An in depth comparison of these spatiotemporal interest point detectors for traditional spatiotemporal tasks, is available in the survey paper by Tuytelaars and Mikolajczyk [121].

4D Harris Corners

The Harris Corner [62] is a very popular 2D saliency measure, which was extended into the spatiotemporal domain (3D) by Laptev *et al.* [71]. The detector operates on the spatiotemporal volume $\boldsymbol{I}_{1\ldots t}$, by first applying Gaussian smoothing (via convolution with the Gaussian function $g(0,\sigma)$), and then calculating the second-moment-matrix $\boldsymbol{H}_s(x,y,\tau)$ at each spatial location (x,y) and each temporal location τ. Salient points are defined as locations where $\boldsymbol{H}_s$ contains 3 large eigenvalues, i.e. there is strong intensity variation along 3 distinct spatiotemporal axes. In order to improve efficiency, and avoid calculation of eigenvalues at every location, maxima are instead detected in the volume $\boldsymbol{I}_{Ha}$ calculated using equation 3.1 (where k is typically 0.001).

$$\boldsymbol{I}_{Ha-3D}(x,y,\tau) = \det\left(\boldsymbol{H}_s(x,y,\tau)\right) - k\ \mathrm{trace}\left(\boldsymbol{H}_s(x,y,\tau)\right)^3 \tag{3.1}$$

It is possible to evaluate this equation without explicitly calculating the eigenvalues of the matrix, and all points with 3 large eigenvalues will also be maxima of $\boldsymbol{I}_{Ha}$ (however some maxima of $\boldsymbol{I}_{Ha}$ may not have 3 large eigenvalues).

Extending the operator into an additional dimension is trivial. The second-moment-matrix $\boldsymbol{H}_s$ is increased to include gradients along the new dimension z, as shown in equation 3.2 (where $\boldsymbol{I}_{\nabla x}$ is the gradients along the x dimension, and so on), and the power of the trace in equation 3.1 is increased to 4. For clarity the spatiotemporal index (x,y,τ) is omitted below, but a separate matrix $\boldsymbol{H}_s$ is calculated at each space-time location, using the local gradients for that location.

$$\boldsymbol{H}_s = g(0,\sigma) * \begin{pmatrix} \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla x} & \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla y} & \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla \tau} & \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla z} \\ \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla y} & \boldsymbol{I}_{\nabla y}\boldsymbol{I}_{\nabla y} & \boldsymbol{I}_{\nabla y}\boldsymbol{I}_{\nabla \tau} & \boldsymbol{I}_{\nabla y}\boldsymbol{I}_{\nabla z} \\ \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla \tau} & \boldsymbol{I}_{\nabla y}\boldsymbol{I}_{\nabla \tau} & \boldsymbol{I}_{\nabla \tau}\boldsymbol{I}_{\nabla \tau} & \boldsymbol{I}_{\nabla \tau}\boldsymbol{I}_{\nabla z} \\ \boldsymbol{I}_{\nabla x}\boldsymbol{I}_{\nabla z} & \boldsymbol{I}_{\nabla y}\boldsymbol{I}_{\nabla z} & \boldsymbol{I}_{\nabla \tau}\boldsymbol{I}_{\nabla z} & \boldsymbol{I}_{\nabla z}\boldsymbol{I}_{\nabla z} \end{pmatrix} \tag{3.2}$$

Unfortunately, the combination of appearance and depth streams provides 3.5D rather than 4D data. Unlike volumetric data such as MRI scans, the intensity measurements are not dense along the z dimension, instead the measurements form a 3D surface, within the 4D space. As a result of this, the gradient $\boldsymbol{I}_{\nabla z}$ cannot be estimated directly using convolution with a filter in the z dimension. However, the relationship between gradients of the depth and appearance streams, can be exploited using the chain rule, as in equation 3.3, where observation $\boldsymbol{I}_{a,\nabla x}$,$\boldsymbol{I}_{a,\nabla y}$,$\boldsymbol{I}_{a,\nabla \tau}$ are the appearance gradients along the spatial and temporal dimensions, while $\boldsymbol{I}_{d,\nabla x}$,$\boldsymbol{I}_{d,\nabla y}$,$\boldsymbol{I}_{d,\nabla \tau}$ are the gradients of the depth stream, along the same dimensions.

$$\boldsymbol{I}_{\nabla z} = \frac{\delta a}{\delta z} = \frac{\frac{\delta a}{\delta x}}{\frac{\delta z}{\delta x}} + \frac{\frac{\delta a}{\delta y}}{\frac{\delta z}{\delta y}} + \frac{\frac{\delta a}{\delta \tau}}{\frac{\delta z}{\delta \tau}} = \frac{\boldsymbol{I}_{a,\nabla x}}{\boldsymbol{I}_{d,\nabla x}} + \frac{\boldsymbol{I}_{a,\nabla y}}{\boldsymbol{I}_{d,\nabla y}} + \frac{\boldsymbol{I}_{a,\nabla \tau}}{\boldsymbol{I}_{d,\nabla \tau}} \tag{3.3}$$

Using these estimated gradients, the second-moment-matrix can be calculated at every location, and the set of 4D Harris interest points $F_{4D\text{-}Ha}$ is defined as the set of spatiotemporal locations within the sequence, for which the response value $\boldsymbol{I}_{Ha}$ is above a threshold $\lambda_{4D\text{-}Ha}$, as in equation 3.5. Lower threshold values lead to more information being extracted from the sequence, but increase the likelihood of noise being included. The effect of this threshold on recognition performance, is examined in detail in section 3.3.3.

$$F_{3D\text{-}Ha} = \{x, y, \tau \mid \boldsymbol{I}_{Ha-3D}\,(x, y, \tau) > \lambda_{3D\text{-}Ha}\} \tag{3.4}$$

$$F_{4D\text{-}Ha} = \{x, y, \tau \mid \boldsymbol{I}_{Ha}\,(x, y, \tau) > \lambda_{4D\text{-}Ha}\} \tag{3.5}$$

4D Hessian Points

In [134], Willems *et al.* extended the Beaudet Saliency Measure [22] into the spatiotemporal domain. Rather than the second-moment-matrix of Laptev *et al.* they calculated the Hessian $\boldsymbol{H}_{e-4D}$ of the spatiotemporal volume after Gaussian smoothing. The definition of saliency differs slightly, as the determinant of the Hessian is used directly,

implying that interest points must exhibit second order gradients along 3, roughly orthogonal, axes. This is in contrast to the Harris corner, where the orientations of the eigenvectors do not matter, as long as the eigenvalues are large enough. This highlights the origins of the Hessian points as a blob detector, however in the 3D extension the sign of the second order derivatives are not assured, leading to the detection of both spatiotemporal blobs, and saddles.

As in the 4D Harris scheme, gradients along z are estimated using the relationship between the depth and intensity stream gradients from equation 3.3. This allows the 4D Hessian $\boldsymbol{H}_{e-4D}$ to be calculated as in equation 3.6. The set of interest points $F_{4D\text{-}He}$ is then calculated as the set of spatiotemporal locations, for which the determinant of $\boldsymbol{H}_{e-4D}$ is greater than the threshold $\lambda_{4D\text{-}He}$ as in equation 3.8.

$$\boldsymbol{H}_{e-4D} = g\left(0, \sigma\right) * \begin{pmatrix} \boldsymbol{I}_{\nabla x \nabla x} & \boldsymbol{I}_{\nabla x \nabla y} & \boldsymbol{I}_{\nabla x \nabla \tau} & \boldsymbol{I}_{\nabla x \nabla z} \\ \boldsymbol{I}_{\nabla x \nabla y} & \boldsymbol{I}_{\nabla y \nabla y} & \boldsymbol{I}_{\nabla y \nabla \tau} & \boldsymbol{I}_{\nabla y \nabla z} \\ \boldsymbol{I}_{\nabla x \nabla \tau} & \boldsymbol{I}_{\nabla y \nabla \tau} & \boldsymbol{I}_{\nabla \tau \nabla \tau} & \boldsymbol{I}_{\nabla \tau \nabla z} \\ \boldsymbol{I}_{\nabla x \nabla z} & \boldsymbol{I}_{\nabla y \nabla z} & \boldsymbol{I}_{\nabla \tau \nabla z} & I_{\nabla z \nabla z} \end{pmatrix} \tag{3.6}$$

$$F_{3D\text{-}He} = \left\{x, y, \tau \mid \det\left(\boldsymbol{H}_{e-3D}\left(x, y, \tau\right)\right) > \lambda_{3D\text{-}He}\right\} \tag{3.7}$$

$$F_{4D\text{-}He} = \left\{x, y, \tau \mid \det\left(\boldsymbol{H}_{e-4D}\left(x, y, \tau\right)\right) > \lambda_{4D\text{-}He}\right\} \tag{3.8}$$

3.2.2 Interest Point Detection 3.5D

In part, the original Harris and Hessian interest point operators were motivated by the idea that object boundary points are highly salient, and that intensity gradients generally relate to boundaries. However, depth data directly provides boundary information, rendering the estimation of the intensity gradient along z somewhat redundant. An alternative approach would be, instead of estimating a full 4D intensity volume, to employ a "3.5D" representation, using a complimentary pair of 3D spatiotemporal volumes, from the appearance and depth sequences. This makes it possible to detect object

boundaries against similarly colour backgrounds using depth, while also detecting the boundary between objects of similar depth, using appearance.

3.5D Harris Corners

Equation 3.9 defines the set of 3.5D Harris points $F_{3.5D\text{-}Ha}$ where $\boldsymbol{I}_{Ha,a}$ is the volume created by applying equation 3.1 to the appearance stream, while $\boldsymbol{I}_{Ha,d}$ is the converted depth volume. The relative weighting of the appearance and depth information, is controlled by α, which is set to 1 for the remainder of the thesis, implying equal weighting of appearance and depth data.

$$F_{3.5D\text{-}Ha} = \{x, y, \tau \mid \boldsymbol{I}_{Ha,a}(x, y, \tau) + \alpha \boldsymbol{I}_{Ha,d}(x, y, \tau) > \lambda_{3.5D\text{-}Ha}\} \tag{3.9}$$

The salient points detected by this scheme are those with either strong intensity gradients in 3 spatiotemporal directions (i.e. the meeting point of 2 moving, high contrast, edges) or with a strong depth discontinuity along 3 spatiotemporal directions (i.e. a moving object corner).

3.5D Hessian Points

Equation 3.10 relates to the set of 3.5D Hessian points $F_{3.5D\text{-}He}$, specified in terms of the determinants of $\boldsymbol{H}_{e-4D,a}$ and $\boldsymbol{H}_{e-4D,d}$ which are the 3D Hessian matrices of the appearance and depth volumes respectively.

$$F_{3.5D\text{-}He} = \{x, y, \tau \mid \det(\boldsymbol{H}_{e-4D,a}(x, y, \tau)) + \alpha \det(\boldsymbol{H}_{e-4D,d}(x, y, \tau)) > \lambda_{3.5D\text{-}He}\} \tag{3.10}$$

As mentioned previously, Hessian points relate to image regions with second order derivatives along 3 near-orthogonal axes. Thus in the 3.5D case, detected points relate to moving regions with either concave/convex structures, or blob/saddle texturing.

3.5D Separable Filters

A third highly successful approach to interest point detection, is the Separable Linear Filters technique of Dollar *et al.* [45]. Unlike the Harris and Hessian approaches, each dimension is treated differently in the separable filters approach. This makes the approach less suitable for a direct 4D extension, however the 3.5D technique can be easily applied.

The original formulation for Separable Filters is to create a spatiotemporal response volume $\boldsymbol{I}_{SF}$, by filtering the input volume, using a 2D Gaussian filter in the spatial dimensions, and a quadrature pair of Gabor filters g_{eve} and g_{odd} along the temporal dimension, as shown in equation 3.11.

$$\boldsymbol{I}_{SF} = (\boldsymbol{I} * g(0,\sigma) * g_{eve})^2 + (\boldsymbol{I} * g(0,\sigma) * g_{odd})^2 \tag{3.11}$$

The even and odd Gabor filters which are applied along the temporal dimension, relate to single cycles of a sin and cosine wave, respectively. Thus the interest point detector targets regions with low intensity, followed by an intensity peak, or regions with an intensity peak surrounded by 2 low intensity points. Such points would relate to the moving edge of two high contrast regions, or a single high contrast line, in motion. The Gaussian filtering applied in the spatial domain, simply relates to the weighted accumulation of responses over a region.

Equation 3.13 shows the 3.5D formulation of the separable filters approach. The response volume created by equation 3.11 when applied to the appearance and depth streams, are denoted as $\boldsymbol{I}_{SF,a}$ and $\boldsymbol{I}_{SF,d}$ respectively. The set of detections $F_{3.5D\text{-}S}$ relates to moving high contrast boundaries as described above, or to a moving structural boundary.

$$F_{3D\text{-}S} = \{x, y, \tau \mid \boldsymbol{I}_{SF}(x, y, \tau) > \lambda_{3D\text{-}S}\} \tag{3.12}$$

$$F_{3.5D\text{-}S} = \{x, y, \tau \mid \boldsymbol{I}_{SF,a}(x, y, \tau) + \alpha \boldsymbol{I}_{SF,d}(x, y, \tau) > \lambda_{3.5D\text{-}S}\} \tag{3.13}$$

3.2.3 Feature Descriptors

These interest point schemes help to reduce the contamination of the features with irrelevant information. By utilising depth data it may be possible to more robustly identify salient regions. However the features themselves do not encode any structural information, which is likely to be discriminatory. Below, two highly successful feature descriptors are discussed. In their original formulation, these descriptors include only 3D spatiotemporal information, and so extensions are proposed, to allow the descriptors to incorporate structural information.

Relative Motion Descriptor

The Relative Motion Descriptor (RMD) of Oshin *et al.* [93] has been shown to perform well in a large range of natural action recognition situations, while making use of only the saliency information obtained during interest point detection. To calculate the descriptor, a spatiotemporal volume $\boldsymbol{I}_{int}$ is created using the interest point detector response strengths. In equation 3.14 the saliency content $\boldsymbol{I}_{cont}(x, y, \tau)$ is defined, for a spatiotemporal region with origin (x, y, τ) and dimensions $(\hat{x}, \hat{y}, \hat{\tau})$. This summation over a region can be calculated rapidly by creating an integral volume, and using only a small number of lookups.

The descriptor $\boldsymbol{I}_{rmd}$ of the saliency distribution at location (x, y, τ) can then be formed, by performing O comparisons, between pairs of randomly offset spatiotemporal subcuboids, with origins at $(x, y, \tau)+\boldsymbol{\eta}$ and $(x, y, \tau)+\boldsymbol{\eta}'$. Figure 3.3 illustrates the relationship between the location in question, the random offsets and the associated subcuboids. Also note that the collections of offsets $\boldsymbol{\eta}_{1..O}$ and $\boldsymbol{\eta}'_{1..O}$ is randomly selected before training. The same set is then applied for all subsequent training and test sequences, so that the same subcuboids are always compared.

$$\boldsymbol{I}_{cont}(x, y, \tau) = \sum_{\hat{\boldsymbol{\eta}}=(0,0,0)}^{(\hat{x}, \hat{y}, \hat{\tau})} \boldsymbol{I}_{int}([x, y, \tau] + \hat{\boldsymbol{\eta}}) \tag{3.14}$$

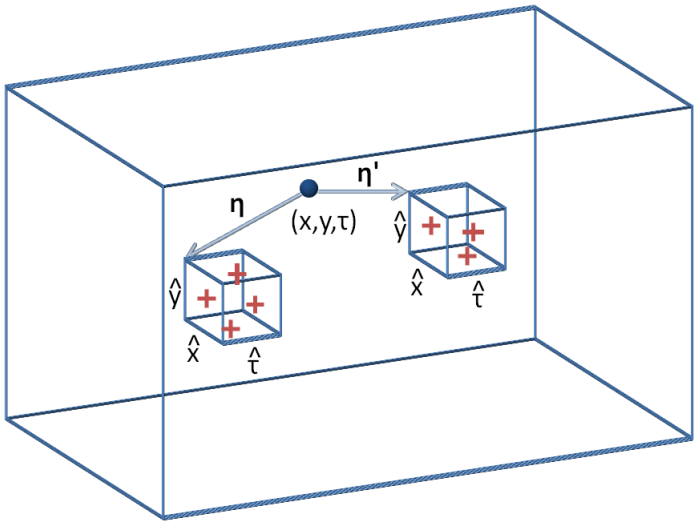


Figure 3.3: An RMD saliency content comparison. A single saliency content comparison, within the spatiotemporal volume $\boldsymbol{I}_{int}$. The comparison is being performed at location (x, y, τ) using offsets $\boldsymbol{\eta}$ and $\boldsymbol{\eta}'$. The subcuboids have dimensions of $\hat{x}, \hat{y}, \hat{\tau}$, and contain 4 and 3 interest points respectively.

$$\boldsymbol{I}_{rmd}(x, y, \tau) = \sum_{o=0}^{O} \begin{cases} 2^o & \boldsymbol{I}_{cont}\left([x, y, \tau] + \boldsymbol{\eta}_o\right) > \boldsymbol{I}_{cont}\left([x, y, \tau] + \boldsymbol{\eta}'_o\right) \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

Continuing the example from figure 3.3, the subcuboids at offsets $\boldsymbol{\eta}$ and $\boldsymbol{\eta}'$ contain 4, and 3, interest points, respectively. If all interest points have a weight of 1, then $\boldsymbol{I}_{cont}\left([x, y, \tau] + \boldsymbol{\eta}_o\right) = 4$ and $\boldsymbol{I}_{cont}\left([x, y, \tau] + \boldsymbol{\eta}'_o\right) = 3$. This means the comparison will succeed, and the descriptor $\boldsymbol{I}_{rmd}(x, y, \tau)$ will be incremented by 2^o.

The response $\boldsymbol{I}_{rmd}$ can then be computed at every spatiotemporal location in the sequence, and accumulated into a histogram, which encodes how often relative saliency distributions occur within the sequence. The descriptor is applicable to any form of saliency measure, and does not require appearance or motion data to be available. The amount of information encoded by the descriptor is controlled by the number of comparisons O. However increasing this value also leads to sparser histograms. To cope with this, it is common to concatenate several $\boldsymbol{I}_{rmd}$ histograms, each calculated using different collections of random offsets. This allows the descriptor to encode more

information, without becoming sparse, by sacrificing the independence of the bins (i.e. some of the encoded information may be redundant).

Relative Motion Descriptor - 4D

The RMD descriptor can be extended, to incorporate 3D structural information, by recording the z measurements of each interest point detection. A 4 dimensional integral hyper-volume $\boldsymbol{I}_{int\text{-}4d}$ can then be populated, which contains the behaviour of the saliency distribution, within the 3D scene, rather than on the image plane. The RMD-4D descriptor $\boldsymbol{I}_{rmd\text{-}4d}$ can then be computed, as in equation 3.17, by comparing the saliency content of randomly offset sub-hypercuboids.

$$\boldsymbol{I}_{cont\text{-}4D}\left(x,y,z,\tau\right) = \sum_{\hat{\boldsymbol{\eta}}=0}^{(\hat{x},\hat{y},\hat{z},\hat{\tau})} \boldsymbol{I}_{int}\left([x,y,z,\tau]+\hat{\boldsymbol{\eta}}\right) \tag{3.16}$$

$$\boldsymbol{I}_{rmd\text{-}4d}(x,y,z,\tau) = \sum_{o=0}^{O} \begin{cases} 2^o & \boldsymbol{I}_{cont\text{-}4D}\left([x,y,z,\tau]+\boldsymbol{\eta}_o\right) > \boldsymbol{I}_{cont\text{-}4D}\left([x,y,z,\tau]+\boldsymbol{\eta'}_o\right) \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

As with the original RMD formulation, this new descriptor can operate in conjunction with any definition of saliency (i.e. using any interest point detection scheme). Importantly, the RMD-4D descriptor is not restricted to operating with the depth aware interest point detectors of section 3.2.1. The feature descriptor is equally applicable to standard spatiotemporal interest points, provided a depth stream is available, to look up corresponding z values.

Bag of Visual Words

A more popular approach to feature extraction for action recognition, is the Bag of Visual Words (BoVW) technique, employed by Laptev *et al.* [72]. This is inspired by similar, highly successful work in the field of object recognition. In brief, a collection of distinctive "exemplar" features are selected from the training data. These are the

"words", which collectively form the codebook. Sequences are then described by a frequency histogram (the "bag" of words), with each bin relating to one word from the codebook. Any feature space may be used to generate the words (in object recognition SIFT points are commonly used). One highly successful approach, in the field of action recognition, is to employ a combined HOG and HOF feature vector. Unlike the RMD, this descriptor directly encodes both appearance and motion information, making it more descriptive, but somewhat less flexible.

Equation 3.18 defines the descriptor ρ at location (x, y, τ) in the sequence, where $\omega\left(\boldsymbol{I}_{\triangledown x}, \boldsymbol{I}_{\triangledown y}, (x, y, \tau)\right)$ is a function, which creates a joint histogram of the values taken from $\boldsymbol{I}_{\triangledown x}, \boldsymbol{I}_{\triangledown y}$ within a local region centred on point (x, y,τ). When the function ω is applied to gradient images, the output histogram relates to a HOG descriptor, while the output for flow images ($\boldsymbol{I}_{\frac{dx}{d\tau}}$ and $\boldsymbol{I}_{\frac{dy}{d\tau}}$) is a HOF descriptor. Some spatial information is maintained, by splitting the neighbourhood into blocks, and concatenating each blocks histogram.

$$\rho(x, y, \tau) = \left(\omega\left(\boldsymbol{I}_{\triangledown x}, \boldsymbol{I}_{\triangledown y}, [x, y, \tau]\right), \omega\left(\boldsymbol{I}_{\frac{dx}{d\tau}}, \boldsymbol{I}_{\frac{dy}{d\tau}}, [x, y, \tau]\right)\right) \tag{3.18}$$

After ρ has been calculated for all interest points in the training sequences, clustering is performed to find a distinctive subset of descriptors which covers the range of features observed, in order to form the codebook. In this thesis, K-Means clustering is employed, with a standard euclidean distance function as in [72]. When the contents of the codebook have been determined, the training sequences (and any subsequent sequences) may be encoded, by assigning each extracted ρ to the closest cluster in the codebook. These assignments are then accumulated into the histogram describing the sequence as a whole.

For each interest point, a single ρ is extracted, which equates to a single entry in the histogram. This may prove more robust than the RMD approach, where saliency comparisons are performed at every point within the sequence, as irrelevant interest points will affect one histogram entry, rather than many. This idea is explored in section 3.3.3.

Bag of Visual Words - 4D

To extend ρ to encode structural information, the simplest approach is to include a Histogram of Oriented Depth Gradients (HODG) structure descriptor, alongside the appearance and motion HOG/HOFs. This is shown in equation 3.19. It is natural to ask "why not do the equivalent for the HOF information in the depth stream", however standard optical flow algorithms tend to yield poor results when applied to depth videos, and performing accurate 3D motion estimation is a non trivial task. The estimation of such 3D motion information is explored in depth in the following chapters.

$$\rho_{4D}(x,y,\tau) = \left(\omega\left(\boldsymbol{I}_{a,\nabla x}, \boldsymbol{I}_{a,\nabla y}, [x,y,\tau]\right), \omega\left(\boldsymbol{I}_{\frac{dx}{d\tau}}, \boldsymbol{I}_{\frac{dy}{d\tau}}, [x,y,\tau]\right), \omega\left(\boldsymbol{I}_{d,\nabla x}, \boldsymbol{I}_{d,\nabla y}, [x,y,\tau]\right)\right) \tag{3.19}$$

The same bag of words encoding scheme is employed for ρ_{4D} features as for the ρ features of the previous section. As with the RMD-4D algorithm, ρ_{4D} is not dependent on the interest point detector used, and specifically does not require depth aware interest points, however a depth video is necessary for calculating the HODG.

3.3 Experiments

Classification of sequence descriptors was performed using a Support Vector Machine (SVM), with a Radial Basis Function (RBF) kernel. To facilitate comparisons with results on the Hollywood 1 and Hollywood 2 datasets, the Mean Average Precision (MAP) was calculated as an error metric. The metric is explained in detail in the PASCAL VOC [47], but essentially relates to the area under the precision recall curve i.e. the "cleanness" of results, across all detection thresholds. Due to this integration over recalls, the error measure is extremely robust, even without cross validation. This allows a specific train and test set to be defined as in the Hollywood 1 and 2 datasets, which improves reliability of comparisons with future work.

For the computation of RMD features, 4 region comparisons were performed per histogram (O=4) and 10 histograms were concatenated to form the descriptor. The bag

of visual words descriptor was generated using a codebook of 4,000 elements, as used by Laptev *et al.* [72].

3.3.1 Interest Point Evaluation

First the incorporation of 3D structural information during interest point detection is evaluated. The analysis aims to determine whether points chosen as salient by each scheme, are more or less relevant to the actions being performed. For this analysis, the Bag of Visual Words feature descriptor of Laptev *et al.* [72] was employed. Table 3.4 shows the performance breakdown per class for each interest point detection scheme.

The type of saliency measure used has a surprisingly large effect on the eventual performance of the system, with the average precision of the best scheme being roughly double that of the worst, even using the same feature encoding. For the standard spatiotemporal schemes, Harris points ($F_{3D\text{-}Ha}$) outperform separable filter points ($F_{3D\text{-}S}$) for all actions. This is also reflected in the depth aware schemes, and is unsurprising, as separable filters were designed primarily for computational speed. Hessian based interest points prove less informative than the Harris operators in both the 4D and 3.5D case. For all detectors, the 4D scheme outperforms its standard spatiotemporal counterpart, however the 3.5D approaches prove the most informative. This confirms the belief that the calculation of intensity gradients along z is redundant, and that the combination of intensity and structure gradients, is a stronger measure of saliency.

Interestingly, certain actions consistently perform better, when described by depth aware interest points. These are actions such as *Kiss*, *Hug*, *Drive* and *Run* where there is an informative foreground object, which depth aware interest points are better able to pick out. In contrast, actions such as *Swim*, *Dance* and *Shoot* are often performed against a similar depth background, or within a group of people, and the inclusion of depth in the saliency measure is less valuable. For certain actions such as *Punch* and *Kick* the original spatiotemporal formulation actually offers the best performance. This suggests that a combination of standard spatiotemporal, and depth aware schemes, may prove valuable.

The complexity of the depth aware interest point detectors remains of the same order

Action	$F_{3D\text{-}S}$	$F_{3D\text{-}Ha}$	$F_{4D\text{-}He}$	$F_{4D\text{-}Ha}$	$F_{3.5D\text{-}S}$	$F_{3.5D\text{-}He}$	$F_{3.5D\text{-}Ha}$
NoAction	11.4	12.1	**12.2**	**12.9**	11.4	12.0	**13.7**
Run	12.6	19.0	15.9	**22.4**	12.7	**21.8**	**27.0**
Punch	2.9	**10.4**	2.9	4.8	2.9	5.7	5.7
Kick	3.6	**9.3**	4.2	4.3	3.8	3.7	4.8
Shoot	16.2	**27.9**	18.9	17.2	16.2	16.2	16.6
Eat	3.6	5.0	3.6	**5.3**	3.6	**7.7**	**5.6**
Drive	15.3	24.8	**25.6**	**69.3**	15.5	**76.5**	**69.6**
UsePhone	6.5	6.8	**14.7**	**8.0**	6.5	**17.7**	**7.6**
Kiss	6.5	8.4	**8.5**	**10.0**	6.5	**9.4**	**10.2**
Hug	2.6	4.3	3.5	**4.4**	2.6	3.4	**12.1**
StandUp	6.8	**10.1**	7.0	7.6	6.9	9.1	9.0
SitDown	4.2	5.3	4.5	4.2	4.2	4.3	**5.6**
Swim	5.5	**11.3**	7.8	5.5	5.5	5.9	7.5
Dance	2.3	10.1	4.2	**10.5**	2.2	3.8	7.5
Overall	7.1	12.6	9.8	**13.3**	7.1	**13.4**	**14.1**

Table 3.4: Interest Point detector performance. The displayed values are the average precision, for each class in the Hollywood 3D dataset. A range of interest point detectors are tested, including simple spatiotemporal interest points, and depth aware schemes. The Bag of Visual Words feature descriptor was used. Classes are shown in bold, when depth aware interest points outperform both 3D schemes.

as their spatio-temporal counterparts (linear with respect to x, y and τ). Naturally the multiplicative factor is increased however, with 3.5D techniques being roughly twice as costly, and 4D techniques taking 4 times as long.

3.3.2 Feature Evaluation

In this section, the value of including 3D structure information during feature encoding is analysed. Due to the quantity of results, only the average performance across classes is shown, however the correct classification rate is presented in addition to the average precision. This performance measure is more relevant in tasks where the sequence is known to belong to one and only one class (such as video clustering and categorisation). In table 3.5 each combination of interest point and feature descriptor is contrasted, with the best performing feature for each saliency measure shown in bold.

The previously noted relationship between saliency measures, appears to hold regardless of the feature descriptor used. In all cases the fast $F_{3D\text{-}S}$ and $F_{3.5D\text{-}S}$ points, performed the worst, followed by the Hessian based schemes, while the extended Harris operators provided the best performance. Also following the previously noted trend, spatiotemporal interest points offer the worst performance overall, while the 3.5D scheme prove to be the most effective way to incorporate depth information.

Both types of descriptor show a consistent improvement when incorporating structural information, with increases of roughly 30% in both average precision and correct classification. Overall, the Bag of Words descriptors slightly outperform the RMD descriptors. This is unsurprising as the RMD relies only on interest point detections, without the inclusion of the visual and motion information that is employed for the bag of words.

It would have been reasonable to assume, that including structural features would prove more valuable with a standard saliency measure, as the depth information had not previously been exploited. In fact the opposite proves to be true, 4D features provide more modest gains for **3D-S** and **3D-Ha** (up to 20%) than they do when combined with extended saliency measures (up to 45%). This demonstrates that depth aware saliency measures are capable of focusing computation, within regions where structural features are especially valuable.

Interest Points	Descriptor	CC Rate	AP
$F_{3D\text{-}S}$	RMD	7.2%	7.2
$F_{3D\text{-}S}$	RMD-4D	**7.3**%	**7.4**
$F_{3D\text{-}S}$	HoG/Hof	7.2%	7.1
$F_{3D\text{-}S}$	HoG/Hof/HoDG	**7.3**%	7.2
$F_{3D\text{-}Ha}$	RMD	15.3%	12.2
$F_{3D\text{-}Ha}$	RMD-4D	15.9%	**15.0**
$F_{3D\text{-}Ha}$	HoG/Hof	16.2%	12.6
$F_{3D\text{-}Ha}$	HoG/Hof/HoDG	**19.8**%	13.2
$F_{4D\text{-}He}$	RMD	10.4%	11.2
$F_{4D\text{-}He}$	RMD-4D	**16.2**%	**12.7**
$F_{4D\text{-}He}$	HoG/Hof	9.4%	9.3
$F_{4D\text{-}He}$	HoG/Hof/HoDG	11.4%	9.8
$F_{4D\text{-}Ha}$	RMD	10.7%	11.5
$F_{4D\text{-}Ha}$	RMD-4D	10.4%	10.3
$F_{4D\text{-}Ha}$	HoG/Hof	14.3%	12.5
$F_{4D\text{-}Ha}$	HoG/Hof/HoDG	**18.5**%	**13.3**
$F_{3.5D\text{-}S}$	RMD	7.3%	7.3
$F_{3.5D\text{-}S}$	RMD-4D	**7.6**%	**7.8**
$F_{3.5D\text{-}S}$	HoG/Hof	7.2%	7.1
$F_{3.5D\text{-}S}$	HoG/Hof/HoDG	7.3%	7.4
$F_{3.5D\text{-}He}$	RMD	13.3%	12.2
$F_{3.5D\text{-}He}$	RMD-4D	17.5%	**14.3**
$F_{3.5D\text{-}He}$	HoG/Hof	13.6%	11.7
$F_{3.5D\text{-}He}$	HoG/Hof/HoDG	**19.2**%	13.4
$F_{3.5D\text{-}Ha}$	RMD	12.3%	11.9
$F_{3.5D\text{-}Ha}$	RMD-4D	17.2%	**14.4**
$F_{3.5D\text{-}Ha}$	HoG/Hof	17.9%	13.0
$F_{3.5D\text{-}Ha}$	HoG/Hof/HoDG	**21.8**%	14.1

Table 3.5: Descriptor and Saliency performance. For each combination of descriptor and saliency measure, the Correct Classification rate and Average Precision is shown. The best feature for each saliency measure is shown in bold.

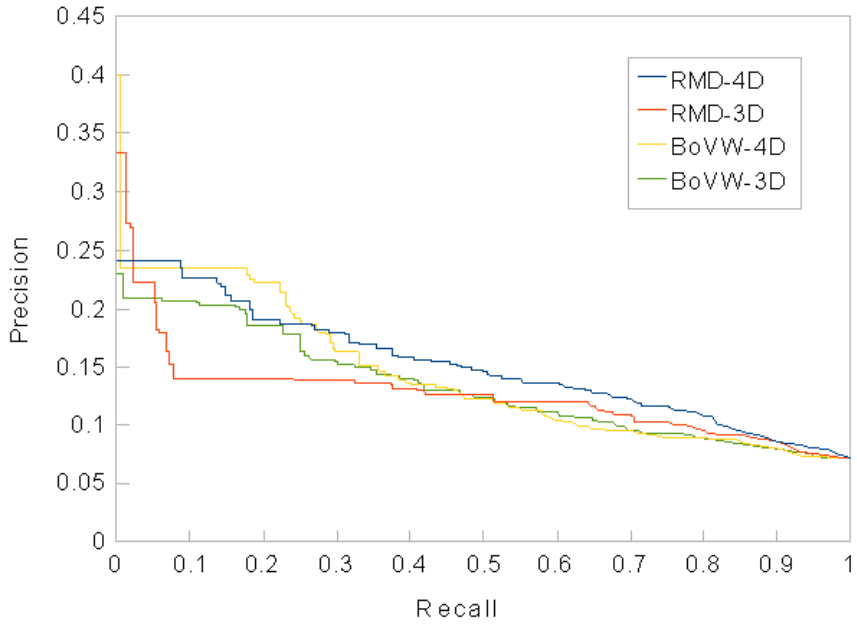


Figure 3.4: Precision-recall curves. One curve for each feature type, using the $F_{3.5D\text{-}Ha}$ saliency measure.

Figure 3.4 shows the precision-recall curves for each feature descriptor when using the $F_{3.5D\text{-}Ha}$ saliency measure. The bag of words approaches tend to offer the greatest precision at lower recall levels, while at higher recall levels the features converge to a similar level of precision. The 4D feature descriptors offer better precisions than their spatiotemporal counterparts for almost all recalls.

The complexity of the RMD-4D is greater than the standard RMD (being linear in the range of depth values, as well as in x, y and τ). This is somewhat mitigated by the use of integral volumes however, meaning that runtimes are still on the order of seconds using a single CPU. In contrast the extraction of HoDG features relates to a 50% increase in cost, while still remaining linear. However the increased feature vector length does lead to and increased cost during codebook generation, as k-means is generally polynomial in the number of dimensions.

3.3.3 Interest Point Threshold Evaluation

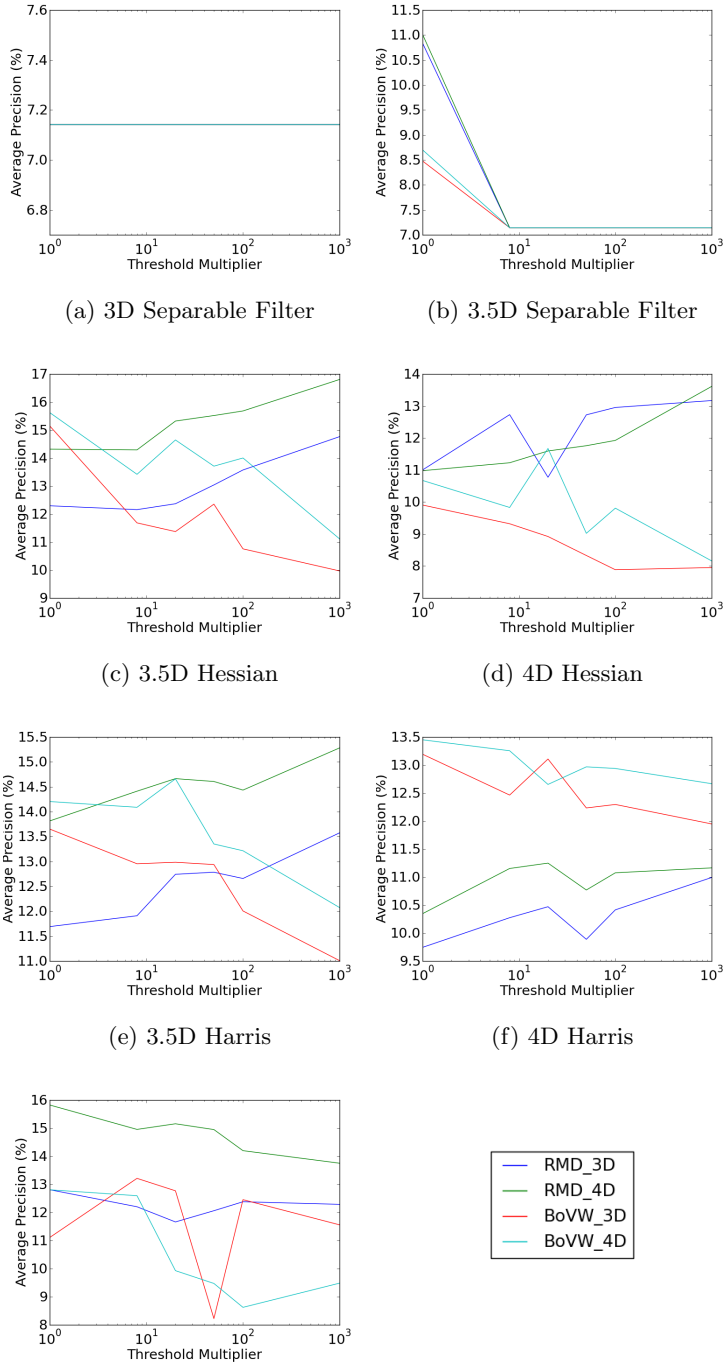
To generate the Precision-Recall curve of figure 3.4, the detection threshold of the SVM is varied. However, the interest point detection stage also involves a separate threshold, which controls the density of features, and the chance of spurious regions being included in the features. Different interest point operators produce very different response strengths, meaning the optimal threshold for extracting salient points varies between techniques. In most previous work an arbitrary threshold is selected (or empirically determined), indeed the experiments in the previous sections employed saliency thresholds based on those suggested by previous literature. In figure 3.5 the relationship between the saliency threshold and the action recognition performance, is contrasted for all interest point detectors.

The separable filter saliency responses appear to lie within a much tighter range than the other detectors. As a result, changing the threshold by more than an order of magnitude, leads to so few interest points that performance drops to chance.

Regardless of the saliency measure, both the standard feature descriptor and its depth aware extension, exhibit the same behaviour. Interestingly, the trend is positive for RMD based features, i.e. higher saliency thresholds lead to increased accuracy (except for the 3D Harris interest points). In contrast, bag of words approaches provide the greatest accuracy at lower saliency thresholds. This suggests that RMD features are more sensitive to noise, while the Bag of Words features are better at isolating non-discriminatory information. This makes sense, as a weak interest point relates to only a single histogram entry under the bag of words scheme. In contrast, poor interest points will affect the RMD descriptor at all surrounding locations. This is particularly interesting, as it demonstrates that interest point detection is valuable, not only for reducing the computation required, but also for improving the signal to noise ratio of the features. Denser features do not always lead to improved performance.

3.3.4 Detection modes

Figure 3.6 contrasts the Average Precision of the various interest point and feature combinations, when calculated under various detection schemes. In figure 3.6.a static



(a) 3D Separable Filter

(b) 3.5D Separable Filter

(c) 3.5D Hessian

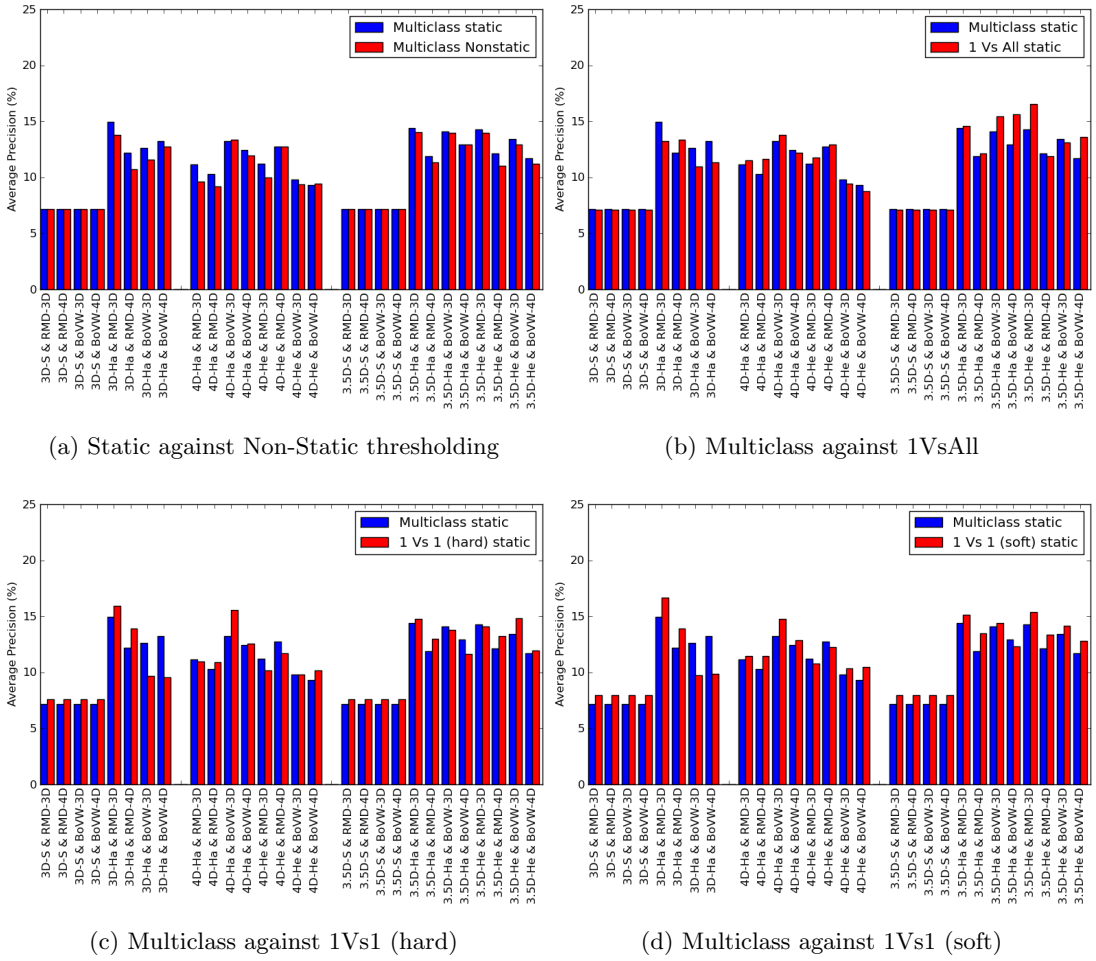
(d) 4D Hessian

(e) 3.5D Harris

(f) 4D Harris

(g) 3D Harris

Figure 3.5: Saliency threshold behaviour. The Average Precision of each interest point detector and descriptor combination, is shown as the saliency threshold is varied.



(a) Static against Non-Static thresholding

(b) Multiclass against 1VsAll

(c) Multiclass against 1Vs1 (hard)

(d) Multiclass against 1Vs1 (soft)

Figure 3.6: Analysis modes. Average precision across classes for the Hollywood 3D dataset, analysed with 4 different classification modes and 2 thresholding schemes.

mode, relates to a threshold on the class's likelihood, in order for the class to be detected. In contrast, non-static mode requires the class's likelihood, to be greater than the likelihood of the *NoAction* class, by a certain amount. The best performance, for all combinations of interest point and feature type, is obtained using the simpler static thresholding, likely due to the difficulties in learning the *NoAction* class.

The remaining detection modes are analysed in figures 3.6.b to 3.6.d, with a standard multiclass formulation, being compared to a bank of "1 versus all" classifiers and to a pair of "1 versus 1" schemes. The "hard" 1 versus 1 scheme, shown in figure 3.6.c restricts each classifier, to selecting a single class to vote for. The "soft" 1 versus 1

scheme shown in figure 3.6.d allows each classifier to cast a weighted vote for all classes, using their estimated likelihood.

Overall the best performance is provided by the "soft" 1 versus 1 voting scheme, as it breaks the problem down into simpler sub tasks without unnecessary quantisations on the probabilities. However, this is also the least scalable approach, with both training and testing times increasing quadratically with the number of classes (but not the number of samples).

3.4 Conclusions

The high intra class variation inherent in natural action recognition, can be somewhat mitigated by making use of 3D information, which removes variations between actors appearance, lighting and gives useful information about occlusions. To demonstrate this a new natural action recognition dataset was compiled, and a number of existing techniques, were shown to provide improved performance when extended to incorporate depth information. These included 5 different saliency operators, and 2 feature descriptors.

Gains in performance were most significant when both the saliency measure and the features included depth (up to a gain of 30% average precision), which demonstrates that the points detected as salient by the new operators contain especially descriptive structural features. The most effective measure of saliency for action recognition proved to be the Harris operator, with complimentary appearance and structural saliency (3.5D) shown to be the best extension.

The bag of words feature descriptors also proved to be the most valuable. However, it was also shown that increases to the saliency thresholding leads to improved performance for RMD descriptors, and reduced performance for bag of words, implying that the RMD scheme was more sensitive to noisy detections. This also served to highlight the fact that dense features are not always better, and that comparisons performed with different arbitrarily selected saliency thresholds are likely to draw different conclusions.

The dataset has been released to the community, facilitating future development of

more complex structural encoding for action recognition. In the future, an investigation into scale selection methods for the high-dimensional interest point detectors, would allow even greater performance improvements, particularly on high resolution data. Additionally, the encoding of depth into the descriptors may be developed. Structural information has been directly incorporated, but the motion features still only make use of movement within the image plane, despite the fact that estimation of out-of-plane motions should be possible.

Chapter 4

Particle Based 3D Motion Estimation

In addition to direct use of the structural data available from the Hollywood 3D dataset, it may be possible to enhance the motion information, to account for out of plane motion. Such motions are often discriminatory for actions, particularly in unconstrained scenarios. In order to integrate such information with the interest point based approach of chapter 3, it must be possible to obtain an instantaneous 3D motion estimate, at any space-time location deemed salient 3.2.1, obviously this is easiest if the estimated motion field is dense. Unfortunately, due to the size of the dataset, existing techniques for estimating dense 3D motion (also known as "scene flow") are infeasible. As an example, recent approaches [64, 99] often require several hours to process a single frame, and would thus take more than 20 years, for the whole dataset.

In addition to (and as a consequence of) these speed issues, existing techniques for calculating scene flow rarely consider estimation over more than a single frame. Thus, the formulations do not allow for the propagation of information between frames, leading to no enforcement of temporal consistency, and producing "flickering" of the resulting motion field. The instability of these motion fields makes them less reliable for recognising actions.

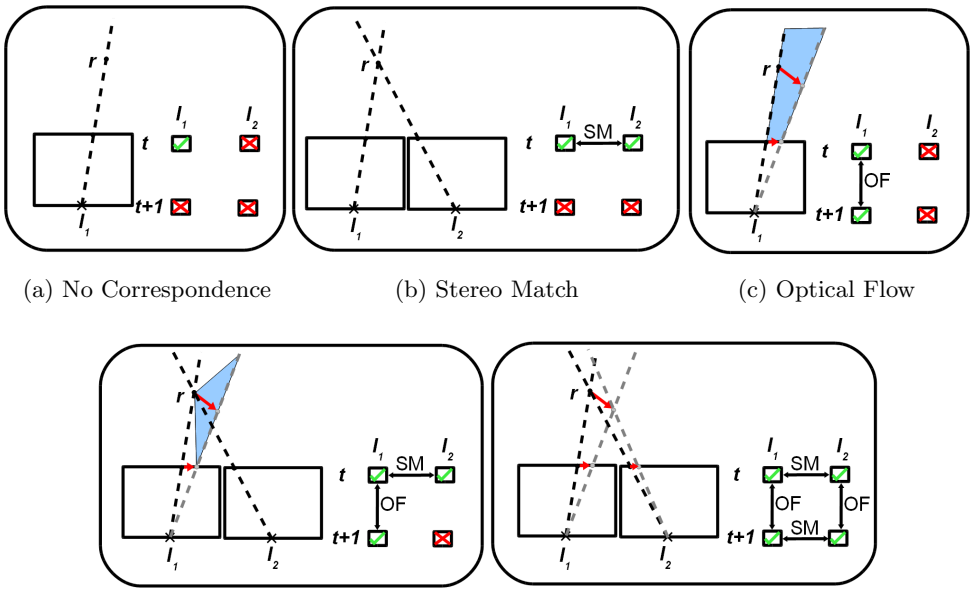
A third difficulty with current 3D motion estimation techniques, as discussed in sec-

tion 2.2.2, is that the ambiguity inherent in the task (particularly due to the aperture problem) has lead previous work to focus on the accumulation of constraints spatially to increase the aperture by assuming smoothness of the motion field (either within a local neighbourhood, or using a global variational optimisation). In practice, this means that, from the set of ambiguous motions, a single result is chosen, which gives rise to the smoothest motion field. However, at discontinuities in the motion field, such as object boundaries, the assumption often leads to the selection of sub-optimal motions, in favour of preserving smoothness. Places where this occurs are referred to as over-smoothing artefacts, and it was noted by Basha *et al.* that these artefacts are the primary source of errors in modern Scene Flow Estimation (SFE) techniques [19]. In the context of the action recognition system from chapter 3, these artefacts are especially damaging, as they corrupt motion at object boundaries, where the vast majority of salient points are detected.

A brief introduction to the problem of SFE is presented below. The remainder of this chapter then explores a novel probabilistic approach to SFE which was published in ICCV 2011[59], where temporal consistency is respected by propagating information between frames, and oversmoothing artefacts are avoided through the maintenance of multiple hypotheses. Additionally, the approach presented is several orders of magnitude faster than previous techniques, making it suitable for application to the action recognition dataset of chapter 3.

4.1 3D Motion Field Estimation

If we consider a scene, observed by 2 cameras $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, at 2 points in time, the task of estimating the 3D motion of a point can be considered as a 4 way correspondence problem. Figure 4.1 illustrates this using 5 different examples of correspondence combinations (images involved in the correspondence are indicated in the bottom right of each subfigure). In figure 4.1b the correspondence of a point between the two cameras, at a particular frame, specifies the 3D location of the point at that frame (as in stereo matching tasks). The correspondences between two frames of the same camera (as in optical flow tasks) is shown in figure 4.1c and defines a plane of possible 3D motions



(a) No Correspondence (b) Stereo Match (c) Optical Flow

(d) Optical Flow and Stereo Match (e) Full Correspondence

Figure 4.1: 2D simplification of the scene flow estimation task. $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are represented as boxes with the forward edge being the image plane and the cross being the focal point. Dashed lines indicate rays, along which all points project to the same pixel. Five scenarios are shown, with the same point detected in multiple sensors and frames, defined in the bottom right of each scenario. Detections of the point at frame t+1 are shown in grey.

with uncertain start and end positions. The combination of a single stereo match and a single optical flow match (figure 4.1d), meaning the point is detected in both frames of one sensor and a single frame for the other sensor, leads to a triangular section of the optical flow plane. In this case, either the start or end point is known, while the other is undetermined. Any combination of three constraints (either two stereo matches and one optical flow, or two optical flow and one stereo match) are sufficient to uniquely determine the 3D motion of the point as shown in 4.1e. In fact the combination of three correspondences uniquely defines the fourth correspondence, with a missing optical flow correspondence being given by the offset between stereo matches at each frame, while a missing stereo correspondence is determined by the end points of the flow correspondences in each sensor.

When examining figure 4.1, it is natural to also consider a scenario with only 2 stereo correspondences, or with only 2 optical flow correspondences. Such a scenario would involve all 4 images, using only two connections. This is a useful question, as it provides some insight into the difference between SFE and stereo reconstruction or optical flow estimation. Having only two stereo correspondences means we know the 3D position of a point at two frames, but there is no reason to believe it is the same point. Put another way, we can create a separate 3D structural reconstruction at every frame, but we don't know which parts of the structure are moving. Conversely two optical flow constraints means we have followed moving points on the image plane of each sensor, but they are unlikely to correspond to the same point in the scene. The end result in this case is a pair of 2D motion fields, with no knowledge of the 3D structure which gave rise to them. In contrast, SFE aims to fully describe the evolution of a 3D scene over time, by obtaining a single unified solution to both problems.

The situation shown in figure 4.1 can be extended to include additional sensors or frames, making the problem further over-constrained, without changing the requirements. Having three or more stereo matching correspondences at a single frame, is still insufficient to define motion, while larger numbers of optical flow fields still cannot define the underlying structure. This tells us that the necessary constraints for determining scene flow, are for the point to have at least 3 detections, covering at least 2 sensors and at least 2 points in time.

If information from a depth sensor is available, this replaces the stereo correspondences at every frame, meaning only a single optical flow correspondence is required to uniquely determine the 3D motion. The optical flow correspondences may be obtained from the same sensor, providing "monocular" scene flow. Such techniques are referred to as "Range Flow" [82, 107, 108]. However, it is difficult to determine motion correspondences from depth data alone, and so an appearance sensor is often used to obtain the optical flow constraint, creating a "hybrid" approach.

4.2 Probabilistic Scene Flow

Given any scene, there is a continuous 3 dimensional space of possible structure points, which the scene may contain. A particular point from this space is specified by the vector $\boldsymbol{r}$. In addition, at every one of these structure points, there is a continuous 3 dimensional space of possible motions, with points in this space referred to using the vector $\boldsymbol{v}$. The goal of SFE is to extract the set of points and motions with the highest probability, given the input observations $\boldsymbol{I}$ ("observations" in this thesis, refer to the set of images obtained from a number of synchronised appearance and/or depth sensors, in a known but unrestricted configuration). To this end, the 6 dimensional Scene Probability distribution $\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I})$ is defined, specifying how probable every combination of structure and motion is, in the observed scene. The peaks of this distribution can be used to generate an estimate of the scenes structure and motion field. Note that, there are no restrictions limiting the distribution to have a single peak per optical ray, which gives rise to the multiple hypothesis properties mentioned earlier in this chapter.

Bayes theorem may be used to define the posterior distribution $\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I})$ in terms of the prior distribution $\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v})$ and the likelihood $\boldsymbol{p}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})$.

$$\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}) \propto \boldsymbol{p}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})\, \boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v}) \tag{4.1}$$

Bayes theorem may also be extended to joint estimation over time (i.e. the probability of a sequence of structure and motion fields, given a sequence of observations), leading to equation 4.2.

$$\boldsymbol{p}_{1..t}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}) = \boldsymbol{p}_{1..t}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})\, \boldsymbol{p}_{1..t}(\boldsymbol{r}, \boldsymbol{v}) \tag{4.2}$$

A transition function $\boldsymbol{p}\left(\boldsymbol{r}_t, \boldsymbol{v}_t \mid \boldsymbol{r}_{t-1}, \boldsymbol{v}_{t-1}\right)$ may also be defined, using a constant velocity motion model, for every point $\boldsymbol{r}, \boldsymbol{v}$ in the distribution, such that $\boldsymbol{r}_t = \boldsymbol{r}_{t-1} + \boldsymbol{v}_{t-1}$ and $\boldsymbol{v}_t = \boldsymbol{v}_{t-1}$. Obviously more advanced motion models could be used, such as the incorporation of acceleration, however this would require an increase in the dimensionality of the particle filter. This transition function represents the evolution of a scene

flow field over time, and can be used to generate the prior probabilities for a given sequence, as in equation 4.3. Note that a uniform distribution is used for the prior at the first frame $\boldsymbol{p}_1(\boldsymbol{r}, \boldsymbol{v})$. The resulting system exhibits exponential stability [43], meaning that the effect of these initial conditions, decays exponentially with each new observation.

$$\boldsymbol{p}_{1..t}(\boldsymbol{r}, \boldsymbol{v}) = \boldsymbol{p}_1(\boldsymbol{r}, \boldsymbol{v}) \prod_{\tau=2}^{t} \boldsymbol{p}(\boldsymbol{r}_\tau, \boldsymbol{v}_\tau \mid \boldsymbol{r}_{\tau-1}, \boldsymbol{v}_{\tau-1}) \tag{4.3}$$

The likelihood $\boldsymbol{p}_{1..t}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})$ of a sequence of observations, given a sequence of scene flow fields is equal to the product of the likelihoods, analysed at each frame. This, in conjunction with equation 4.3, allows the transition and likelihood terms for the most recent frame t to be factored out, as shown in equation 4.4.

$$\boldsymbol{p}_{1..t}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}) = \boldsymbol{p}_t(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})\, \boldsymbol{p}\left(\boldsymbol{r}_t, \boldsymbol{v}_t \mid \boldsymbol{r}_{t-1}, \boldsymbol{v}_{t-1}\right) \boldsymbol{p}_{1..t-1}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})\, \boldsymbol{p}_{1..t-1}(\boldsymbol{r}, \boldsymbol{v}) \tag{4.4}$$

The last two terms can then be combined with equation 4.1, to obtain a recursive definition.

$$\boldsymbol{p}_{1..t}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}) = \boldsymbol{p}_t(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})\, \boldsymbol{p}\left(\boldsymbol{r}_t, \boldsymbol{v}_t \mid \boldsymbol{r}_{t-1}, \boldsymbol{v}_{t-1}\right) \boldsymbol{p}_{1..t-1}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}) \tag{4.5}$$

This formulation requires only the likelihood term from the current frame, in conjunction with the posterior from the previous frame, multiplied by the transition function. Hence, this recursive formulation is suitable for online applications, as complexity does not increase for additional frames.

The only term of equation 4.5 which remains to be defined, is the likelihood $\boldsymbol{p}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})$ specifying the chance that a set of observations would occur, given that the scene contains a structure point at $\boldsymbol{r}$ moving with velocity $\boldsymbol{v}$. The following sections define two terms, based on two information sources, which together provide the likelihood distribution. The first of these, $\boldsymbol{p}_a(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v})$ is based on observations $\boldsymbol{I}_a$ from a collection of appearance sensors. The second, $\boldsymbol{p}_d(\boldsymbol{I}_d \mid \boldsymbol{r}, \boldsymbol{v})$ is based on observations $\boldsymbol{I}_d$ from a

collection of depth sensors. The likelihood is taken as the product of these two terms, which assumes independence between the two information sources. Due to the Bayesian formulation presented, it is trivial to incorporate further information sources if such independence can be assumed, as shown in section 4.5.

4.2.1 Appearance Likelihood

If the appearance observations $\boldsymbol{I}_a$ are the output of M appearance sensors (either RGB or greyscale) with projection functions $\boldsymbol{\Pi}_{1\ldots M}$ (converting 3 element vectors $\boldsymbol{r}$ into 2D pixel locations), then the likelihood of a structure point at a given $\boldsymbol{r}$ and $\boldsymbol{v}$ can be estimated, using the brightness constancy assumption. This assumption states that the brightness of an object, remains constant over time, and when viewed from any direction (this fundamental assumption is explored and improved upon in section 5.7.1 of the next chapter). For RGB sensors, the assumption is often extended to colour constancy. If we assume the true appearance of a point is the average of the appearance observed in each sensor $\bar{I}$:

$$\bar{I} = \frac{\sum_{m=1}^{M} \sum_{\tau=t\text{-}1}^{t} \boldsymbol{I}_{a,m}^{\tau}(\boldsymbol{\Pi}_m(\boldsymbol{r}_\tau))}{2M} \tag{4.6}$$

then the squared error (divergence from brightness constancy) is equivalent to the variance of the projected appearance, across all sensors, at both the previous and current frame ($\boldsymbol{I}_{a,m}^{t\text{-}1}$ and $\boldsymbol{I}_{a,m}^{t}$ respectively). Using this cost function, a likelihood can be obtained:

$$\boldsymbol{p}_a(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v}) = \frac{1}{1 + e_a \sum_{m=1}^{M} \sum_{\tau=t\text{-}1}^{t} \frac{\left(\boldsymbol{I}_{a,m}^{\tau}(\boldsymbol{\Pi}_m(\boldsymbol{r}_\tau)) - \bar{I}\right)^2}{2M}} \tag{4.7}$$

where $\boldsymbol{r}_{t\text{-}1}$ can be found from $\boldsymbol{r}_t - \boldsymbol{v}$. Thus the likelihood depends only on the current $\boldsymbol{r}$ and $\boldsymbol{v}$ values (i.e. the likelihood formulation is memoryless).

Note that previous works often assume either Gaussian measurement noise [63], or utilise a function referred to "smooth" or "robust" approximation to the L_1 norm [19]

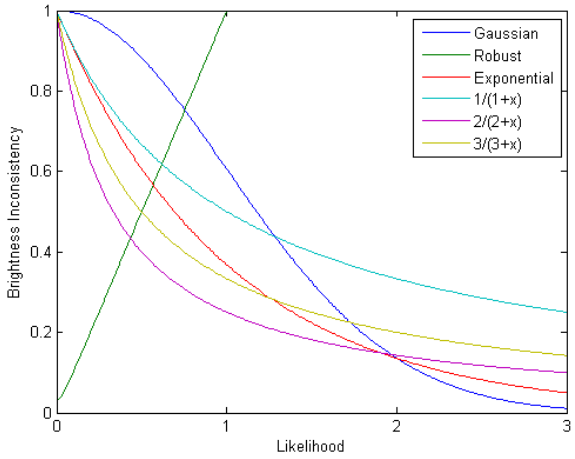


Figure 4.2: Comparison of brightness constancy functions. In cyan, purple and yellow the formula of equation 4.7 is shown with different values of e_a (1,2 and 3 respectively). In Green the unbounded "smooth L_1" is shown, in blue the Gaussian formulation is shown and in red the exponential decay cost is shown.

when penalising divergences from brightness constancy. Figure 4.2 illustrates the behaviour of these functions. The robust estimation function grows unboundedly and is thus more suitable for energy minimisation tasks than as a probability measure. The Gaussian assumption on the other hand, was found to poorly reflect true motion estimation noise by Sun *et al.* [118], who found that motion estimation experiences tighter peaks and heavier tails, for a given variance. The formulation presented here produces a well formed likelihood (i.e. between 0 and 1) for all possible constancy divergences, while also displaying the characteristics identified by Sun *et al.* (tighter peaks and heavier tails than both the Gaussian and exponential functions). The parameter e_a is used to control the kurtosis of the distribution, and is set to 1 for the remainder of this manuscript.

4.2.2 Depth Likelihood

When depth information is present, the task is simplified, as most ambiguity in $\boldsymbol{r}$ is removed, this is discussed in more detail in section 4.3. Additionally depth sensors can

contribute to the likelihood calculation, in a similar way to appearance sensors.

Given observations $\boldsymbol{I}_d$ from a collection of L depth sensors, with projection functions $\boldsymbol{\Psi}_{1...L}$ and reverse projection functions $\boldsymbol{\Psi'}_{1...L}$, the likelihood of a structural point at $\boldsymbol{r}_t$ is determined by how closely it matches the back-projections of the depth observations at frame t. Similarly $\boldsymbol{r}_{t\text{-}1}$ should fit the back-projection of the previous depth observations at frame t-1. To quantify this, the average square distance, between the position $\boldsymbol{r}$, and each sensors back-projection along that ray, is calculated as in equation 4.8.

$$\boldsymbol{p}_d(\boldsymbol{I}_d \mid \boldsymbol{r}, \boldsymbol{v}) = \frac{1}{1 + e_d \sum_{l=1}^{L} \sum_{\tau=t\text{-}1}^{t} \frac{\left(\boldsymbol{\Psi'}_l(\boldsymbol{I}^{\tau}_{d,l}(\boldsymbol{\Psi}_l(\boldsymbol{r}_\tau))) - \boldsymbol{r}_\tau\right)^2}{2L}} \tag{4.8}$$

Note that the e_a and e_d parameters may be modified, to control the relative contribution of the appearance and depth information, based on expected noise in each. Also note that if either M or L are zero (only one type of sensor is present) the relevant likelihood term simplifies to 1, and $\boldsymbol{p}\left(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v}\right)$ depends entirely on the remaining sensor modality (assuming the sum of an empty set is defined as zero). Thus the approach is generalizable to any combination of inputs, including appearance only, depth only and hybrid setups.

Images which $\boldsymbol{r}_t$ does not project to, are ignored during the likelihood calculation. However, if a point does not project to at least 3 sensors, including at least 1 at the current frame and one at the previous frame, then its motion is undefined, and its likelihood is set to 0. These constraints match those explained in section 4.1, necessary to uniquely define a 3D motion, and are irrespective of sensor configuration (although depth sensors provide the equivalent of a stereo match at each frame). These constraints prevent the degenerate case, where a point visible to only one sensor would always have zero colour variance, and be assigned a likelihood of 1.

4.3 Scene Particles

As $\boldsymbol{p}\left(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I}\right)$ is a continuous distribution, there are an infinite number of possible structure points, with an infinite number of motions. Even applying a coarse discreti-

sation, it remains intractable to compute the 6 dimensional distribution. Even a simple setup using a pair of 640 by 480 cameras leads to millions of possible structure points, each with an equivalent number of possible velocities. Instead a Monte-Carlo sampling approach is employed, to analyse only a small subset of points within the distribution, and provide an approximation of the continuous probabilistic system, while remaining computationally feasible. Specifically, inspiration is taken from particle filtering techniques, with samples having associated weights and resampling techniques employed to concentrate sampling density in more promising areas of the distribution. Particle filtering approaches are valuable, as they offer bounded convergence rates, which do not depend on the dimensionality of the space [43]. It is important to note however, that there are a number of fundamental differences (discussed in detail in section 4.3.1) between the algorithm presented here, and standard particle filtering techniques.

Throughout the remainder of this manuscript the samples of $\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I})$ are referred to as "scene particles". Each of the N scene particles ($\boldsymbol{P}_{0\ldots N}$) specifies a 3D spatial location and 3D velocity (i.e. an $\boldsymbol{r}$ and $\boldsymbol{v}$ pair, such that $\boldsymbol{P}_n \in \mathbb{R}^6 = \{\dot{x}, \dot{y}, \dot{z}, v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}\}$, where dot notation represents the 3D scene rather than image plane), and when taken together they form the particle population P. The weight w_n of scene particle $\boldsymbol{P}_n$ is obtained by analysing the likelihood $\boldsymbol{p}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})$ at the specified point. As new observations are obtained, the scene particles from the previous frame are propagated using a constant velocity motion model. This then provides a sampled approximation of the prior probability $\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v})$ as in equation 4.4. This formulation leads to the principled propagation of information over time, ensuring temporally consistent results without the flickering seen in many SFE techniques. At the first frame, the prior distribution is initialised with a randomly generated, uniformly weighted collection of scene particles. Figure 4.3 shows the flow diagram for the system.

When the number of depth sensors L is nonzero, the additional "clamp" preprocessing stage is performed to account for the reduction in spatial ambiguity. The projections of each scene particle $\boldsymbol{P}_n$ are calculated for all depth images $\boldsymbol{I}^t_{d,0\ldots L}$. The subset of sensors with a valid depth estimate at the projected location is extracted, and one sensor is randomly selected to provide a new spatial location $\boldsymbol{r}$ for $\boldsymbol{P}_n$. Note that there are multiple samples along each ray. Thus, in the case where some sensors observe structure

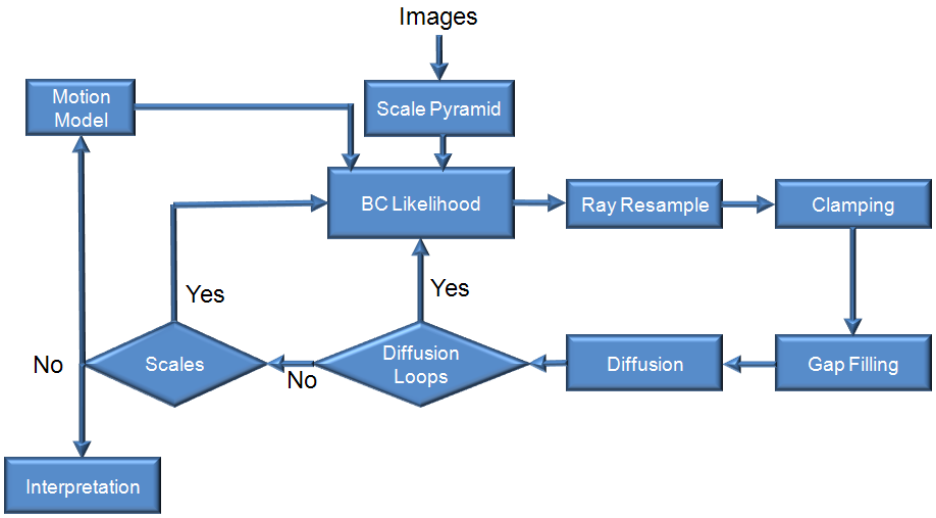


Figure 4.3: Flow diagram for the scene particle algorithm. The two iterative layers (Diffusion and scales) are discussed in section 4.3.2. Also note the feedback of the previous estimate as an input for the next frame.

which is occluded to other sensors, the random selection process ensures the resulting particle population reflects the proportions of sensor visibility in the scene. The velocity values of the scene particles remain unchanged, such that multiple hypotheses sample different velocities $\boldsymbol{v}$ at each spatial location $\boldsymbol{r}$. This can be seen as the collapse (or clamping) of the 6D distribution onto a complex 3D surface, lying within the original 6D space.

4.3.1 Ray Resampling

Particle filtering techniques involve an additional step known as resampling, in which samples are re-drawn from the probability distribution, to solve the problem of "particle depletion". Resampling serves to focus hypotheses into promising regions, in order to make the best use of the finite number of particles. If the initial particle cloud was directly re-used at every frame, via the transition kernel, it is likely that only a handful of samples would be evaluating high probability areas. This is termed "particle depletion" as very few of the particles are useful. One common resampling schemes is Multinomial Resampling [56], in which the sampling density is determined by the

current estimate of the posterior, such that high probability regions are more densely sampled. This is achieved by constructing the resampled set P_t, by drawing N particles from the previous set $P_{t\text{-}1}$, with the probability of drawing particle n given by its normalised weight $\bar{w}_n$.

$$\bar{w}_n = \frac{w_n}{\sum_{j=0}^{N} w_j} \tag{4.9}$$

In this manuscript the residual resampling scheme [46] is employed, which makes multinomial resampling more deterministic, forcing each particle $\boldsymbol{P}_n$ to spawn $\bar{w}_n|P|$ particles, then randomly drawing any remaining samples (due to rounding) based on the residual weights. This scheme redistributes the particles, such that the density of samples at each point in the space is determined by the estimated posterior probability at that point. Thus, the new particles are assigned uniform weighting, such that the population as a whole reflects the underlying distribution.

Unfortunately, the scene particles algorithm is fundamentally different to standard particle filtering systems. In general for a particle filter, each hypothesis relates to a complete solution to the task. A standard particle filter implementation would be based on the joint space across all pixels, i.e. a $width \times height \times 6$ dimensional space, rather than simply 6. However, the number of samples required for a particle filter is proportional to the number of dimensions, and so the scene particles algorithm is formulated such that the output motion field comprises a large collection of high probability samples. This shifts the focus of the system towards "preserving" all potentially valid hypotheses, rather than "killing" hypothesis which may be invalid.

It is well known, and documented in detail by Sidenbladh [114], that a particle population undergoing many resampling iterations will converge, until it is performing only local exploration of the largest discovered peak, losing any secondary modes from the distribution. In contrast, the scene particle algorithm aims to estimate a large number of local maxima, in order to estimate the motion of the whole scene. In this context, the loss of secondary peaks leads to reduced scene coverage of the estimated motion field, eventually leading to motion only being estimated at a single point in the scene.

In this manuscript, this is referred to as the "overconvergence problem". To avoid this behaviour, a modified resampling scheme is used, termed "Ray Resampling". Under this scheme, the particle population P_t is partitioned into subsets, based on the optical rays the particles lie along, i.e. the subset $\Theta_{c,x,y}$ for sensor c and ray originating at pixel x, y is defined as:

$$\Theta_{c,x,y} \subset P_t \tag{4.10}$$

$$\Theta_{c,x,y} = \left\{ \boldsymbol{P}n \,\middle|\, \boldsymbol{\Pi}_c(\boldsymbol{P}n) = (x,y)^T \right\} \tag{4.11}$$

It is important to note that subsets for a single sensor are disjoint, i.e. no scene particle lies along more than one ray of the same sensor. However, subsets from different sensors are not mutually exclusive, so a scene particle may be part of multiple subsets, up to the number of sensors (depending on visibility). Put more formally:

$$\Theta_{c,x,y} \cap \Theta_{c,x',y'} = \emptyset \text{ for all } x,y \neq x',y' \tag{4.12}$$

$$\left| \left\{ \Theta_{c,x,y} \,\middle|\, \boldsymbol{P}n \in \Theta_{c,x,y} \right\} \right| \leq M + L \tag{4.13}$$

Once the scene particle population is partitioned, residual resampling is performed separately within each partition. The particle population P_t is constructed by drawing $\frac{N}{R}$ scene particles from each of the R partitions. The probability of each scene particle being drawn from a particular partition is defined as $\bar{w}n$, given by the weight, normalised within that partition, as in equation 4.14.

$$\bar{w}n = \frac{w_n}{\sum\limits_{\boldsymbol{P}j \in \Theta_{c,x,y}} w_j} \tag{4.14}$$

This process ensures that a minimum of $\frac{N}{R}$ scene particles are used to estimate the motion at every point in the scene, avoiding the loss of coverage, while still allowing

convergence of the particles in the motion dimensions, and spatially along rays. As subsets are extracted from all sensors, the resultant motion has the valuable property of being dense in all viewpoints. This is in stark contrast to the majority of SFE and Stereo Reconstruction algorithms, which estimate densely only in a single, arbitrarily selected, "reference" view.

Occasionally no scene particles will lie along an optical ray, leaving the related subset empty. In this case, the subset is filled with scene particles randomly selected from the 8 neighbouring rays, in the same sensor. Drawn particles have their position modified, so they lie long the empty ray, while leaving the motion parameters and distance along the ray intact. The random selection scheme ensures that the main modes of the surrounding data will be reflected in the new subset. This filling of empty partitions can be seen as smoothing of the scene particle cloud (the only instance of such in the algorithm). In practice however, this is necessary for less than 0.1% of partitions, and ensures a fully dense motion field.

4.3.2 Multiscale Iterative Estimation

To improve the accuracy of the estimated motion field, two iterative refinement techniques are utilised. Firstly for each new set of observations $\boldsymbol{I}$, the likelihood $\boldsymbol{p}\left(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v}\right)$ is iterated over S scales in a coarse to fine manner, using scaled representations of the input observations ($\boldsymbol{I}^{1\ldots S}$). These are obtained via convolution with zero mean Gaussian kernels $g\left(0, \sigma\right)$, with σ representing the variance of the Gaussian. Scaled observations are constructed such that each is half the scale of the previous, as shown in equation 4.15.

$$\boldsymbol{I}^{s} = \begin{cases} \boldsymbol{I}, & s = 0 \\ \boldsymbol{I} * g\left(0, 2^{s}\right), & \text{otherwise} \end{cases} \tag{4.15}$$

This leads us to define the likelihood term from 4.5 as the product of terms for each scale observation.

$$\boldsymbol{p}\left(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v}\right) = \prod_{s=0}^{S} \mathbf{p}(\boldsymbol{I}^s \mid \boldsymbol{r}, \boldsymbol{v}) = \prod_{s=0}^{S} \mathbf{p}(\boldsymbol{I}_a^s \mid \boldsymbol{r}, \boldsymbol{v}) \mathbf{p}(\boldsymbol{I}_d^s \mid \boldsymbol{r}, \boldsymbol{v}) \tag{4.16}$$

These multi-scale approaches are commonly used in variational algorithms, where the optimisation always converges to the local minima closest to its initial state. Without a multi-scale system, it is very difficult for such algorithms to estimate large motions, as moving further along the cost surface increases the chances of getting stuck in local minima. When performing a coarsely discretised optimisation, it is hoped that some of these local minima are smoothed out, allowing the initial estimate at the next finer scale to begin closer to the global minima.

In the scene particle algorithm, the multi-scale iteration serves a different purpose. Due to the multi-hypothesis nature of the approach, local minima are less of an issue, and large motions are no more or less probable than small motions. Instead the multi-scale estimation allows "contextual" observations (i.e. the same point observed at different scales) to be employed, to disambiguate between similar motions.

4.3.3 Iterative Diffusion

The second iterative scheme, is inspired by Particle Swarm Optimisation (PSO) techniques, and involves a change in the behaviour of the scene particles during the diffusion stage. After resampling, the particle population contains significant redundancy, due to certain particles being copied multiple times. To counteract this, the diffusion stage is used to inject variance into the population, following the Resample-Move approach outlined by Berzuini and Gilks [23]. As shown in equation 4.17, random perturbations $\boldsymbol{\gamma}$ are drawn from a 6 dimensional Gaussian $g\left(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s\right)$.

$$\boldsymbol{P}n = \boldsymbol{P}n + \boldsymbol{\gamma} \text{ where } \boldsymbol{\gamma} \sim g\left(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s\right) \tag{4.17}$$

The mean vector and covariance matrix of the Gaussian distribution used for diffusion, is determined by the range of values visible in the scene $\boldsymbol{\Phi}$ such that the diagonal matrix $\boldsymbol{\sigma}_s$ rescales each dimension independently, based on its range.

$$\boldsymbol{\mu}_s = \frac{\max_e(\boldsymbol{\Phi}) + \min_e(\boldsymbol{\Phi})}{2} \tag{4.18}$$

$$\boldsymbol{\sigma}_s = \mathrm{diag}\left(\delta\left(\max_e(\boldsymbol{\Phi}) - \min_e(\boldsymbol{\Phi})\right)\right) \tag{4.19}$$

Note that $\min_e$ and $\max_e$ represent the vector formed from the elementwise maximum and minimum respectively. Equation 4.20 defines $\boldsymbol{\Phi}$ as the range of $(\boldsymbol{r}, \boldsymbol{v})$ values which can project to every sensor at both the current and the previous frame. Note that this is a property of the scene and camera setup. For stationary cameras $\boldsymbol{\Phi}$ does not vary over time. and is not dependent on the particle population.

$$\boldsymbol{\Phi} = \left\{ (\boldsymbol{r}, \boldsymbol{v}) \;\middle|\; \begin{array}{c} \boldsymbol{\Pi}_m(\boldsymbol{r}) \in \boldsymbol{I}^t_{a,m} \text{ and } \boldsymbol{\Pi}_m(\boldsymbol{r} - \boldsymbol{v}) \in \boldsymbol{I}^{t\text{-}1}_{a,m} \text{ for all } m \\ \boldsymbol{\Psi}_l(\boldsymbol{r}) \in \boldsymbol{I}^t_{d,l} \text{ and } \boldsymbol{\Psi}_l(\boldsymbol{r} - \boldsymbol{v}) \in \boldsymbol{I}^{t\text{-}1}_{d,l} \text{ for all } l \end{array} \right\} \tag{4.20}$$

If depth sensors are present, then only exploration along the velocity space manifold is necessary. As a result the first 3 elements of $\boldsymbol{\mu}_s$ and along the diagonal of $\boldsymbol{\sigma}_s$ are set to 0, i.e. there is a 100% chance that no diffusion occurs in these dimensions.

Note that the variance of the distribution that perturbation $\boldsymbol{\gamma}$ is drawn from, is scaled by the parameter δ in equation 4.19. This parameter controls a trade-off between exploration and accuracy. Higher levels of δ lead to exploratory behaviour with the possible discovery of new modes, while low levels lead to the refinement of the current modes. The second layer of the iterative estimation involves the repeated sampling of the likelihood $\boldsymbol{p}\left(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v}\right)$ as in equation 4.16, followed by resampling and diffusion with gradually reduced values of δ.

4.3.4 Interpretation

The scene particle cloud provides a probabilistic, multi-hypothesis representation of the scenes structure and motion. However, this representation is unsuitable for some tasks, most notably for comparative evaluation of the algorithm. In general, the ground truth for existing scene flow datasets, is provided as an optical flow field in conjunction with a stereo flow field (change in disparity between frames, for every pixel). This is due to

traditional approaches often being formulated with only 2 viewpoints, and only being dense in one reference view.

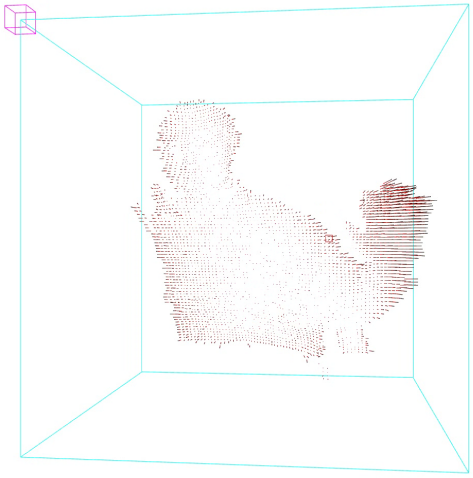
In order to convert the scene particle cloud to a compatible representation for comparison, the multiple hypotheses and associated probabilities need to be simplified. To do this, an image $\boldsymbol{I}_o$ is generated for a "reference sensor" c, where each pixel x, y is filled, using the weighted average of the scene particles from that rays partition $\Theta_{c,x,y}$. The diffusion system injects variance into the samples, which can lead to jittery results, however the weighted average proves more stable, as the resampling scheme has already clustered the scene particles around high probability regions.

The output image $\boldsymbol{I}_o$ contains 4 channels, encoding the horizontal and vertical image plane motions v_x, v_y (found by projecting the start and end points of the 3D flow vector), the out of plane motion v_d (in terms of disparity change between frames), and the disparity of the projected point d. The interpretation stage has no effect on the original scene particle cloud, and is performed on a per frame basis.

The weighted averaging is a simple interpretation process and removes any notion of multiple modes in the data, such as multiple structures, at different distances from the sensor. $\boldsymbol{I}_o$ can be seen as a simplified view of the scene particle cloud, from the point of view of reference sensor c. Thus it is reasonable that multiple modes are not present, even when such modes may be resolvable by other sensors. This implies that the interpretation of the same scene particle cloud may differ considerably between sensors. This may prove valuable, for tasks where each sensor is required to make a locally optimal decision, for example a collection of co-operating robotic systems.

4.4 Scene Flow Evaluation

Next the performance of the scene particle algorithm will be compared to previous state of the art SFE techniques. Additionally, various properties of the algorithm will be explored. Qualitative evaluation is performed with a number of datasets, including long (around 300 frames) "hybrid" sequences, composed of appearance and depth data, recorded from a Microsoft Kinect$^{\text{TM}}$. Figure 4.4 shows an example frame from one



(a) Motion Field

(b) RGB

(c) Depth

Figure 4.4: Example scene flow estimation. Images taken from frame 66 of the Kinect "Wave" sequence. Images 4.4b and 4.4c show the appearance and depth input data respectively. Image 4.4a contains the estimated motion field. Motion direction is from the red to the black vertices.

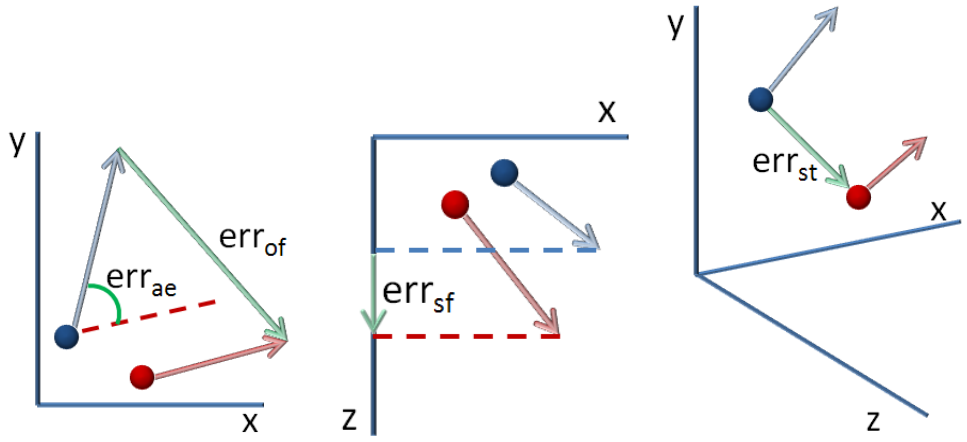
of these sequences. Performing quantitative comparisons of SFE techniques is more challenging, due to the difficulty acquiring ground truth 3D motion, at every point in a scene. Synthetic data such as the simulated driving sequences of the Enpeda dataset [123] provide one possibility for obtaining the dense ground truth motion, however it has been demonstrated by Vaudrey *et al.* [123] that stereo and motion estimation techniques exhibit fundamentally different behaviour, when tested on real and synthetic datasets (the reasons behind this, are explored more deeply in section 5.7). As a result, quantitative comparisons in this chapter are performed using 3 datasets from the Middlebury benchmarking system at `vision.middlebury.edu` [106], which are the only non-synthetic scene flow datasets available, with existing state of the art SFE results for comparison.

The Middlebury datasets used were originally designed as benchmarks for stereo reconstruction algorithms, example images are shown in figure 4.5. In order to use them for the analysis of SFE, the procedure developed originally by Huguet and Devernay [64] is followed. Each dataset comprises of a complex scene, imaged for a single frame,

Figure 4.5: Middlebury examples. Images from the middlebury Cones and Teddy datasets (left and right respectively). Note the scenes are highly cluttered, and include many discontinuities.

by 9 equally spaced and parallel RGB cameras. Ground truth disparity maps are also provided, accurate up to $\frac{1}{4}$ of a disparity (i.e. 0.25 pixels), which were captured using a structured light system. From this data it is possible to construct a collection of multi-frame "sequences", which are equivalent to viewing a moving scene from a smaller number of sensors. As a specific example, the images $\{I_1...I_9\}$ could form 4 sequences, each of 2 frames, with the first frames of each sequence being $\{I_1, I_3, I_5, I_7\}$ respectively, and the last frames being $\{I_2, I_4, I_6, I_8\}$. In this case, each of the 4 sensors appears to undergo a translation equal to the separation between sensors in the original setup. Equivalently to this, the sequences also represents 4 static sensors, viewing a rigidly translating complex scene. Obviously other combinations of images can be used to simulate different sequences, including videos of more than 2 frames, however due to the high computational cost of existing SFE techniques, no results are available for comparison in these cases.

To evaluate performance, the Normalised Root Mean Square (NRMS) error metrics proposed by Basha *et al.* are used [19], which normalise error scores by the range of values present in the ground truth, in order to facilitate comparisons between different scenes. As in most previous work [19, 64, 133, 122], four main types of error are measured (illustrated in figure 4.6). The first error (relating to the green arrow in



(a) Directional and Optical Flow errors

(b) Stereo flow error

(c) Structural error

Figure 4.6: Illustration of the types of scene flow error measure. A ground truth structure point and flow is shown in blue, and an estimated point and flow is shown in red. The various error measures are shown in green. Note that in reality err_{of} and err_{sf} disregard error components due to incorrect structural estimation (which is measured entirely by err_{st})

figure 4.6a) relates to the image plane motion magnitude. The pixelwise error is given by equation 4.21, based on the output of the interpretation stage $\boldsymbol{I}_o$ and an equivalent 4 channel image $\boldsymbol{I}_g$, which contains the ground truth v_x, v_y,v_d and d values. Note that in the following equations, the second subscript for images denotes the channel under consideration (i.e. $\boldsymbol{I}_{g,v_x}$ is the x velocity channel of the ground truth image).

$$err_{of}(x,y) = \left| \sqrt{\boldsymbol{I}_{g,v_x}(x,y)^2 + \boldsymbol{I}_{g,v_y}(x,y)^2} - \sqrt{\boldsymbol{I}_{o,v_x}(x,y)^2 + \boldsymbol{I}_{o,v_y}(x,y)^2} \right| \quad (4.21)$$

The normalised RMS form of this error err^n_{of}, is then calculated by averaging the squared pixelwise errors across the image, taking the square root of the result, and rescaling by the ground truth range.

$$err_{of}^{n} = \frac{\sqrt{\frac{1}{|x,\, y \in \boldsymbol{I}_o|} \sum\limits_{x,\, y \in \boldsymbol{I}_o} \left(err_{of}\left(x,y\right)\right)^2}}{\max\limits_{x,\,y}\left(\sqrt{\boldsymbol{I}_{g,v_x}\left(x,y\right)^2 + \boldsymbol{I}_{g,v_y}\left(x,y\right)^2}\right) - \min\limits_{x,\,y}\left(\sqrt{\boldsymbol{I}_{g,v_x}\left(x,y\right)^2 + \boldsymbol{I}_{g,v_y}\left(x,y\right)^2}\right)} \tag{4.22}$$

The second error measure (shown in figure 4.6b) relates to the out of plane motion magnitude. The pixelwise form (err_{sf}) is defined as:

$$err_{sf}\left(x,y\right) = \left|\boldsymbol{I}_{g,v_d}\left(x,y\right) - \boldsymbol{I}_{o,v_d}\left(x,y\right)\right| \tag{4.23}$$

While the total NRMS error score (err_{sf}^{n}), is given by:

$$err_{sf}^{n} = \frac{\sqrt{\frac{1}{|x,\, y \in \boldsymbol{I}_o|} \sum\limits_{x,\, y \in \boldsymbol{I}_o} \left(err_{sf}\left(x,y\right)\right)^2}}{\max\limits_{x,\,y}\left(\boldsymbol{I}_{g,v_d}\left(x,y\right)\right) - \min\limits_{x,\,y}\left(\boldsymbol{I}_{g,v_d}\left(x,y\right)\right)} \tag{4.24}$$

The third error measurement (illustrated in figure 4.6c) relates to the accuracy of the structural reconstruction. Obviously this error is negligible when operating in "hybrid" mode (i.e. when structural information is used as an input to constrain the estimation), but it is useful when analysing of multi-view techniques. The per pixel error (err_{st}) is defined in equation 4.25.

$$err_{st}\left(x,y\right) = \left|\boldsymbol{I}_{g,d}\left(x,y\right) - \boldsymbol{I}_{o,d}\left(x,y\right)\right| \tag{4.25}$$

Leading to the overall NRMS error score (err_{st}^{n}):

$$err_{st}^{n} = \frac{\sqrt{\frac{1}{|x,\, y \in \boldsymbol{I}_o|} \sum\limits_{x,\, y \in \boldsymbol{I}_o} \left(err_{st}\left(x,y\right)\right)^2}}{\max\limits_{x,\,y}\left(\boldsymbol{I}_{g,d}\left(x,y\right)\right) - \min\limits_{x,\,y}\left(\boldsymbol{I}_{g,d}\left(x,y\right)\right)} \tag{4.26}$$

Following the convention of previous authors [19, 64, 133, 123, 122], the final error measure (directional accuracy, represented by the green arc in figure 4.6a) is defined in a slightly different form. The per pixel directional error (err_{ae}) is defined in equation 4.27.

$$err_{ae}(x,y) = \left| \cos\left(\frac{\boldsymbol{I}_{g,v_x}(x,y)}{\boldsymbol{I}_{g,v_y}(x,y)}\right) - \cos\left(\frac{\boldsymbol{I}_{o,v_x}(x,y)}{\boldsymbol{I}_{o,v_y}(x,y)}\right) \right| \tag{4.27}$$

For the overall directional error, the Average Angular Error (AAE) measure is used as in equation 4.28. The lack of scene specific normalisation, implies the assumption that every scene contains a similar range of motion orientations.

$$err^{n}_{ae} = \frac{1}{|x,y \in \boldsymbol{I}_o|} \sum_{x,\,y \in \boldsymbol{I}_o} (err_{ae}(x,y)) \tag{4.28}$$

Unless specified otherwise, experiments in the following sections are performed using 20 scene particles per ray ($\frac{N}{R} = 20$), and with particle filter bounds equal to 50% of the maximum values $\boldsymbol{\Phi}$ possible for the scene. The number of image scales S is set to 6, and the diffusion variance δ is set to 3% of $\boldsymbol{\Phi}$, with 3 iterations reducing δ by a third each time.

4.4.1 Algorithmic Comparison

For comparison with existing techniques, the scene particles algorithm is run in two different modes. First, using appearance information only from 4 sensors, and secondly in hybrid mode using appearance and depth data from a single viewpoint (similar to the output of a Microsoft Kinect $^{\text{TM}}$). Results are compared to the multi-scale variational approach of Huguet *et al.* [64] and the extensible 3D variational formulation of Basha *et al.* [19], both of which are appearance only techniques. Very few hybrid techniques exist for comparison, and none which have been tested on publicly available data. Thus, for comparison purposes, an alternative hybrid technique is implemented. This technique operates by running the state of the art optical flow algorithm of Farneback *et al.* [48] on the appearance data, and combining these results with the ground truth depth, in order to infer 3D motion. In results table 4.1 this technique is referred to as OF+D.

Examining the performance of the hybrid mode techniques, it is apparent that the OF+D algorithm is significantly faster than the other algorithms. However, accuracy suffers, with the hybrid mode scene particles approach exhibiting greater motion magnitude accuracy across all datasets, and greater directional accuracy for 2 of the 3

Algorithm	Dataset	App. Sensors	Depth Sensors	err^n_{of}	err^n_{sf}	err^n_{st}	err^n_{ae}	Coverage	Runtime per frame
Scene Particles	Cones	1	1	0.09	**0.00**	-	5.02	**100%**	198 secs
OF+D	Cones	1	1	0.22	0.38	-	4.63	88%	**20 secs**
Scene Particles	Cones	4	0	**0.06**	**0.00**	**0.29**	4.08	**100%**	213 secs
Basha *et al.* [19]	Cones	2	0	3.07	0.03	6.52	0.39	**100%**	-
Basha *et al.* [19]	Cones	4	0	1.32	0.01	6.22	**0.12**	**100%**	-
Huguet *et al.* [64]	Cones	2	0	5.79	8.24	5.55	0.69	**100%**	5 Hours
Scene Particles	Teddy	1	1	0.11	**0.00**	-	5.04	**100%**	201 secs
OF+D	Teddy	1	1	0.31	0.29	-	12.33	68%	**18 secs**
Scene Particles	Teddy	4	0	**0.04**	**0.00**	**0.20**	4.23	**100%**	207 secs
Basha *et al.* [19]	Teddy	2	0	2.85	0.07	7.04	1.01	**100%**	-
Basha *et al.* [19]	Teddy	4	0	2.53	0.02	6.13	**0.22**	**100%**	-
Huguet *et al.* [64]	Teddy	2	0	6.21	11.58	5.64	0.51	**100%**	5 Hours
Scene Particles	Venus	1	1	0.09	**0.00**	-	5.44	**100%**	191 secs
OF+D	Venus	1	1	0.38	0.23	-	12.21	98%	**21 secs**
Scene Particles	Venus	4	0	**0.04**	**0.00**	**0.25**	4.26	**100%**	213 secs
Basha *et al.* [19]	Venus	2	0	1.98	**0.00**	6.36	1.58	**100%**	-
Basha *et al.* [19]	Venus	4	0	1.55	**0.00**	5.39	1.09	**100%**	-
Huguet *et al.* [64]	Venus	2	0	3.70	3.05	5.79	**0.98**	**100%**	5 Hours

Table 4.1: Scene flow estimation performance for a range of algorithms. Both hybrid (depth and appearance) mode, and pure appearance only mode are tested. Note that runtimes were not reported by [19] and structure errors are negligible when using depth sensors. Bold entries highlight the best performance in each column, for that dataset.

datasets. It is interesting to note that the OF+D approach employs a dedicated optical flow algorithm, and yet the motion accuracy within the image plane, is still worse than that of the scene particles approach. This implies that the incorporation of depth data at an earlier stage allows more accurate flow estimates, even in 2D. Additionally OF+D is the only technique which does not guarantee 100% scene coverage for the estimated motion field (although in some cases it may approach this). The final difficulty with OF+D is that it is not extensible, unlike Basha *et al.* which can incorporate any number of appearance sensors, or the scene particles system which can incorporate any combination of appearance and depth sensors.

In the multiview appearance mode, the performance of the scene particles algorithm appears to slightly improve over the hybrid mode results. This suggests that the larger number of input images and disambiguation provided by multiple viewpoints, compensates for the increased difficulty of estimating both structure and motion simultaneously. The structure estimation of the multiview scene particle algorithm is significantly improved over the dedicated multiview techniques. Additionally, the motion magnitude accuracy for the scene particle algorithm is consistently higher, both within the image plane, and perpendicular to it. However, directional estimation accuracy is slightly reduced. This can be attributed to the stochastic nature of the scene particles algorithm, which leads estimated motion fields to contain a low level of noise. In terms of motion magnitude, this noise is generally insignificant, however in regions of low motion (such as background regions) a small change in absolute motion, leads to a large change in direction.

To make runtime comparisons more fair, the speed listed is for sequential computation on a single thread, without exploiting the possibility for massive parallelisation provided by the independence of scene particles. Computation time was not provided by Basha *et al.* however similarities in the optimization scheme mean it can reasonably be assumed to be of a similar order to that of Huguet *et al.* As such, the scene particles approach operates roughly 100 times faster, across all datasets, than previous state of the art algorithms.

The results shown in this section were analysed using single frame sequences for each

Algorithm	Dataset	App. Sensors	Depth Sensors	err^n_{of}	err^n_{sf}	err^n_{ae}	Coverage
scene particles Non-RR	Cones	1	1	0.10	**0.00**	5.10	49%
scene particles	Cones	1	1	**0.09**	**0.00**	**5.02**	**100%**
scene particles Non-RR	Teddy	1	1	**0.10**	**0.00**	5.10	50%
scene particles	Teddy	1	1	0.11	**0.00**	**5.04**	**100%**
scene particles Non-RR	Venus	1	1	**0.08**	**0.00**	5.50	51%
scene particles	Venus	1	1	0.09	**0.00**	**5.44**	**100%**

Table 4.2: Ray Resampling performance. Results of scene flow estimation in a hybrid (depth and appearance) sensor system. scene particles with and without ray resampling (RR) are compared, in terms of accuracy and coverage.

sensor. As a result, there is no propagation of information between frames for the scene particles algorithm. Instead improvements in performance can be attributed to the maintenance of multiple hypotheses, the principled fusion of information across scales, and the lack of oversmoothing artefacts.

4.4.2 Ray Resampling Analysis

To demonstrate the overconvergence issues discussed in 4.3.1 and quantitatively examine the value of the Ray Resampling approach, table 4.2 compares the performance of the scene particles algorithm using a standard resiudal resampling scheme (Non-RR), and using the proposed Ray Resampling scheme.

It would have been reasonable to assume that the overconvergence issue, would lead to the accumulation of particles into regions of high accuracy, causing reduced coverage and correspondingly increased accuracy in the regions which remain covered. However, this does not appear to be the case, in fact the Ray Resampled approach produces

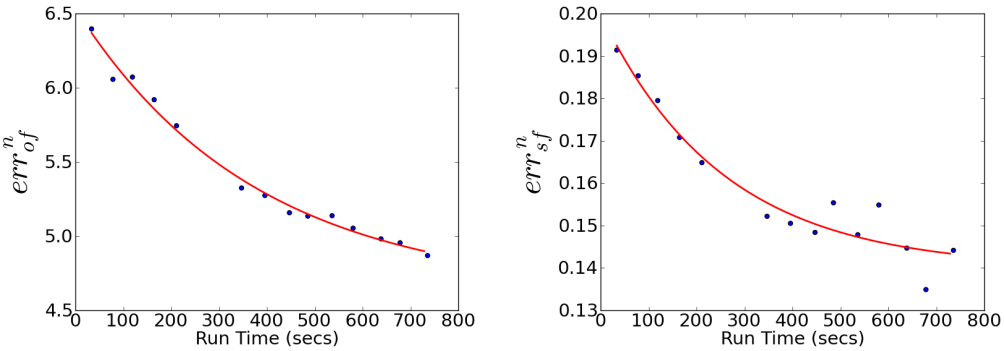
more accurate directional estimates on all 3 datasets (and roughly equivalent motion magnitude performance). This implies that, within the subset of local maxima of $\boldsymbol{p}(\boldsymbol{r}, \boldsymbol{v} \mid \boldsymbol{I})$, higher probabilities do not automatically correspond to reduced ambiguity.

4.4.3 Sampling Density

As with any Monte-Carlo based approach, the sampling density of the scene particles algorithm determines how accurately the underlying distribution is approximated. This in turn specifies how effectively the local maxima may be extracted. The trade-off, is that denser sampling implies greater computation cost. For the scene particles algorithm, it is convenient to define the sampling density in terms of scene particles per ray, which relates it to the fidelity of the input observations. As an example, two scenes with the same range of $\boldsymbol{r}$,$\boldsymbol{v}$ values $\boldsymbol{\Phi}$ to explore, are observed by two sets of sensors. The first set contains high resolution sensors, and the second low resolution. If the same number of scene particles are used in both cases, the sampling density in terms of the $\boldsymbol{r}$,$\boldsymbol{v}$ is the same, however analysis of the systems performance will reveal significantly increased error rates for the high resolution system. This is because the accuracy of the resultant motion field, is also examined more densely, and fewer hypotheses are present in each partition, during ray-resampling. By specifying the number of hypotheses per ray, the sampling density is defined in a manner which is invariant to the specifics of the scene, and the sensors under consideration.

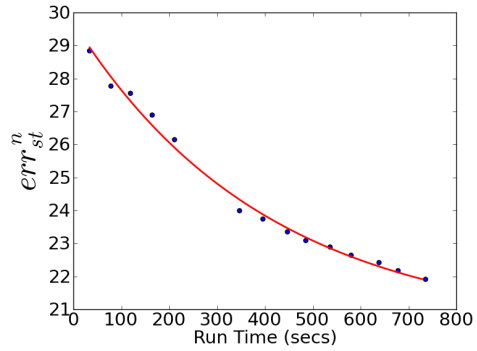
The ability to vary the sampling density is a useful feature of the scene particles algorithm. In essence, it is a parameter which enables a trade-off between speed and accuracy, depending on the requirements of the task. A parallelised implementation reduces the importance of this trade-off, by allowing additional samples to be analysed concurrently. However the sampling density also controls a second trade-off, between accuracy and memory usage, which may become more relevant with the limitations of GPU memory. Figure 4.7 demonstrates the effects of varying sampling density, for values from 1 to 40 hypotheses per ray.

All error measures exhibit an exponential relationship to the sampling density. At around 5 minutes per frame, the directional accuracy plateaus, with an angular error

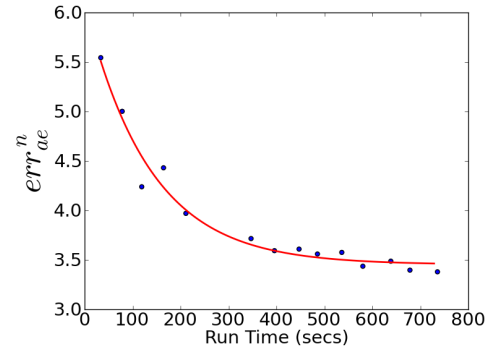


(a) Optical flow error against speed

(b) Stereo flow error against speed



(c) Structural error against speed



(d) Directional error against speed

Figure 4.7: Performance against runtime. Results obtained by varying the number of scene particles on the Cones dataset.

of 3.5 degrees. This highlights the fundamental limit of stochastic estimation systems for this error metric. The motion magnitude and structural error rates decay more slowly, and do not appear to have completely saturated, even at runtimes greater than 10 minutes per frame, indicating that the theoretical limit is far lower for these error metrics.

4.4.4 Search Space Analysis

The state space dimensions $\mathbf{\Phi}$ for the scene particles system, specify the range of structure and velocity values for which observations are possible, with a given the sensor setup. Scene particles which fall outside this range (either through diffusion, or due to

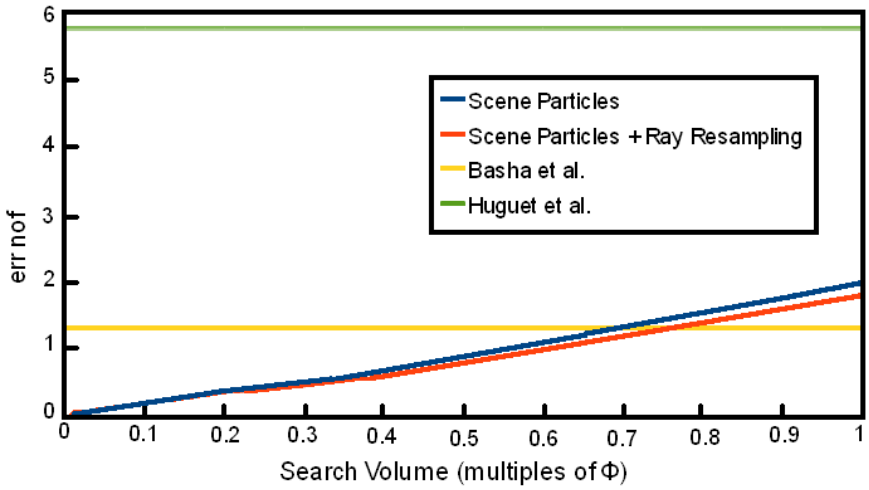


Figure 4.8: Search volume against performance. The search volume is listed as a fraction of $\mathbf{\Phi}$. The performance measure used is err^n_{of} (other error measures show the same linear trend), on the Cones dataset. For comparison the approaches of [19] and [64] are also displayed.

the transition between frames) receive a likelihood of 0, as they cannot be observed, and are subsequently removed during resampling. Utilising a smaller state space requires fewer scene particles to achieve the same sampling density. Thus, if application specific knowledge is available, for example that objects are unlikely to cover more than $\frac{1}{3}$ of the scene in a single frame, the state space and number of scene particles may both be reduced, to achieve a reduction in runtime with no cost to accuracy. Similarly, reducing the size of the state space while maintaining a constant number of scene particles (and hence runtime), leads to improved accuracy. Figure 4.8 demonstrates this effect.

The scene particles algorithm outperforms previous state of the art techniques, when exploring velocities up to 75% of the maximum state space $\mathbf{\Phi}$. Any scene particles with velocity values above this would cover most of the scene in a single frame, and are unlikely to be visible in the following frame. Thus in most realistic applications, these higher portions of the velocity space may be safely ignored. If it is necessary to search these areas of the state space, additional scene particles would be required to maintain top performance, increasing computational cost (although the approach would still be orders of magnitude faster than alternatives). Ray Resampling leads to

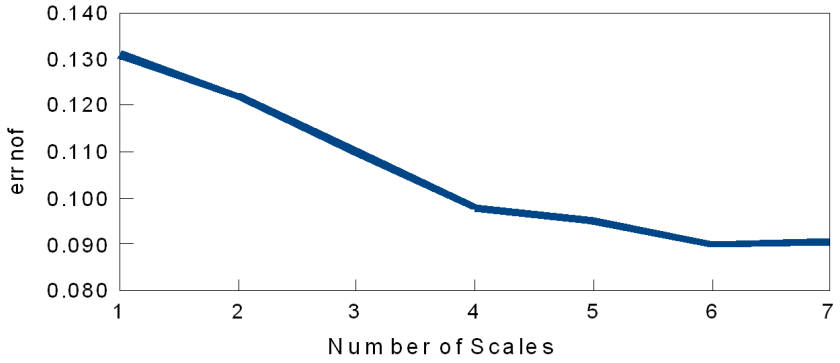


Figure 4.9: Pyramid scales against performance. The relationship between the number of image scales S and the optical flow error err^n_{of} on the Cones dataset. Other error measures exhibit the same behaviour.

a slight reduction in the growth rate, with increasing search space. This implies that Ray Resampling is able to more efficiently cover a given state space.

4.4.5 Iteration Analysis

In section 4.3.2 two iterative estimation schemes were introduced. The first is a multi-scale approach similar to that employed in variational motion estimation. The second iterative layer is based on modifying the diffusion behaviour, similar to PSO techniques. The benefit of these schemes is quantitatively analysed in figures 4.9 and 4.10 respectively. Both forms of iterative refinement show similar behaviours. As the number of iterations increases, the error rate drops. The use of multiple scales proves the most valuable, which is unsurprising as in addition to the scene particles having more time to converge, it provides additional "contextual" observations. As the number of iterations increases, performance gains tail off, which implies that most scene particles have found the maxima of the distribution. Thus remaining errors are likely due to mismatch between the likelihood function and the true motion field (due to the breaking of brightness constancy, sensor noise, occlusions, etc). Computational cost increases linearly with the number of iterations.

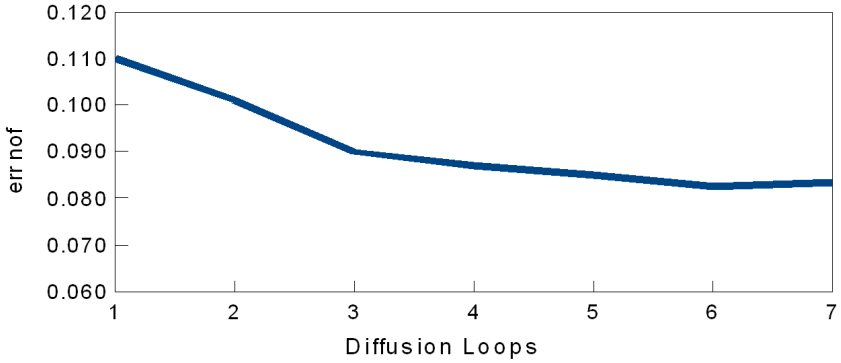


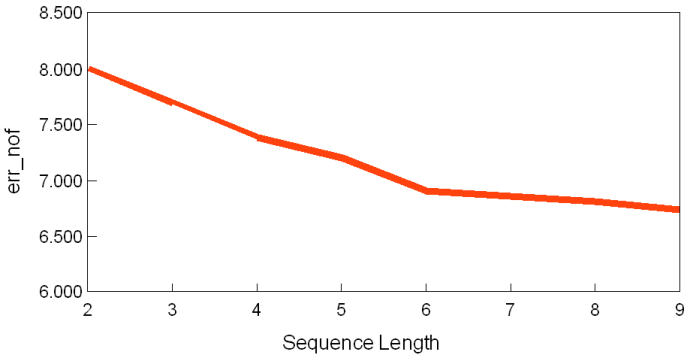
Figure 4.10: Diffusion loops against performance. The relationship between the number of diffusion loops and the optical flow error err^n_{of} on the Cones dataset. Other error measures exhibit the same behaviour.

4.4.6 Propagation Of Information

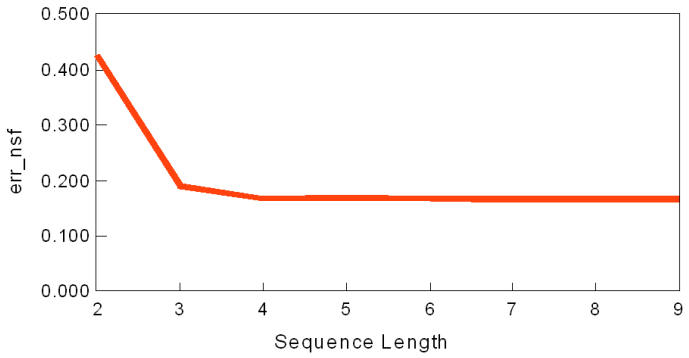
As the scene particle algorithm is several orders of magnitude faster than approaches to SFE, it is useful to analyse the behaviour of the approach on sequences longer than two frames. This is important for the application of the algorithm to real vision problems. In particular, such concerns are relevant for the action recognition application of chapter 3.

Figure 4.11 shows the performance of the algorithm on sequences of varying lengths, with error scores averaged across all frames. It can be seen that adding a single frame to the sequence causes a considerable reduction in both directional (figure 4.11c), and out of plane motion error (figure 4.11b). Subsequent frames also provide improved performance, but to a much less significant degree. Intuitively, this behaviour makes sense, as motion hypotheses rarely remain ambiguous across more than two sets of observations. The observed lower limit on the performance is likely due to regions emerging from occlusion, for which the prior information cannot provide any gains.

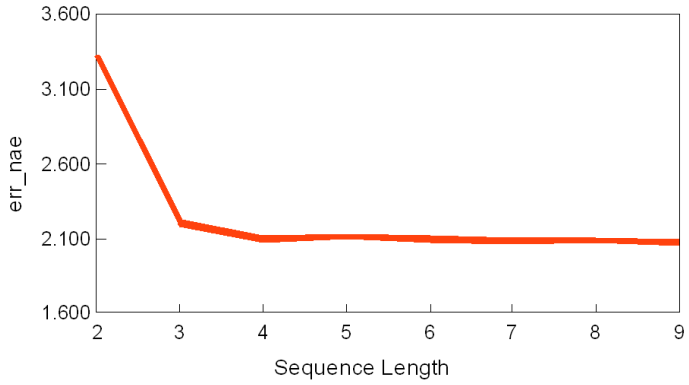
The optical flow error measure (figure 4.11a)does not saturate as rapidly as the other metrics, requiring sequences of around 6 frames before improvements become negligible. This suggests that image plane motions suffer more significantly from ambiguity, and require the accumulation of many observations before disambiguation becomes possible. This behaviour is surprising, and serves to reinforce the value of an estimation scheme



(a) Optical flow against sequence length



(b) Structural error against sequence length



(c) Directional error against sequence length

Figure 4.11: Sequence length against performance. Analysis of 3 error measurements on the cone dataset, when simulating sequences of various lengths. Experiments performed with a single appearance and depth sensor, using the standard scene particles algorithm.

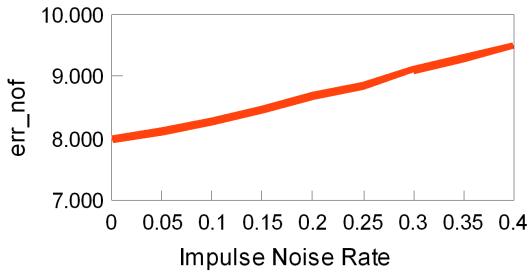
which is capable of propagating information through sequences.

4.4.7 Robustness To Noise

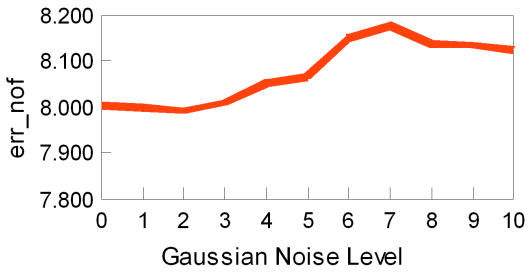
For the algorithm to be applicable to real data such as the action recognition dataset of chapter 3, it must be robust to noise from a range of sources, including sensor noise, transmission noise and compression artefacts. To quantitatively analyse this robustness, the input sequences were corrupted with varying levels of noise. Both Gaussian and "salt and pepper" noise were tested, with results shown in figure 4.12. For the Gaussian noise tests, every pixels value was modified by a Gaussian distributed offset. Performance was then analysed while varying the standard deviation of the noise distribution. For the salt and pepper noise experiments, a varying fraction of pixels were randomly selected, and set to either 0 or 255. In both cases, viewpoints were treated independently.

The algorithm performs very well when subjected to salt and pepper noise. All error metrics increase linearly with the number of corrupted pixels, across the full range of noise levels (figures 4.12a,4.12c and 4.12e). Due to the independent nature of the scene particles, it is impossible to estimate the motion of a corrupted pixel, but uncorrupted pixels remain unaffected. In contrast, a variational or patch based approach may be expected to have a more gradual linear relationship at low noise levels, as some corrupted pixels may be estimated correctly due to smoothness constraints. However, at higher levels of noise, such global approaches are likely suffer from catastrophic failure.

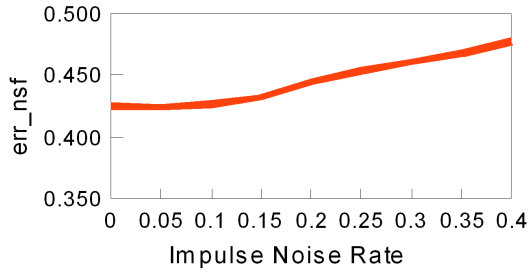
The performance of the system under Gaussian noise (figures 4.12b,4.12d and 4.12f) is less consistent, although still generally displaying a linearly increasing trend. In absolute terms, Gaussian noise causes less degradation of the estimated motion than salt and pepper noise, with a standard deviation of 10 intensity values being equivalent to around 15% salt and pepper noise corruption. This is likely due to the multi-scale approach employed, as the smoothing applied to create coarser image scales, also reduces the effect of noise.



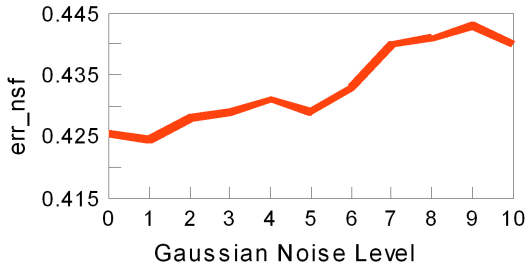
(a) Optical flow error under salt and pepper noise



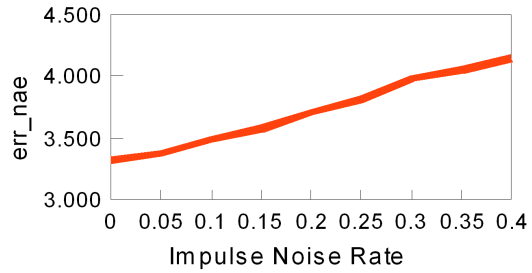
(b) Optical flow error under Gaussian noise



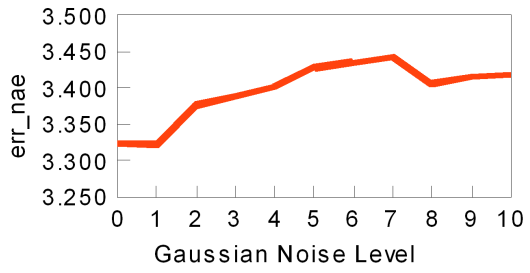
(c) Structure error under salt and pepper noise



(d) Structure error under Gaussian noise



(e) Directional error under salt and pepper noise



(f) Directional error under Gaussian noise

Figure 4.12: Noise robustness. Three error measurements, as a function of noise level. The left column shows salt and pepper noise performance, while the right column is the Gaussian additive noise performance. Tests performed using a single appearance and depth sensor, on the cones dataset, with the standard scene particles algorithm.

4.5 Object Tracking Application

The scene particle algorithm allows 3D motion fields to be estimated more rapidly than existing variational approaches by several orders of magnitude, while also providing temporally consistent results for long sequences. These two features make SFE a more feasible prospect for solving practical computer vision problems, rather than being of academic interest only, as it has in the past. Here an example application is demonstrated, using SFE to perform 3D object tracking. For such tracking and segmentation tasks, the lack of oversmoothing artefacts in the scene particle algorithm has the additional benefit of providing clear object boundaries.

There are generally 3 approaches to tracking. Fitting a model of the object in question to the observations, tracking a collection of distinctive feature points on the object, and using a data-driven approach to choose a representation where objects of interest naturally emerge from the data. This work fits into the third category, which gives it the useful property that objects need not be known or modelled before tracking. The approach has similarities to tracking by colour distributions [84], which is common for hand tracking, and more recent work on tracking using optical flow [49, 104]. In comparison to these techniques, tracking by scene flow is more informative, as the output describes how the object is moving in the 3D world, rather than how it moves on the image plane after projective distortions.

4.5.1 Object Extraction

In order to extract objects from the scene flow field, the scene particles are clustered in 6D, i.e. in both location and motion dimensions. This not only creates clusters for consistent collections of 3D structure points, but is also able to separate objects which are close to each other, but moving in distinct directions. For the purposes of this example application, clustering is performed via expectation maximization, fitting a collection of 6 dimensional Gaussians to the scene flow field. The advantage of this approach over simpler alternatives such as K-Means, is that it allows objects which are of differing sizes, and have different extents along each dimension, as opposed to

clusters being spherical. However, the number of clusters must still be specified, rather than emerging naturally the data.

4.5.2 Trajectory Estimation

In order to construct trajectories from the clusters at each frame, a consistent set of identity labels must be assigned. To achieve this, at each frame predicted cluster centres are generated from each clusters previous state, using a constant velocity motion model. Then a greedy assignment is performed to match the labels of the prediction to the newly estimated clusters. The cluster predictions are also used to initialise the clustering of the following frame. At the first frame clusters need to be initialised either randomly, or using some task specific information.

The clusters are 6 dimensional, and as such the cluster centre defines not only the centre of the object in space, but also its average velocity, which can be used during the prediction stage. In algorithms where motion estimates are not available, it is common to estimate the velocity using the difference between the previous two centroids. This is less reliable however, as objects may change shape over time, making the centroids unstable.

4.5.3 Hand Tracking During Sign Language

To analyse the performance of the tracking system based on scene flow estimation, the specific task of tracking hands and head during sign language was chosen. This is a useful application area, as the 3D trajectories can be expected to provide valuable information for sign language recognition. Additionally, the number of objects to be tracked is known, and model based approaches are extremely difficult due to the objects' high levels of articulation.

Figure 4.13 provides an overview of the approach. The scene flow estimation system processes the input data from the multi-camera rig, and produces an estimate of 3D scene structure, along with an estimated 3D motion field. Additional task specific information is included in the system, in the form of an extra term in the likelihood

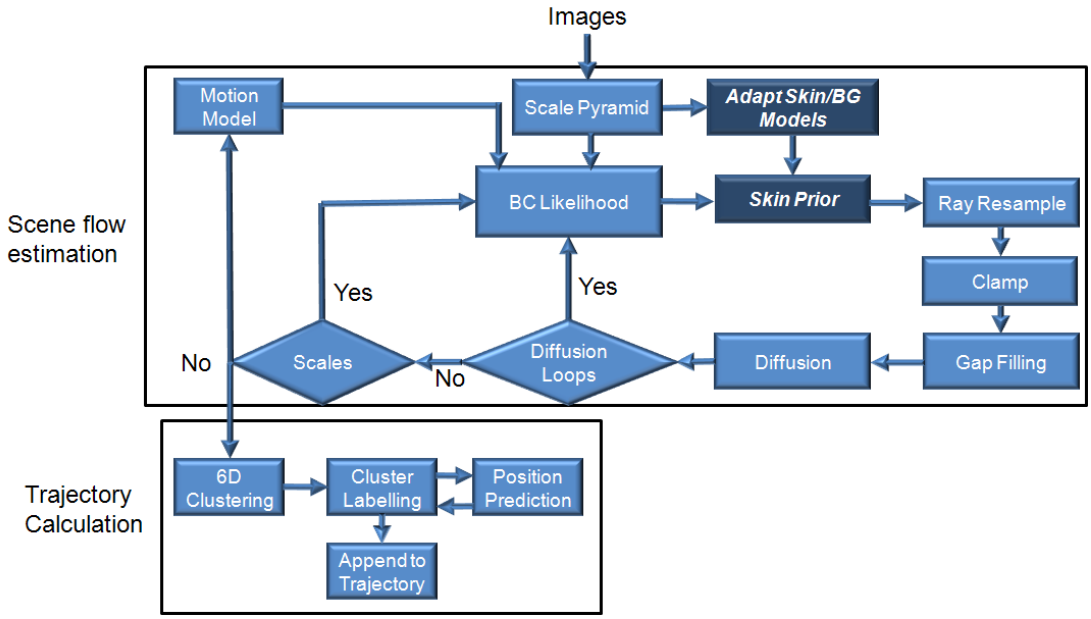


Figure 4.13: Trajectory estimation block diagram. Modifications to the original scene particle algorithm are shown in darker blue, and the interpretation stage has been replaced with the trajectory estimation system.

equation 4.7 which focuses accuracy in relevant regions. This term is defined in equation 4.29 and relates to the likelihood that each combination of $\boldsymbol{r}$ and $\boldsymbol{v}$ belong to an object of interest, based on an adaptive skin colour model. By concentrating scene flow estimation into regions of interest, a reduced number of scene particles can be used, further improving computation times. The initialisation of the clustering at the first frame is performed using the extremities of the motion field, which is more appropriate than random selection for sign language tasks.

The skin colour model comprises of 2 probability distributions in RGB colour space, one for skin colour $\boldsymbol{p}_{sk}(\bar{I})$ and one for background colour $\boldsymbol{p}_{bg}(\bar{I})$. For a point in the scene space $\boldsymbol{r}, \boldsymbol{v}$, with colour vector $\bar{I}$, the extra likelihood term $\boldsymbol{p}_s(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v})$ is calculated using the log-likelihood ratio of the skin and background models. By averaging the responses across viewpoints, the system obtains resistance to single view occlusions.

$$\boldsymbol{p}_s(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v}) = \frac{1}{M} \sum_{m=1}^{M} \min \left(\log_{10} \left(\frac{\boldsymbol{p}_{sk}(\bar{I})}{\boldsymbol{p}_{bg}(\bar{I})} \right), 1 \right) \tag{4.29}$$

Both models $\boldsymbol{p}_{sk}(\bar{I})$ and $\boldsymbol{p}_{bg}(\bar{I})$ are initialised from a generic colour model, and then allowed to adapt to the data, based on the results of face detection. The colour distribution within the central region of the face detection is used to update $\boldsymbol{p}_{sk}(\bar{I})$, and the distribution across the remainder of the image is used to update $\boldsymbol{p}_{bg}(\bar{I})$.

4.5.4 Hand Tracking Evaluation

Evaluating the performance of a passive 3D tracking system is difficult, as the ground truth 3D position for moving objects in real scenes is rarely available. This is particularly true in the sign language scenario, where the objects in question move and deform rapidly. There is no currently available sign language dataset which provides this ground truth, thus results were obtained on newly captured multi-view dataset, where 2D trajectories were available from image plane tracking. Figure 4.14 shows example frames from parts of this dataset, with the 3D trajectories projected to every viewpoint. It can be seen that the estimated 3D trajectories of all objects are consistent with the observations in every viewpoint. In total, 2.8 million frames from the dataset were tracked, relating to over 31 hours of 3D sign language trajectories. This application would have been intractable for traditional SFE techniques, requiring around 1600 years to complete. In contrast the scene particle algorithm would have required 2 years for a sequential implementation, making it feasible to split the processing across only a small number of machines, in order to obtain results within a week. In this application, the scene particles algorithm was able to run faster than shown in table 4.1 (around 30 seconds per frame), as the skin colour prior enabled good tracking results to be obtained with only 4 particles per ray, 3 scales ($S = 3$) and 3 diffusion iterations. No calibration information was provided with this dataset, necessitating the use of manually labelled correspondence points to estimate the camera parameters. This serves to demonstrates that the technique provides some degree of robustness to imprecise calibration.

Figure 4.14 parts A-C are taken from a narrow baseline, 2 view, sequence. Figure 4.14.C illustrates a failure case, where the narrow baseline creates some ambiguity in the Z dimension. The resulting trajectory matches the data well when projected to the left viewpoint, however the projection in the right sensor shows an erroneous 1 frame jump

Object	Agreement	X RMS error	Y RMS error
Head	100%	0.057	0.097
Right Hand	93.535%	0.191	0.100
Left Hand	88.054%	0.277	0.091

Table 4.3: Hand tracking performance. Agreement between projection of estimated 3D trajectories, and 2D trajectories from an alternative system (values in palm widths).

towards the face. Wider baseline systems such as that shown in figure 4.14.D are able to resolve such ambiguities. Additionally the third viewpoint in this case allows the system to operate robustly in the case of occlusions. Despite being completely occluded in the side view for several frames, the estimated trajectory for the right hand is consistent with the observations from the other 2 views.

Quantitatively, the system was analysed by comparing the projection of the 3D trajectories to the high accuracy 2D tracking provided for the data [96]. This tests the consistency of the trackers, and can be seen as providing a lower bound on performance, as some inconsistencies will arise from errors in the 2D tracker, due to frontal occlusions, while the 3D tracking is maintained using the other views. Trajectories were assumed to be in agreement, if the separation of their predicted location was less than one third of a palm width. The percentage of frames where trajectories are in agreement is listed for each object in table 4.3, as is the RMS distance between trajectories, in terms of palm widths. Results were compared over 30,000 tracked frames, the close agreement between the systems indicates that the 3D tracking is plausible. If the system was inaccurate, it is unlikely it would coincide with the 2D system so frequently. The greater accuracy of the head trajectory is to be expected, as it moves little during the sequences. The accuracy of the left hand is slightly higher than the right, as one viewpoint is placed on this side, and so the hand is occluded less often.

(A)

(B)

(C)

(D)

Figure 4.14: Hand tracking example results. (A) to (C) are taken from the narrow baseline, 2 view, sequence, showing the projection of 3D trajectories which are consistent with both viewpoints. (C) Illustrates a failure case where the narrow baseline of the 2 views gives rise to ambiguity in Z. (D) Shows a frame from the wide baseline, 3 view, sequence, illustrating robustness to object occlusions.

4.6 Conclusions

This chapter focussed on a particle filter based technique for the estimation of dense 3D motion fields, with the aim of providing a rich description of dynamic scenes. In contrast to existing techniques, the work in this chapter focussed on practical applicability. By maintaining multiple hypotheses, the approach was able to avoid common oversmoothing artefacts which damage the estimated motion in boundary regions. These regions tend to be the most salient in vision tasks, including action recognition. Additionally, the technique was shown to provide quantitatively improved results on benchmark scene flow datasets.

The formulation of the scene particles algorithm is flexible, allowing any combination of appearance and depth sensors, in any setup. This, coupled with runtimes several orders of magnitude faster than previous state of the art techniques, makes 3D motion estimation feasible in a wider variety of vision tasks, and also applicable to realistic sized datasets, such as the Hollywood 3D dataset. Good robustness to various types of sensor noise was demonstrated, and the propagation of information was evaluated, leading to temporally consistent results.

To demonstrate the suitability of the approach to real vision tasks, an example application was formulated, for the 3D tracking of hands during sign language. Due to the speed of the motion estimation system, tracking could be trivially performed on a vast amount of data (2.8 million frames, or 31 hours of HD video), consisting of multiple camera combinations and setups. The success of this application, demonstrates that the scene particles algorithm is flexible enough to cope with a wide range of motions, scenes and sensor configurations, which will be important in the action recognition task, using the Hollywood 3D dataset.

However, despite the flexibility of the algorithm, and its suitability for real vision tasks, there is still room for improvement in accuracy. The current formulation is based on the naive assumption of brightness constancy, which is known to be violated in real situations. Additionally, smoothness constraints are not utilised, despite being generally valid. This leads to valuable properties, such as producing clear object boundaries without oversmoothing. However, it is reasonable to ask, whether these (and other)

constraints may be incorporated, to improve accuracy while preserving these properties.

Chapter 5

Revisiting Motion Estimation Assumptions

Despite the flexibility and speed of the scene particles algorithm from chapter 4, improvements in accuracy may be possible, by examining the fundamental assumptions used in the formulation. For instance, improvements may be made to the likelihood, which was based on a naive linear relationship with the brightness constancy assumption. Also, additional assumptions may be incorporated, such as the smoothness assumption, and the constant velocity assumption.

5.1 Smoothed Scene Particles

The original impetus for the scene particles algorithm, was to avoid oversmoothing artefacts at discontinuities, by removing smoothness constraints and instead utilising multiple hypothesis, and the temporal accumulation of constraints. However, the assumption of motion field smoothness *is* generally valid, across large areas of the scene. Following are 3 different approaches for incorporating smoothness information into the scene particle algorithm.

5.1.1 Patch Based Scene Particles

Lucas and Kanade [81] formulated one of the earliest optical flow algorithms, which accumulated flow constraints for each pixel within a local neighbourhood or "patch". This technique leads to smoothness of the motion field (as neighbouring patches contain many of the same pixels) while being less stringent than the variational approaches developed from the work of Horn and Schunk [63]. In both cases, motion estimates at discontinuities are likely to be degraded. However, for local techniques, the extent of this degradation is limited by the patch size, while under variational schemes it may affect the entire motion field. The scene particles algorithm may be adapted to follow such a local scheme, by modifying the likelihood term of equation 4.7, to sum pixel-wise likelihoods over a 2D neighbourhood Ω. This neighbourhood is extracted around the projection of the particle in each sensor, and scores are calculated in conjunction with the mean patch $\bar{\boldsymbol{I}}_p$. This modifies the algorithm to consider some spatial constraints, in addition to the temporal constraints obtained via the prior distribution. Again, a similar modification may also be made for the depth likelihood term of equation 4.8.

$$\boldsymbol{p}_p(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v}) = \frac{1}{1 + e_a \sum_{\boldsymbol{o} \in \Omega} \sum_{\tau=t\text{-}1}^{t} \sum_{m=1}^{M} \frac{\left(\boldsymbol{I}^{\tau}_{a,m}(\boldsymbol{\Pi}_m(\boldsymbol{r}_\tau) + \boldsymbol{o}) - \bar{\boldsymbol{I}}_p(\boldsymbol{o})\right)^2}{2M|\Omega|}} \tag{5.1}$$

This new function can be seen as an extension of the Sum of Squared Differences (SSD) similarity score. Standard SSD metrics encode the pixel-wise differences between 2 patches. In contrast, equation 5.1 compares the pixel-wise consistency of any number of patches (including 2), which allows the metric to be employed with any number of sensors.

An important limitation of this approach however, is that it assumes all sensors are in an approximately parallel set-up, which drastically limits the flexibility of the algorithm. Additionally, there is an inherent assumption that small regions of the scene are approximately planar and fronto-parallel, such that patches on the image plane of one sensor, correspond to similarly shaped patches in all other sensors.

5.1.2 Image Plane Smoothing

The patch based scheme directly enforces smoothing in the scene particle cloud. As a result, oversmoothing artefacts are likely to accumulate over time due to the propagation of smoothness information through the prior. An alternative approach is to employ smoothness constraints only during the interpretation stage (i.e. when averaging out the multiple hypothesis) while leaving the underlying scene particle cloud intact.

The simplest way to achieve this, is to follow the suggestions of previous authors [55, 122], and apply 2D smoothing to the image plane projection of the scene flow. This can be achieved by smoothing the 4 channel output image $\boldsymbol{I}_o$, produced during interpretation, as described in section 4.3.4. The values in each channel of $\boldsymbol{I}_o$ (relating to v_x, v_y,v_d and d) are treated independently. In order to minimize the oversmoothing at discontinuities, the edge preserving bilateral filter is used. Bilateral filtering is a biologically inspired technique, similar to the operation of the human visual system. Put simply, the bilateral filter performs smoothing by averaging values across a region, with weightings of each pixel determined by both a spatial Gaussian and a "value" Gaussian. The spatial Gaussian weighting, is the equivalent to a standard convolution with a Gaussian kernel. The "value" Gaussian re-weights this combination, based on value dissimilarity. The outcome of this is that pixels with significantly different values to the central pixel, contribute less to the average, while most smoothing is applied to values that are already nearly consistent. Intuitively this means that when the smoothing kernel overlaps a discontinuity, the values on the same side of the discontinuity as the central pixel have increased weighting, while the values on the other side have less impact.

Equation 5.2 formalises this idea, with the filtered value $\hat{\boldsymbol{I}}_o(x, y)$ at pixel x, y being the result of a weighted averaging over all pixels $\boldsymbol{I}_o((x, y) + \boldsymbol{o})$ with 2D offsets $\boldsymbol{o}$ lying in a neighbourhood Ω. The contribution of each pixel is determined partly by the magnitude of the offset $\boldsymbol{o}$, with a zero mean Gaussian weighting function (written as $g(|\boldsymbol{o}|; 0, \sigma_{i2})$), with variance σ_{i2}, to provide spatial weighting. The second term affecting the contribution of each pixel, is based on the difference of that pixels value to the central pixel, using a second zero mean Gaussian with variance σ_{v2}. The total

weight over the neighbourhood Ω_w is then used to normalised the filter response.

$$\hat{\boldsymbol{I}}_o(x,y) = \frac{1}{\Omega_w} \sum_{\boldsymbol{o}\in\Omega} \boldsymbol{I}_o\left((x,y)+\boldsymbol{o}\right) g\left(|\boldsymbol{o}|;0,\sigma_{i2}\right) g\left(\boldsymbol{I}_o(x,y) - \boldsymbol{I}_o\left((x,y)+\boldsymbol{o}\right);0,\sigma_{v2}\right) \tag{5.2}$$

$$\Omega_w = \sum_{\boldsymbol{o}\in\Omega} g\left(|\boldsymbol{o}|;0,\sigma_{i2}\right) g\left(\boldsymbol{I}_o(x,y) - \boldsymbol{I}_o\left((x,y)+\boldsymbol{o}\right);0,\sigma_{v2}\right) \tag{5.3}$$

Using bilateral filtering greatly reduces smoothing artefacts, compared to a simple Gaussian filter. The value of σ_{i2} relates to the level of smoothing, i.e. the level of high frequency information removed, as in standard Gaussian smoothing. The value of σ_{v2} corresponds to the level of "jump" expected at discontinuities. Edges with a change of twice σ_{v2} or more will suffer from little oversmoothing, while lower contrast edges are heavily smoothed. Obviously this is an improvement over standard Gaussian convolution, however it requires the selection of two parameters, and small details with low contrast, such as wrinkles in clothing, are still unlikely to be preserved.

5.1.3 3D Smoothing

When applying smoothing as a post processing step, it may be more sensible to smooth the scene particle cloud P directly, rather than its 2D projection $\boldsymbol{I}_o$. Smoothness assumptions are more likely to be valid in the original scene, before projective distortions. Additionally, points with significantly different distances to the sensor, would no longer affect each other simply because they project to neighbouring pixels. To achieve this, the 2D bilateral filtering may be adapted, so as to perform weighted averaging of scene particles, within a 3 dimensional neighbourhood.

To achieve this without accumulating oversmoothing errors, the unsmoothed scene particle cloud P is preserved, and used to generate the prior for the next frames estimation. A second, smoothed, scene particle cloud $\hat{P}$ is also created, and used as input to the interpretation stage, then discarded. Every scene particle $\boldsymbol{P}_n$ in the original cloud consists of $(\boldsymbol{r}_n, v_{\dot{x}n}, v_{\dot{y}n}, v_{\dot{z}n})$, and has a smoothed equivalent $\hat{\boldsymbol{P}}_n$ consisting of

$(\boldsymbol{r}_n, \hat{v}_{\dot{x}n}, \hat{v}_{\dot{y}n}, \hat{v}_{\dot{z}n})$. Equation 5.4 shows how the smoothed $\hat{v}_{\dot{x}}$ value is calculated for particle $\hat{\boldsymbol{P}}_n$, via the weighted averaging of all scene particles, with a 3D offset $\boldsymbol{\eta}$ lying inside the neighbourhood Λ. Following the bilateral filtering approach, the weighting of each neighbouring scene particle is determined, in part, by the magnitude of its offset, Gaussian weighted with a variance of σ_{i3}. The 3 velocity dimensions are then integrated out, so that all motion hypotheses at this position $(\boldsymbol{r}_n + \boldsymbol{\eta})$ are considered. In addition to the standard Gaussian "value" weighting (using the variance σ_{v3}) the weight of each hypothesis (i.e its sample from the posterior) is considered, in order to simulate the weighted averaging of the original approach. As in the 2D case, Λ_w equates to the total weighting for the patch, and is used to normalise the filter response.

$$\hat{v}_{\dot{x}n} = \frac{1}{\Lambda_w} \sum_{\boldsymbol{\eta} \in \Lambda} g\left(|\boldsymbol{\eta}|; 0, \sigma_{i3}\right) \iiint_{v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}} v_{\dot{x}} \boldsymbol{p}(\boldsymbol{r}_n + \boldsymbol{\eta}, v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}} \mid \boldsymbol{I}) g\left(v_{\dot{x}n} - v_{\dot{x}}; 0, \sigma_{v3}\right) \, \mathrm{d}v_{\dot{x}} \mathrm{d}v_{\dot{y}} \mathrm{d}v_{\dot{z}} \tag{5.4}$$

$$\Lambda_w = \sum_{\boldsymbol{\eta} \in \Lambda} g\left(|\boldsymbol{\eta}|; 0, \sigma_{i3}\right) \iiint_{v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}} \boldsymbol{p}(\boldsymbol{r}_n + \boldsymbol{\eta}, v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}} \mid \boldsymbol{I}) g\left(v_{\dot{x}n} - v_{\dot{x}}; 0, \sigma_{v3}\right) \, \mathrm{d}v_{\dot{x}} \mathrm{d}v_{\dot{y}} \mathrm{d}v_{\dot{z}} \tag{5.5}$$

Similar equations can be created to provide the smoothed values $\hat{v}_{\dot{y}n}$ and $\hat{v}_{\dot{z}n}$ necessary to construct each $\hat{\boldsymbol{P}}_n$. As in the 2D case above, the variance σ_{i3} relates to the strength of the smoothing (the amount of high frequency features removed) while the variance σ_{v3} affects the size of allowable (unsmoothed) discontinuities.

5.2 Smoothing Evaluation

Table 5.1 compares the different smoothing approaches, while figure 5.1 shows qualitatively the effect 3D post smoothing has on the motion field. Unsurprisingly, much of the low level "jittering" seen in figures 5.1a and 5.1e is removed in the smoothed estimate of figures 5.1b and 5.1f, leaving the general motion trends in each region. Due to the 3D formulation and use of bilateral filtering, there is also little oversmoothing visible

at discontinuities. However, the smoothing also removes fine motion details along with the jitter, for example different areas of clothing no longer move in different directions, and wrinkles are no longer visible.

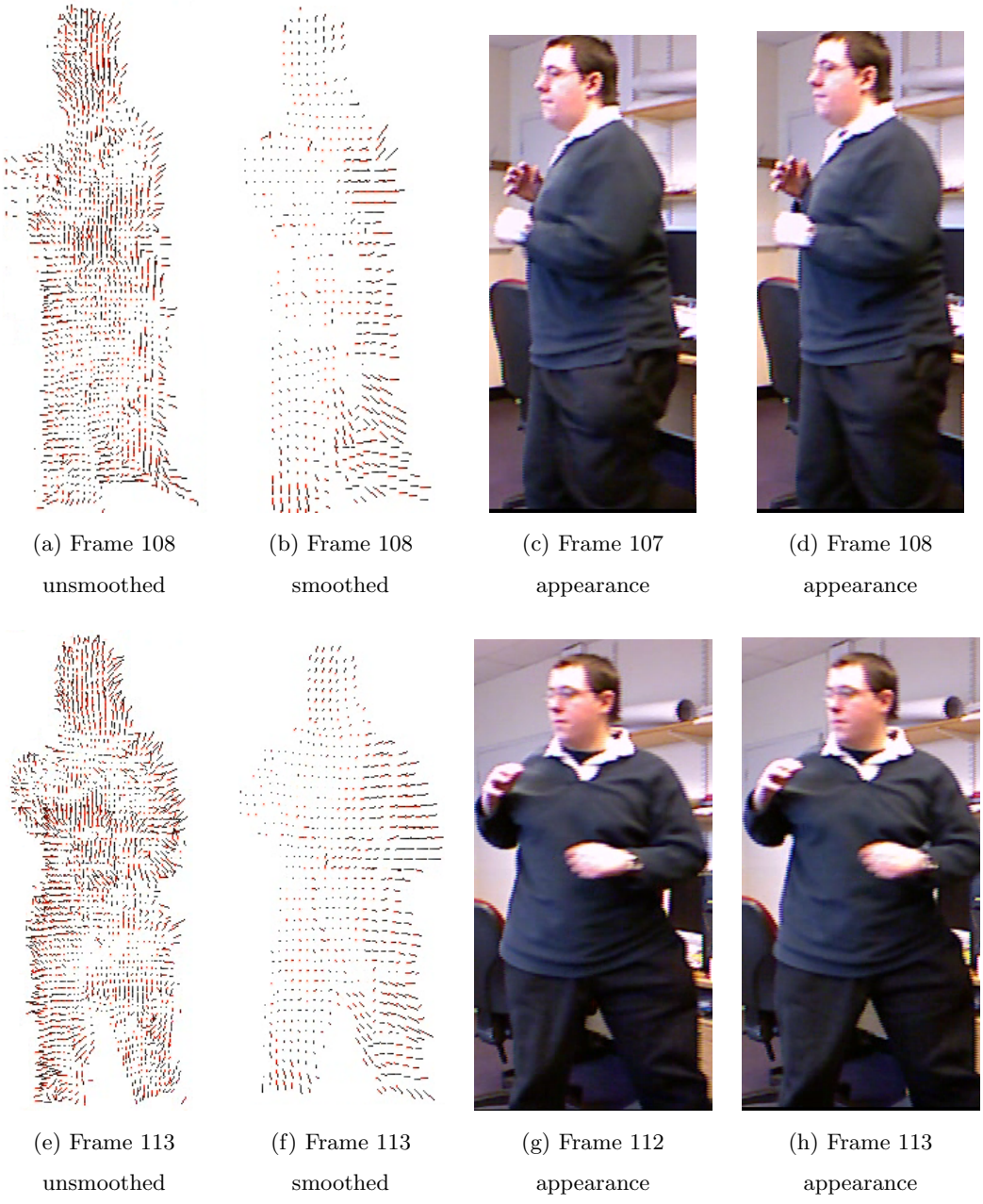
The results for patch based matching costs and 2D post filtering, show little improvement in terms of motion magnitude accuracy. However, applying 3D post filtering shows a significant increases both in magnitude and directional accuracy can be obtained, although this comes at the cost of doubled computation time.

The post filtering techniques are applied only to the motion field, and as such the structural performance does not change. However, using a patch based matching cost significantly reduces the accuracy of structural estimation, to such a degree that it becomes comparable with the previous state of the art approaches from table 4.1. As discussed in the previous chapter, this is due to the loss of fine structure and the introduction of over-smoothing artefacts at discontinuities. Additionally, directional accuracy is improved, bringing it closer to the levels of these previous techniques. This effect was also discussed in section 4.4.1 and is produced by the removal of the low level "jitter" which occurs in unsmoothed motion field. This small amount of motion noise leads to large directional errors, particularly in regions of low motion such as background regions. These findings are extremely interesting, and implying a fundamental similarity between global variational schemes [19, 64], and the local patch based smoothing employed here. Further, it demonstrates that the current incorporation of smoothness constraints during optimisation is a limiting factor in current motion estimation schemes. This remains true when incorporated in the scene particle algorithm, despite the addition of multiple hypothesis and information propagation.

It is useful to examine how the scale of the smoothing affects this behaviour. The results in table 5.1 were obtained using a local 2D neighbourhood Ω equating to an 5 by 5 patch, with the spatial variance σ_{i2} set to $\frac{5}{3}$ such that 99% of the area under the Gaussian is represented in the kernel. For the 3D filtering approach, the neighbourhood Λ was set to cover a region equivalent to 0.5% of the scene range $\mathbf{\Phi}$, with the spatial variance σ_{i3} determined in the same manner. Figure 5.2 explores the effects of varying these parameters, while figure 5.3 displays the related effect of varying the scale of the

Algorithm	Dataset	err^n_{of} ($\times 10^{-2}$)	err^n_{sf} ($\times 10^{-2}$)	err^n_{st}	err^n_{ae}	Runtime
SP	Cones	5.8	0.2	**0.294**	4.08	213 secs
SP (Patch Cost)	Cones	5.7	0.2	5.38	**3.26**	931 secs
SP (2D Post Filter)	Cones	5.7	0.2	**0.294**	4.17	**211 secs**
SP (3D Post Filter)	Cones	**5.4**	**0.1**	**0.294**	3.81	533 secs
SP	Teddy	4.2	0.2	**0.198**	4.23	**207 secs**
SP (Patch Cost)	Teddy	4.1	0.2	5.10	**3.03**	909 secs
SP (2D Post Filter)	Teddy	4.1	0.2	**0.198**	4.28	209 secs
SP (3D Post Filter)	Teddy	**3.8**	**0.1**	**0.198**	3.77	528 secs
SP	Venus	3.6	**0.1**	**0.246**	4.26	**213 secs**
SP (Patch Cost)	Venus	3.6	**0.1**	5.31	**3.41**	944 secs
SP (2D Post Filter)	Venus	3.7	**0.1**	**0.246**	4.25	215 secs
SP (3D Post Filter)	Venus	**3.3**	**0.1**	**0.246**	3.88	532 secs

Table 5.1: Smoothed scene particle performance. The performance of the scene particles algorithm when incorporating smoothness constraints in a variety of ways. Tests are performed with occlusion aware scene particles, using 4 appearance sensors.



(a) Frame 108 unsmoothed

(b) Frame 108 smoothed

(c) Frame 107 appearance

(d) Frame 108 appearance

(e) Frame 113 unsmoothed

(f) Frame 113 smoothed

(g) Frame 112 appearance

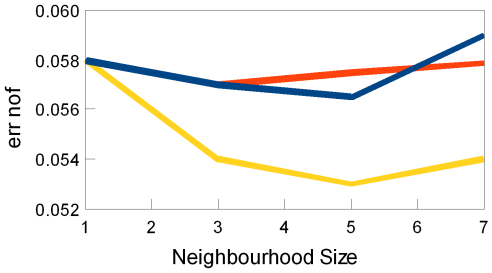
(h) Frame 113 appearance

Figure 5.1: Smoothed flow field examples. Comparison of estimated flow fields, for 2 frames of the "Kicks" Kinect sequence, using the standard scene particles algorithm (figures 5.1a and 5.1e), and using 3D bilateral post-smoothing (figures 5.1b and 5.1f). Estimated motions travel from the light red to the black vertices. Also shown are the frames before (figures 5.1c and 5.1g) and after (figures 5.1d and 5.1h) the motion.

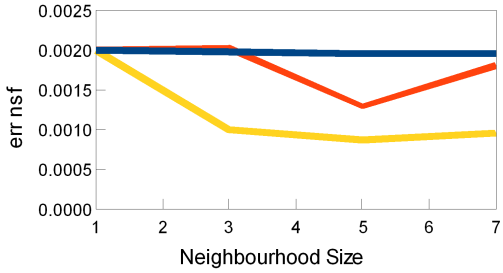
value Gaussians σ_{v2} and σ_{v3} which were originally both set to 0.5% of the scene range $\mathbf{\Phi}$.

As the spatial smoothing scale increases, the behaviour of the patch based system noted above, continues. Structural estimation (figure 5.2c) continues to degrade, while directional accuracy (figure 5.2d) continues to improve, such that performance is much closer to previous state of the art techniques. The directional error of the 3D post filtering increases with scale in a similar manner, however the 2D post filtering seems to provide little directional accuracy regardless of scale. In general, the motion magnitude error measures (figures 5.2a and 5.2b) exhibit optimal performance with a neighbourhood size of 5, regardless of the smoothing technique. Subsequent increases in scale lead to reduced performance, likely as the value of additional spatial constraints are outweighed by oversmoothing artefacts. Another important point to note, is that increasing the neighbourhood size leads to an exponential increase in the computational cost. For 2D post-filtering this is negligible, but for patch based and especially for 3D post-filtering, this is a concern.

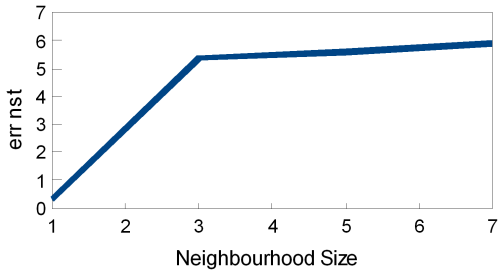
The value smoothing scale, exhibits similar behaviour to that observed for spatial smoothing scale (figure 5.3). Performance improves as the smoothing scale increases, up to a point, after which increasing the scale leads to reduced performance. However, although the behaviour is the same, the causes are reversed. At low scales, a very small difference between neighbouring motions is sufficient for the additional constraints to be attenuated, meaning very little smoothing is performed. As the scale increases, more and more information from neighbouring motions is incorporated, improving performance. However when the scale becomes too high, the useful properties of the bilateral filter are lost, and discontinuities are ignored when considering smoothness, giving rise to oversmoothing artefacts. In general the peak performance is found at at one or two times the initially tested value (meaning 0.5%-1% of the scene range $\mathbf{\Phi}$). Filtering in 2D appears to gain less benefit from changes in value smoothing scale, reflecting the results from table 5.1. In general, for both approaches, changes to the value smoothing have less effect than changes in spatial smoothing. However it is useful to note increasing the value smoothing scale incurs no additional computational cost, unlike spatial smoothing.



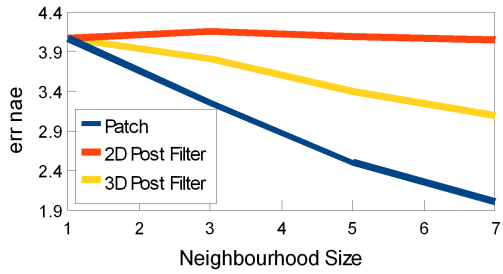
(a) Optical flow error against spatial smoothing scale



(b) Stereo flow error against spatial smoothing scale

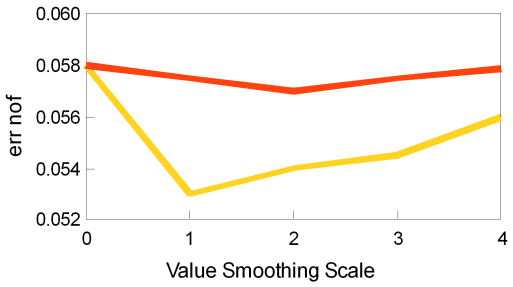


(c) Structure error against spatial smoothing scale (note Post Filtering does not affect structure)

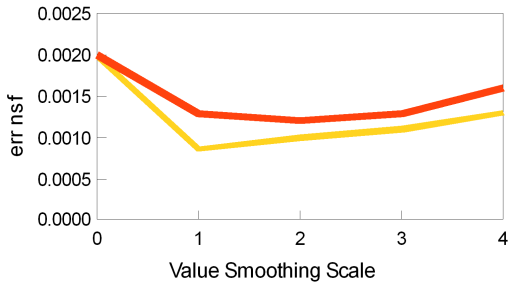


(d) Directional error against spatial smoothing scale

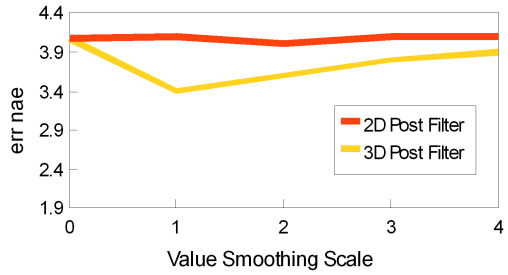
Figure 5.2: Spatial smoothing vs performance. The change in scene flow estimation accuracy, as a function of spatial smoothing scale, for patch based, 2D bilateral and 3D bilateral smoothing approaches. The x axis specifies the kernel or patch size, with the variance of the spatial Gaussians set as above, to encompass 3 standard deviations within the kernel. Note that lower error scores (y values) relates to better performance.



(a) Optical flow error against value smoothing scale



(b) Stereo flow error against value smoothing scale



(c) Directional error against value smoothing scale

Figure 5.3: Value smoothing vs performance. The change in scene flow estimation accuracy, as a function of value smoothing scale, for 2D and 3D bilateral smoothing. The x axis specifies variance of the value Gaussian (σ_{v2} and σ_{v3} respectively) as multiples of the original value defined above. The spatial smoothing neighbourhood was fixed at 5. Note that lower error scores (y values) relates to better performance. Structure reconstruction performance omitted as it is unaffected by post filtering.

5.3 Forwards and Backwards Scene Particles

In addition to incorporating the smoothness assumption, a simple modification to the original likelihood formulation 4.7, allows the assumption of Constant Velocity to be used. This modified likelihood,

$$\boldsymbol{p}_a(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v}) = \frac{1}{1 + e_a \sum_{m=1}^{M} \sum_{\tau=t\text{-}1}^{t+1} \frac{\left(\boldsymbol{I}^{\tau}_{a,m}(\boldsymbol{\Pi} m(\boldsymbol{r}_\tau)) - \bar{I}\right)^2}{3M}} \tag{5.6}$$

favours motions which are consistent from the previous to current frame, and also consistent from the current to future frame. This constant velocity assumption is often used in tracking systems, and is generally assumed to be satisfied if the framerate is high enough, such that changes in velocity between timesteps are near zero. The new formulation provides additional constraints, and reduces the ambiguity between motions. As such, it serves a similar purpose to the propagation of information via the prior probability, explored in section 4.4.6. However, this "Forwards-Backwards" formulation allows immediate gains in accuracy, rather than requiring a number of frames to pass to build up a reliable prior. Such a system does necessitate a frame buffer to achieve this however, which introduces a 1 frame lag between an input image and its motion field estimate. This lag is unlikely to be important in most applications however, as the assumption of Constant Velocity already implies a high enough framerate for acceleration to be negligible.

Note that $\boldsymbol{r}_{\mathcal{T}+1}$ is given by $\boldsymbol{r}_{\mathcal{T}} + \boldsymbol{v}$. A similar modification may also be made to the depth likelihood of equation 4.8. Section 5.5 evaluates the effects of this assumption. Further, the performance under varying degrees of violation for the Constant Velocity assumption, is explored in section 5.6.

5.4 Occlusion Aware Scene Particles

In addition to introducing new assumptions such as smoothness, improvements may also be made to the brightness constancy assumption. Few motion estimation algo-

rithms explicitly account for points entering or exiting occlusion. In most cases this is understandable, as in two camera systems, the motion of occluded points cannot be resolved. However in overconstrained scenarios, such as those possible with the scene particles algorithm, points may be occluded in some views, yet still visible to enough sensors to allow reconstruction of the motion. In this section, inspiration is taken from Fleuret *et al.* [50], to quantify visibility probabilistically, which can then be integrated into the likelihood estimation, to create an occlusion aware version of the scene particles algorithm. In order to modify the likelihood calculation, any occlusion maps must be estimated from the prior distribution.

The probability $\boldsymbol{p}_v(c, \boldsymbol{r}, t)$ of a point $\boldsymbol{r}_t$ being visible in sensor c at time t is related to the probability of another point occluding it. Any occluding points must lie along the optical ray, with unit vector $\hat{\boldsymbol{\theta}}_C$, which originates from camera centre $\boldsymbol{u}_C$ and intersects $\boldsymbol{r}_t$. Occluding points must also have a distance to the sensor lower than that of $\boldsymbol{r}_t$. Thus, the probability an occluding point exists, can be defined as the cumulative probability along the ray, up to $\boldsymbol{r}_t$, which is related to the probability of visibility by some function $f(x)$, as in equation 5.7.

The relationship between $\boldsymbol{r}_t$, $\boldsymbol{u}_C$ and $\hat{\boldsymbol{\theta}}_C$ is shown in figure 5.4. Point 0 is the origin of the world co-ordinates and the other 2 points relate to the structure point under consideration, and the origin of camera c for which the occlusion probability is being estimated. The ray from sensor c through point $\boldsymbol{r}$ is given by $\boldsymbol{r}_t - \boldsymbol{u}_C$. Given this direction, the unit vector $\hat{\boldsymbol{\theta}}_C$ may then be obtained, by dividing by the magnitude. The integration of equation 5.7 is then performed along the ray (including the offset for the sensor origin). Note that unlike many previous approaches to motion estimation [136, 122, 18], the evaluation of a points validity and its occlusion status are evaluated separately. It is common for occlusions to be determined simply by the lack of observational consistency (which in the scene particles algorithm is used for the likelihood function). In contrast, the proposed formulation requires that occlusions be justified by the previously estimated structure and motion fields. Points are not labelled as occluded if no other point is present to occlude it.

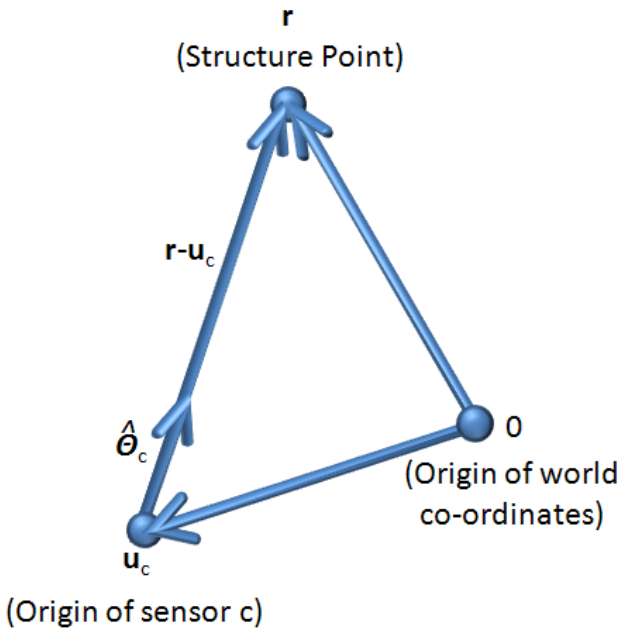


Figure 5.4: Occlusion rays example. The relationship between 3 points in the 3 dimensional world. Point "0" is the origin of the world co-ordinates (often the origin of the reference camera), $\boldsymbol{r}$ is the structure point in the scene under consideration, and $\boldsymbol{u}_C$ is the origin of sensor c. $\hat{\boldsymbol{\theta}}_C$ is the unit vector along the ray from sensor c through point $\boldsymbol{r}$.

$$\boldsymbol{p}_v(c, \boldsymbol{r}, t) = f\left(\int_0^{|\boldsymbol{r}_t - \boldsymbol{u}_C|} \boldsymbol{p}^t(u\hat{\boldsymbol{\theta}}_C + \boldsymbol{u}_C)\,\mathrm{d}u\right) \tag{5.7}$$

Note that the 3 velocity dimensions are marginalised out of the prior to produce $\boldsymbol{p}^t(\boldsymbol{r}_t)$ for this calculation, as the visibility at a point in time is not dependent on the instantaneous velocity at that time. The prior at $t-1$ is re-estimated using $\boldsymbol{r} - \boldsymbol{v}$ before this marginalisation. However, the function $f(x)$ still must be defined. The output probability must be normalised in the range 0 to 1, however the input particle weights do not sum to one. The most obvious choice for $f(x)$, is to normalise by the total weight of all particles along the ray, as in section 4.3.1, as shown in equation 5.8.

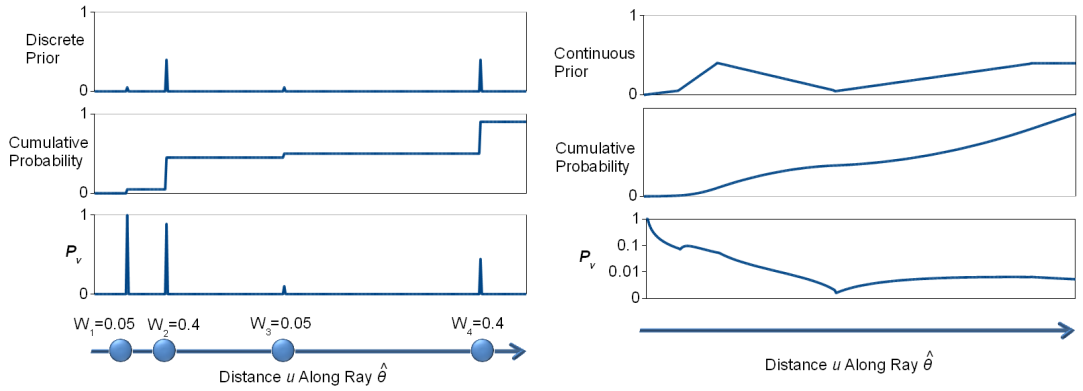
$$\boldsymbol{p}_v(c, \boldsymbol{r}, t) = 1 - \int_0^{|\boldsymbol{r}_t - \boldsymbol{u}_C|} \boldsymbol{p}^t(u\hat{\boldsymbol{\theta}}_C + \boldsymbol{u}_C)\,\mathrm{d}u \tag{5.8}$$

However, this formulation leads to undesirable behaviour. As an example, with such a formulation, the first nonzero point along the ray may have a very low probability of being visibile. This makes little sense as by definition, nothing can occlude the first point on the ray. To extend this example further, the visibility of any particle becomes dependent not only on the weight of particles in front of it, but also behind it. Instead, the occlusion probability is normalised by the probability at $\boldsymbol{r}_t$ as follows:

$$\boldsymbol{p}_v(c, \boldsymbol{r}, t) = \frac{\boldsymbol{p}^t(\boldsymbol{r}_t)}{\int_0^{|\boldsymbol{r}_t - \boldsymbol{u}_C|} \boldsymbol{p}^t(u\hat{\boldsymbol{\theta}}_C + \boldsymbol{u}_C)\,\mathrm{d}u} \tag{5.9}$$

The integral on the denominator relates to the probability at all points up to *and including* $\boldsymbol{r}_t$. Hence, the visibility probability correctly lies within the range $0 \leq \boldsymbol{p}_v(c, \boldsymbol{r}, t) \leq 1$, with the first nonzero point on the ray always assigned $\boldsymbol{p}_v(c, \boldsymbol{r}, t) = 1$. With this formulation, the probability of point $\boldsymbol{r}_t$ being visible, does not depend on the probability of any points lying *behind* $\boldsymbol{r}_t$, which is sensible. The occlusion probability is similarly defined as $1 - \boldsymbol{p}_v(c, \boldsymbol{r}, t)$.

Figure 5.5 illustrates the behaviour of the $\boldsymbol{p}_v(c, \boldsymbol{r}, t)$ distribution, along a single ray, for the discrete case involving 4 particles (figure 5.5a and the continuous case obtained by assuming a constant gradient between each sample (figure 5.5b). Note that the visibility probability for the first particle is one despite its low weight, as nothing is present to occlude it. Further, particle 2 has a very high probability of visibility despite another the presence of particle 1 in front of it, due to its significantly higher weight. Intuitively, the occlusion probability is related to the probability that point $\boldsymbol{r}$ exists, and to the probability that one or more points exist in front of $\boldsymbol{r}$. The complete visibility maps are 3D probability distributions in $\dot{x}, \dot{y}, \dot{z}$. Obviously this is much richer than the 2 dimensional binary masks, that are often employed for handling occlusion. Each viewpoint has a separate visibility map, which may be used in other vision tasks, such as tracking and object recognition.



(a) Discrete case

(b) Continuous case

Figure 5.5: Occlusion Probability example. The cases of both continuous and discrete priors are shown. The discrete example is composed of 4 scene particles of varying weights, while the continuous case assumes a constant gradient between each sample. Note that the probability of occlusion for particle 2 is low, as its weight is much greater than that of particles in front of it.

These visibility maps must still be integrated into the scene flow estimation system. A modification to the original likelihood formulation of equation 4.7 may be performed:

$$\boldsymbol{p}_a(\boldsymbol{I}_a \mid \boldsymbol{r}, \boldsymbol{v}) = \frac{1}{1 + e_a \sum_{m=1}^{M} \sum_{\tau=t\text{-}1}^{t} \frac{\boldsymbol{p}_v(m, \boldsymbol{r}, \tau)}{2M\bar{\boldsymbol{p}}_v(\boldsymbol{r})} \left(\boldsymbol{I}^{\tau}_{a,m}(\boldsymbol{\Pi}_m(\boldsymbol{r}_\tau)) - \bar{I}\right)^2} \tag{5.10}$$

Under this new scheme, the contribution of each sensor to the variance calculation and to the mean colour,

$$\bar{I} = \sum_{m=1}^{M} \sum_{\tau=t\text{-}1}^{t} \frac{\boldsymbol{p}_v(m, \boldsymbol{r}, \tau)}{2M\bar{\boldsymbol{p}}_v(\boldsymbol{r})} \boldsymbol{I}^{\tau}_{a,m}(\boldsymbol{\Pi}_m(\boldsymbol{r}_\tau)) \tag{5.11}$$

is weighted based on its visibility probability. As such, the consistency between viewpoints which are unlikely to be occluded, is assigned greater importance. A similar modification is also applied to the depth likelihood from equation 4.8. It is important to note that the visibility probabilities are normalised, by the total visibility at that point (which is $2(M + L)$ times the average visibility $\bar{\boldsymbol{p}}_v(\boldsymbol{r})$):

$$\bar{\boldsymbol{p}}_v(\boldsymbol{r}) = \sum_{m=1}^{M} \sum_{\tau=t\text{-}1}^{t} \boldsymbol{p}_v(m, \boldsymbol{r}, \tau) \tag{5.12}$$

Thus, the visibility probabilities do not directly increase or decrease the likelihood of a point, regardless of how many viewpoints it is visible in.

Using the new formulation, allows points to receive high likelihoods, even when their appearance is only consistent within a subset of sensors, provided that subset is justified by the visibility maps. Iteratively analysing the likelihood $\boldsymbol{p}(\boldsymbol{I} \mid \boldsymbol{r}, \boldsymbol{v})$ and recomputing the visibility maps, as described in section 4.3.2, leads them both to converge on mutually consistent distributions.

However, the new formulation is susceptible to the degenerate case, where a point is occluded in all but a single sensor. In this case, the consistency (of one sensor with itself) will be high. This is similar to the degeneracy of the original likelihood formulation, discussed in section 4.2.2. The issue can also be solved in a similar way, by adapting the constraints shown in figure 4.1, so that the number of sensors with a greater than average visibility must be at least 3, including at least 1 at both the current and previous frames. These constraints are formalised in the following equations.

$$|\{c, \tau \,|\boldsymbol{p}_v(c, \boldsymbol{r}, \tau) \geq \bar{\boldsymbol{p}}_v(\boldsymbol{r})\}| \geq 3 \tag{5.13}$$

$$|\{c \,|\boldsymbol{p}_v(c, \boldsymbol{r}, 0) \geq \bar{\boldsymbol{p}}_v(\boldsymbol{r})\}| \geq 1 \tag{5.14}$$

$$|\{c \,|\boldsymbol{p}_v(c, \boldsymbol{r}, 1) \geq \bar{\boldsymbol{p}}_v(\boldsymbol{r})\}| \geq 1 \tag{5.15}$$

5.5 Assumption Variant Analysis

In table 5.2 the performance of the standard scene particles algorithm (SP) from the previous chapter, is compared to the forwards-backwards formulation (FB), and the

Algorithm	Dataset	err^n_{of}	err^n_{sf}	err^n_{st}	err^n_{ae}	Runtime
SP	Cones	0.06	**0.00**	0.29	4.08	**213 secs**
SP (FB)	Cones	0.06	0.01	0.28	4.07	217 secs
SP (OCC)	Cones	**0.05**	**0.00**	**0.26**	4.24	268 secs
Basha *et al.* [19]	Cones	1.32	0.01	6.22	**0.12**	-
Huguet*et al.* [64]	Cones	5.79	8.24	5.55	0.69	5 Hours
SP	Teddy	**0.04**	**0.00**	0.20	4.23	**207 secs**
SP (FB)	Teddy	**0.04**	**0.00**	0.20	4.21	213 secs
SP (OCC)	Teddy	**0.04**	**0.00**	**0.18**	4.16	265 secs
Basha *et al.* [19]	Teddy	2.53	0.02	6.13	**0.22**	-
Huguet *et al.* [64]	Teddy	6.21	11.58	5.64	0.51	5 Hours
SP	Venus	0.04	**0.00**	0.25	4.26	**213 secs**
SP (FB)	Venus	0.04	**0.00**	0.24	4.24	214 secs
SP (OCC)	Venus	**0.03**	**0.00**	**0.23**	4.21	277 secs
Basha *et al.* [19]	Venus	1.55	**0.00**	5.39	1.09	-
Huguet *et al.* [64]	Venus	3.70	3.05	5.79	**0.98**	5 Hours

Table 5.2: scene particle performance with constant velocity and occlusions. Results of scene flow estimation for a multi-view, appearance only setup. scene particles embodying both the constant velocity and probabilistic occlusions, are compared to Basha *et al.* [19] and Huguet *et al.* [64].

occlusion aware system (OCC). The previous state of the art techniques from table 4.1 are also included for comparison.

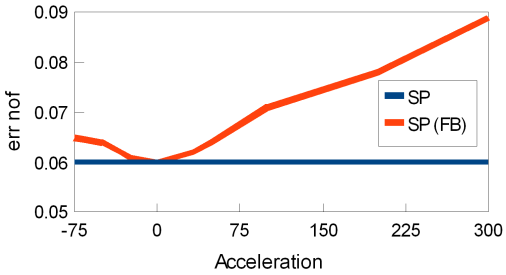
The FB variant appears to provide slightly improved performance, particularly in terms of structural accuracy err^n_{st}, with a negligible increase in computational cost. In contrast the occlusion aware approach (OCC) provides larger improvements, accross all error measures, but at the cost of around 30% increased runtime.

5.6 Non-Constant Velocity

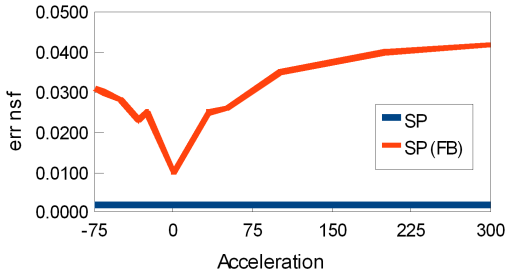
The forwards-backwards formulation appears to offer improved performance for negligible cost. However the analysis was perhaps unfair, as the dataset satisfied the newly incorporated assumption, of constant velocity. For real applications such as the action recognition task that is the focus of this thesis, the assumption will not be perfectly obeyed. Thus, it is important to examine the sensitivity of the system to divergences from the assumption.

It is trivial to use the middlebury datasets, to create test sequences undergoing various levels of acceleration. For example, a 3 frame sequence comprising $\{I_1, I_2, I_4\}$ relates to a 100% increase in velocity, between the second and third frames (scene motion increases from 1 to 2 times the sensor separation). Using this approach sequences can be generated for 10 different levels of acceleration, ranging from -75% to 300%. Results are shown in figure 5.6 with the performance of the standard scene particle algorithm also displayed for comparison purposes.

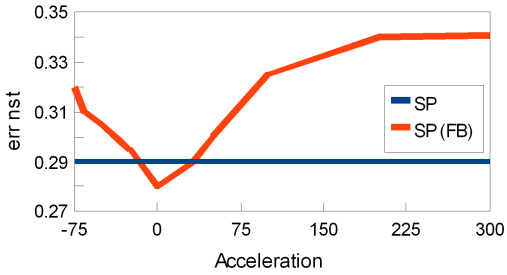
All error metrics display similar trends. Even small deviations from zero acceleration, lead to drastic reductions in accuracy. Thus, although the forwards-backwards system was able to provide significant gains "for free" in the original evaluation, it is far too sensitive to the constant velocity assumption for use in the action recognition task.



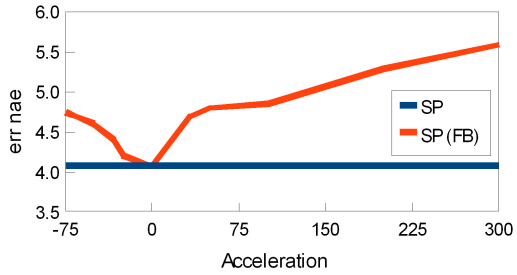
(a) Optical flow error against scene acceleration



(b) Stereo flow error against scene acceleration



(c) Structure error against scene acceleration



(d) Directional error against scene acceleration

Figure 5.6: Constant velocity violation performance. Analysis of scene flow estimation accuracy, using the forwards-backwards scene particles algorithm, under various violations of the constant velocity assumption.

5.7 Improving Brightness Constancy

The sensitivity of the forwards-backwards formulation, to deviations from the constant velocity assumption, raises similar questions about the brightness constancy assumption, which is used in all motion estimation techniques (both optical flow and scene flow). In fact, it is well known that there are many situations in real data, where the brightness constancy assumption is not obeyed. Examples of this include specular effects, shadows, occlusions and directional lighting. In the case of scene flow estimation, these issues are exacerbated by the use of multiple sensors, with different response characteristics.

In previous work, sensitivity to these effects has rarely been explored. Indeed, many motion estimation schemes are analysed on synthetic datasets, as it is quite challenging to get ground truth motion fields for real data (particularly for scene flow fields). As synthetic datasets rarely contain any of these artefacts, this is equivalent to the tests in section 5.5, where the forwards-backwards system seemed valuable, because it was analysed under artificial conditions where its assumption was satisfied. Indeed Vaudrey *et al.* have noted that existing motion estimation techniques often exhibit different behaviours when moved out of the lab and onto real data [123].

The conclusions drawn in the following sections, are equally applicable for multiview scene flow and hybrid scene flow (using appearance and depth). In fact, as the brightness constancy assumption only involves the appearance sensors, the hybrid scene flow scenario is equivalent to single view optical flow in the following analysis. As such, it is useful to briefly outline a unified formulation, before exploring the behaviour of the brightness constancy assumption.

For a given motion vector (either v_x, v_y in the image plane, or $v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}$ in the scene) we can define the set Γ of supporting pixel locations. This set contains the image position x, y and frame number τ for all pixels involved in the motion. For the optical flow ($M = 1$) case, Γ contains one entry from the current frame $(t + 1)$, and one from the previous frame(t):

$$\Gamma = \{(x, y, t), (x + v_x, y + v_y, t + 1)\} \tag{5.16}$$

For scene flow estimation ($M > 1$), a motion has $2M$ associated pixel locations. The set of pixel locations taken from sensor m, obtained by projecting the start and end points of the motion vector, are given by:

$$\Gamma_m = \{(\mathbf{\Pi}_m(\boldsymbol{r}, t)), (\mathbf{\Pi}_m(\boldsymbol{r} + \boldsymbol{v}), t + 1)\} \tag{5.17}$$

Given the sets of supporting pixel locations from each sensor ($\Gamma_{1..m}$), the set of associated pixel *values* is defined as Ψ.

$$\Psi = \{\boldsymbol{I}_m(x, y, \tau) \mid x, y, \tau \in \Gamma_m \text{ for all } m\} \tag{5.18}$$

5.7.1 Matching Functions

Given this unified formulation, the behaviour of the brightness constancy assumption may be analysed. A relatively small number of matching functions have been proposed to embody the brightness constancy assumption, particularly in comparison to the vast number of smoothness formulations.

The simplest of these functions, is based on the absolute deviation (i.e. the l_1 norm) from brightness constancy [100]. This function is defined as ABS in equation 5.20, where $\bar{I}$ is the mean value of the supporting pixels.

$$\bar{I} = \frac{1}{|\Psi|} \sum_{\psi \in \Psi} \psi \tag{5.19}$$

$$ABS(\Psi) = \sum_{\psi \in \Psi} |\psi - \bar{I}| \tag{5.20}$$

A related, approach used by many authors [28, 19, 64, 67, 95, 120] is to employ the l_2 norm, define as SQ in equation 5.21. This relates to the sum of squared deviations,

across all supporting pixels. This is the function used in the scene particle algorithm, and previously presented (in less general form) in equation 4.7.

$$SQ\left(\Psi\right) = \sum_{\psi \in \Psi} |\psi - \bar{I}|^2 \tag{5.21}$$

These functions may be applied to RGB data, as well as greyscale, by creating a separate set of supporting pixel values Ψ_c for each input channel. This leads to a function based on the colour constancy assumption. In a similar vein, gradient images may be used as an input, leading to a set of supporting gradient values Ψ_g, and functions ABS_g and SQ_g, based on the Gradient Constancy Assumption [28, 95, 64]. functions based on the "response consistency" of any linear filter, have also been proposed [118], which includes the gradient constancy functions.

$$ABS_g\left(\Psi_g\right) = \sum_{\psi \in \Psi_g} |\psi - \bar{I}_g| \tag{5.22}$$

$$SQ_g\left(\Psi_g\right) = \sum_{\psi \in \Psi_g} |\psi - \bar{I}_g|^2 \tag{5.23}$$

The final function to be examined, is the so called "optical flow constraint" [44, 75, 82, 107, 24]. This is a more complex formulation, which ignores the supporting pixel values from time $t+1$, and instead uses the temporal gradient image $\boldsymbol{I}_t$. The function equates to a linearised Taylor expansion of the brightness constancy assumption (i.e. with all terms of quadratic or higher power dropped). The function is defined as OFC in equation 5.24, where $\boldsymbol{I}_{\nabla x}$ and $\boldsymbol{I}_{\nabla y}$ are the spatial gradient images.

$$OFC(\Gamma) = \sum_{m=1}^{M} \sum_{(x,y,\tau) \in \Gamma_m} \begin{cases} 0 & \text{if } \tau = t+1 \\ \boldsymbol{I}_t(x,y,t+1) + v_x \boldsymbol{I}_{\nabla x}(x,y,t+1) + v_y \boldsymbol{I}_{\nabla y}(x,y,t+1) & \text{otherwise} \end{cases} \tag{5.24}$$

Figure 5.7: An example frame from the Cones sequence

5.7.2 Function Behaviour Analysis

To analyse these functions, embodying various appearance constancy assumptions, the responses are examined for the ground truth motion fields (both optical flow and scene flow) of real scenes. Results for the Cones sequence [106] are shown in figure 5.8. Figure 5.7 contains an example frame from this sequence. The analyses in the remainder of this chapter, are also performed on 4 additional sequences from the middlebury datasets, including a range of illumination conditions. These additional results are presented in appendix B. Blue regions have no ground truth. Regions of brighter red, indicate areas in which the appearance constancy assumption holds, while dark red areas, are where the scene displays inconsistencies.

Unsurprisingly, the use of l_1 or l_2 norms have little effect on the general behaviour of the functions. The brightness constancy functions ABS and SQ hold well within objects. However, they prove very sensitive to occlusions, which can be seen by the "shadowing" of the cones. Even within objects, there are some regions where the brightness constancy assumption is less valid due to specular effects, such as the nose of the mask. The scene flow case also fails on fine textures, such as the writing on the matchbox and the pattern on the cones. This is because details close to the pixel level

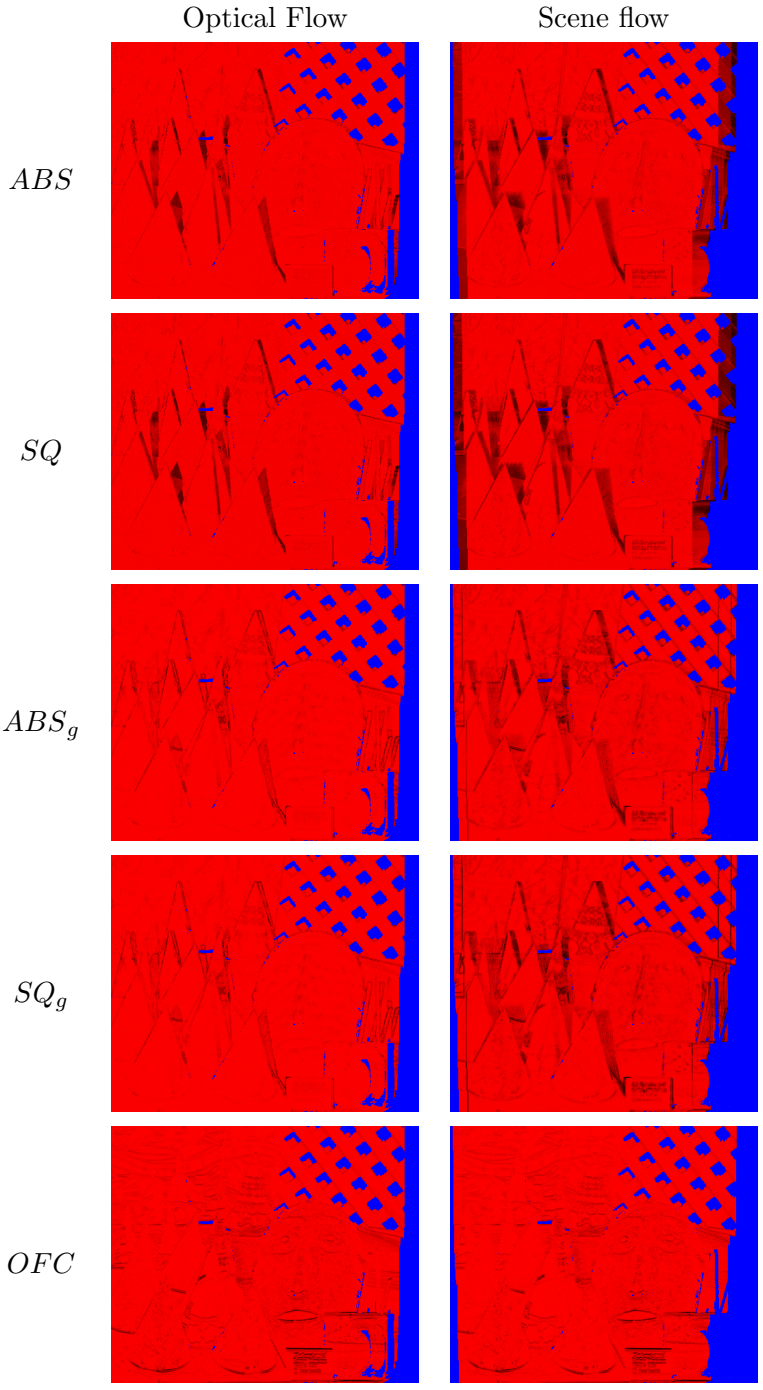


Figure 5.8: Function response images. Response of various motion estimation functions, for the ground truth motion of a real scene. Dark red regions indicate deviations from the constancy assumption. Blue regions lack ground truth.

are quantised differently in each sensor, due to varying subpixel overlap.

The gradient constancy measures ABS_g and SQ_g, prove to be the most robust functions for optical flow, performing reasonably well on occlusions and fine textures. For scene flow estimation, the gradient constancy functions are comparable to brightness constancy. The linearised brightness constancy function OFC appears to be far more robust to occlusions than the other functions. However, it also displays the greatest inconsistency within objects, particularly at edges and fine details. Interestingly, shadowed areas such as the eye and mouth holes of the mask, provide the regions of maximum response for all consistency measures, due to the underexposure and consequent lack of contrast.

Separating Truth From Errors

Although each function breaks down under different circumstances, the inconsistencies are generally in the minority. However, providing a high response for true motions is only half the task for a good function. The second requirement, is to provide a low response for incorrect motions. Indeed, an ideal function would produce a continuously decaying output, as the error on the input was increased. In figure 5.9 the Probability Density Function (PDF) is shown (i.e. what fraction of the scene is assigned each response value) for each function on the cones sequence. Each graph displays two PDFs, one for the ground truth motion field (i.e. the pdf of the red values in figure 5.8), and one for an "erroneous" motion field (where the motion magnitude at every location is doubled). Note that the response of each function are normalised between 0 and 1 to facilitate comparisons. A response of zero implies no inconsistency with the associated constancy assumption, and a response of 1 is a strong inconsistency.

The results are disappointing. As noted previously, most ground truth motions lie within the lower 20% of the responses, while occlusions, specularities and other artefacts are in the minority. However, even the significantly erroneous motions considered here, lead to a similar distribution of responses. The linearised brightness constancy function OFC shows an 80% overlap between the two PDFs. The gradient based functions ABS_g and SQ_g produce heavier tailed distributions for erroneous motions, (i.e. the

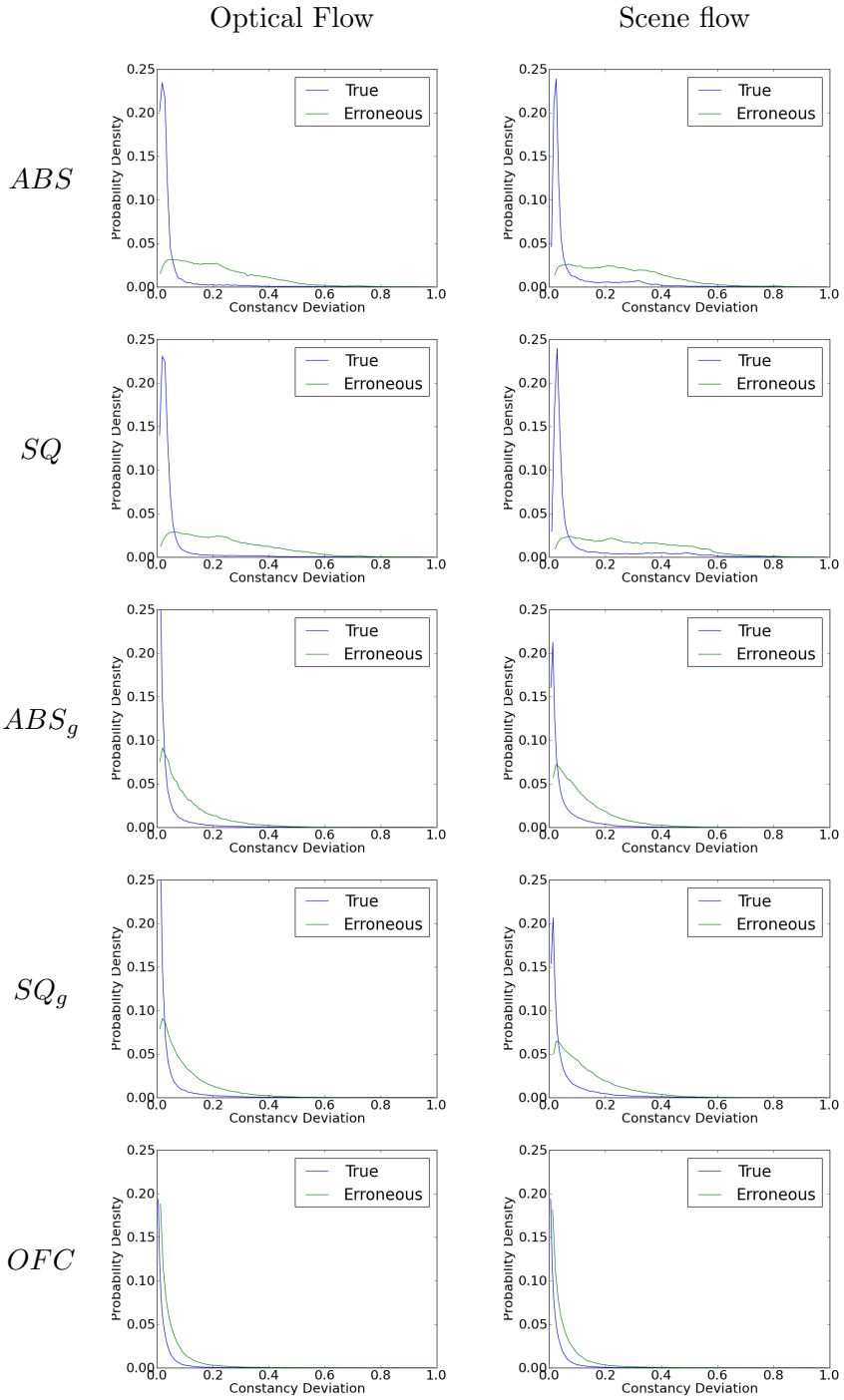


Figure 5.9: Response distributions. Distribution of responses for various motion estimation functions, applied to both ground truth and error motions of a real scene. Responses are normalized in the range 0 to 1.

shape of the two PDFs, and the locations of the peak are the same, but there is a slight increase in the number of outliers). The best separation occurs for the simple brightness constancy functions ABS and SQ, however performance is still quite poor, given the large motion error under consideration.

The large overlap of the response distributions in each case, implies that most erroneous motions, cannot be distinguished from cases where the scene does not obey the constancy assumption (due to specularities, directional lighting etc). Thus, attempting to minimize the function response across the scene will result in almost as many correct motions being discarded, as incorrect.

Convergence Behaviour

In general, motion estimation schemes employ optimization, to follow the gradient of the function response, assuming that reducing this inconsistency leads to reduced motion errors. Although the scene particles algorithm is stochastic, and doesn't explicitly compute the response gradient, the iterative resampling and diffusion does lead to a shift of sampling density (equivalent prior probability) towards response minima. In figure 5.10, graphs are shown for each function, where the y axis displays the average response over the whole motion field, as the amount of motion error is varied on the x axis, between 0 (no motion) and 2 (twice the true motion).

The brightness constancy based schemes (ABS and SQ) do display a general trend of reduced inconsistency towards the true motion, despite the significant overlap of response distributions. In contrast, functions based on Gradient constancy (ABS_g and SQ_g) perform poorly, with roughly the same response produced for all motions more than 10% away from the ground truth. This implies that when such a function is employed for motion estimation, convergence will occur extremely slowly, if at all, unless the initialization is very close to the true value. The linearised brightness constancy constraint OFC provides the worst performance. This function always favours smaller motions, and on average displays little tendency towards the correct motion. This explains why multi-scale approaches are so commonly employed, attempting to enable the function to allow larger flows.

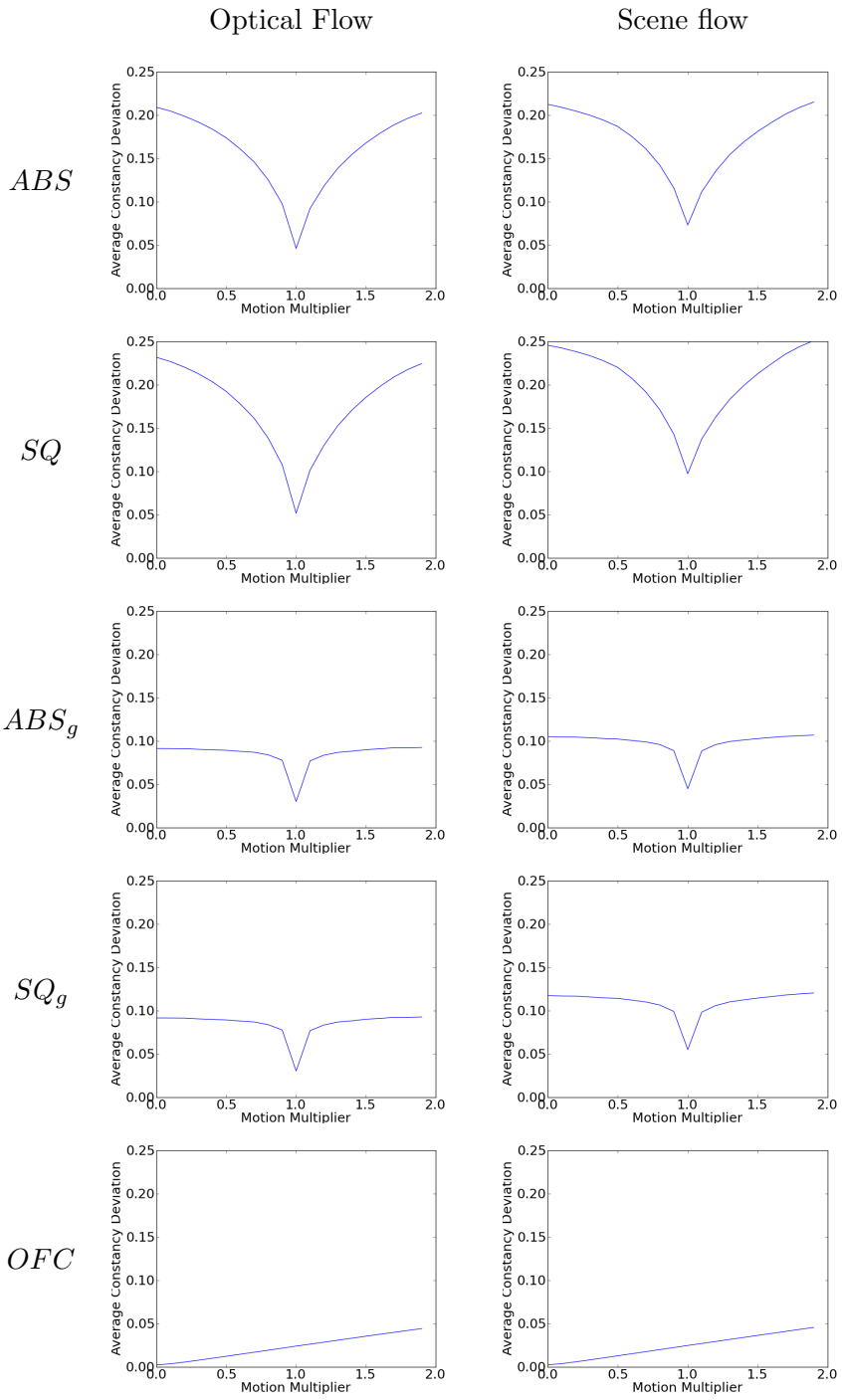


Figure 5.10: Convergence performance. The average constancy deviation across the scene, for each function, analysed with varying degrees of motion error.

5.7.3 Intelligent Transfer Functions

The previously proposed functions for motion estimation, assume either a linear or quadratic relationship between the deviation from appearance constancy, and the "quality" of the match. However, it has been shown that many deviations occur due to the properties of the scene, and do not necessarily reflect errors in the input motion. The functions do not take into account any of these "acceptable" inconsistencies, and thus perform poorly when differentiating motion errors from scene artefacts. To address this, it may be possible to employ machine learning techniques, to find an "intelligent" matching criteria, of unconstrained form, which is robust to the appearance inconsistencies of real data, while still being sensitive to inconsistencies from motion errors.

By learning such a nonlinear relationship, it is possible to embody more complex behaviours. As an example, it may be expected that in very light or dark parts of the scene, image contrast would be reduced. In this case, little variation may be expected naturally, and any appearance deviations may be more significant. Learning such behaviour, would serve to flatten the exaggerated response peaks, that were observed in underexposed regions of figure 5.8. As a second example, specular effects often cause a large change in appearance across all colour channels, while changes in the appearance of only one channel may be more likely to relate to an erroneous motion.

In order to produce this nonlinear function, an Randomised Decision Forest (RDF) classifier [27] is employed. The RDF is capable not only of classifying an input, but also of estimating the likelihood for each class. This is advantageous in a probabilistic framework such as the scene particles algorithm, as it removes the need for the normalisation step used to produce the PDFs in section 5.7.2. Such normalisation steps fit poorly into online estimation schemes such as motion estimation.

For a classifier which is trained to distinguish true motions from erroneous motions, the estimated likelihood of the "erroneous motion" class, may be used as a function, which is equivalent to the previous constancy deviation functions (i.e. a response of 1 indicates the motion was definitely erroneous, and a response of 0 indicates the motion was definitely true). For the following evaluations (and those presented in appendix B) the ITFs were trained using 500,000 samples, extracted from an unseen

sequence (the Teddy series [106]). In the following chapter (section 6.3) generalisation of these functions on data outside the Middlebury sequences is demonstrated.

Figure 5.11 shows the performance of various ITF, based on a range of input feature sets. The first is the variance feature set F_{var}, which is defined in equation 5.25. This set contains only a single element, which is equivalent to the output of the squared differences function SQ previously defined in equation 5.21. By learning a nonlinear mapping function, improved separation over direct use of SQ is possible in the optical flow scenario. However, in the more difficult scene flow estimation task, little improvement is obtained over the original function.

$$F_{var}(\Psi) = \left\{ \frac{1}{|\Psi|} \sum_{\psi \in \Psi} |\psi - \bar{I}|^2 \right\} \tag{5.25}$$

The second feature set are the differences features F_{dif}, which are calculated as the distance of each supporting pixel value, from the mean supporting value, as in equation 5.26. This is equivalent to the input of the absolute difference function ABS previously defined in equation 5.20. By learning from the naive functions inputs, rather than attempting to remap its outputs, the ITF is able to provide greatly improved separation, due to considering nonlinear combinations of inputs. The improvement is especially noticeable in the scene flow case, where the number of features is larger.

$$F_{dif}(\Psi) = \left\{ |\psi - \bar{I}| : \psi \in \Psi \right\} \tag{5.26}$$

The third feature set F_{pix} attempts to learn directly from the raw pixel values. Interestingly, for the optical flow case this leads to improved performance, over using the differences feature F_{dif}. However, for scene flow estimation the F_{dif} proves more effective. This makes sense, as the RDF (like most classifiers) makes the assumption that features are independent, but for F_{pix}, most of the information is contained in the correlation between features. To illustrate this point, note that for an ideal true motion, every difference feature F_{dif} should be low, and may be examined in isolation, and then aggregated. However, the pixel features F_{pix} may take any value for an ideal motion, as long as every feature takes the same value. Given this knowledge, it is obvious that

the more features present in F_{pix}, the more difficult it will be for the RDF to learn the correlation relationship. This explains why performance is so much lower in the scene flow case (which consists of 8 features) than in the optical flow case (2 features). Despite this difficulty, the raw pixel features do contain additional information, and allow regions to be treated differently depending on their intensity (for example, one rule may be used in dark areas, and another in light areas) which is not possible using the normalized F_{dif} features. This explains the improvement gained in the optical flow case.

$$F_{pix}\left(\Psi\right) = \Psi \tag{5.27}$$

All the ITFs display a reduction in the effect of outliers. In the standard functions, the vast majority of the motion fields (both true and erroneous) fell within the bottom 20% of the response range, while the remaining 80% of the response values, were populated by a very small number of outliers.

In addition to enabling complex nonlinear combinations of the information from supporting pixels, it is possible to include higher level information in an ITF. As an example, features based on local context may be utilised, such as the local image variance F_{pv} around each supporting pixel, which may indicate whether the pixel lies along a boundary or within a surface. These features are defined in equation 5.29, where $\bar{\boldsymbol{I}}_m\left(x, y, \tau\right)$ is the mean value of a patch centred at x, y, τ and Ω is the set of 2D offsets, relating to a local neighbourhood (a 3×3 patch was used in these experiments).

$$\bar{\boldsymbol{I}}_m\left(x, y, \tau\right) = \frac{1}{|\Omega|} \sum_{x', y' \in \Omega} \boldsymbol{I}_m\left(x + x', y + y', \tau\right) \tag{5.28}$$

$$F_{pv}\left(\Gamma\right) = \left\{ \frac{1}{|\Omega|} \sum_{x', y' \in \Omega} \left(\boldsymbol{I}_m\left(x + x', y + y', \tau\right) \text{-} \bar{\boldsymbol{I}}_m(x, y, \tau)\right)^2 \mid x, y, \tau \in \Gamma_m, \forall m \right\} \tag{5.29}$$

The local means around each supporting pixel may also be used to create a feature set F_{pm}, which describes how light or dark the regions around the supporting pixels are.

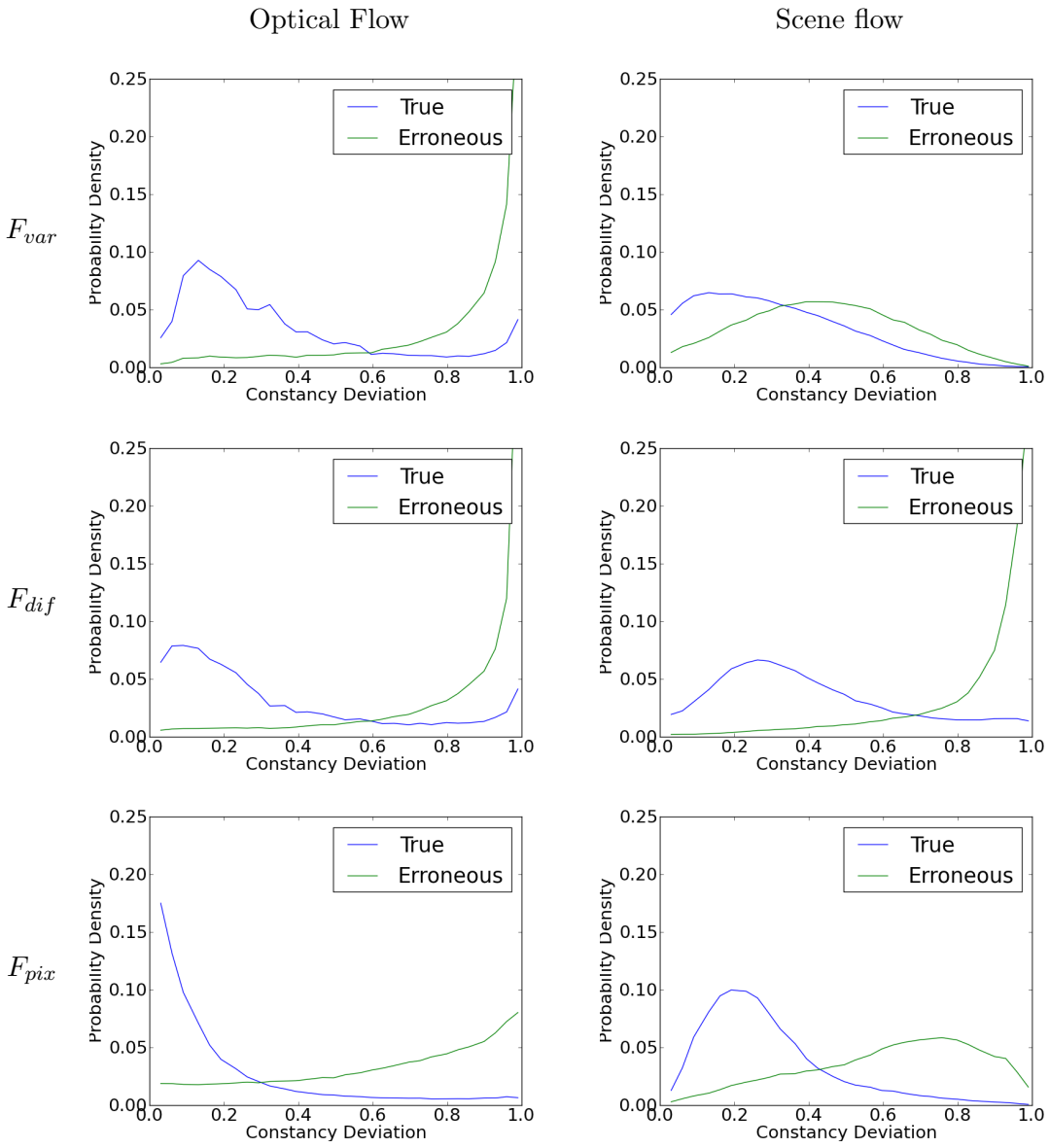


Figure 5.11: ITF response distributions. Distribution of responses for various ITFs, applied to ground truth and erroneous motions in a real scene.

This is a more robust version of the information present in F_{pix}, which proved valuable in the optical flow case.

$$F_{pm}(\Gamma) = \left\{ \bar{\boldsymbol{I}}_m(x, y, \tau) \mid x, y, \tau \in \Gamma_m, \forall m \right\} \tag{5.30}$$

A more complex form of contextual feature is defined as F_{ph} in equation 5.31. This relates to a coarse (3 bin for these experiments) histogram of the local values around each supporting pixel. This contains similar information to the contextual mean and variance features F_{pm} and F_{pv} (i.e. light versus dark regions and boundaries versus surfaces) with some additional information.

$$F_{ph}(\Gamma) = \left\{ \underset{x', y' \in \Omega}{\text{hist}} \left(\boldsymbol{I}_m \left(x + x', y + y', \tau \right) \right) \mid x, y, \tau \in \Gamma_m, \forall m \right\} \tag{5.31}$$

It's important to note that these contextual features are more general than the direct "patch matching" which is commonly employed in local motion estimation algorithms. For these contextual features, there does not need to be an explicit one to one correspondence between pixels. This creates robustness to patch changes, due to projective distortions, which are normally handled by a separate warping scheme. ITFs may be employed within local "patch matching" schemes however, by replacing the pixel-wise comparison measure.

As can be seen in figure 5.12 the inclusion of the local variance feature F_{pv} provides little benefit over using F_{dif} alone, in both scenarios. This implies that knowing if the motion is on an object boundary, is not useful when determining its validity. The more complex local histogram features F_{ph} do provide improved performance in the optical flow scenario, but interestingly prove detrimental for scene flow estimation. It is possible that the larger number of features (216 in the scene flow scenario) makes it more difficult to determine accurate class boundaries. The most valuable contextual features actually prove to be the local means F_{pm}, which are also simplest and fastest to compute.

Figure 5.13 shows the average response of the learned function, across the scene, for varying levels of motion error. The curvature of the response is far greater than even

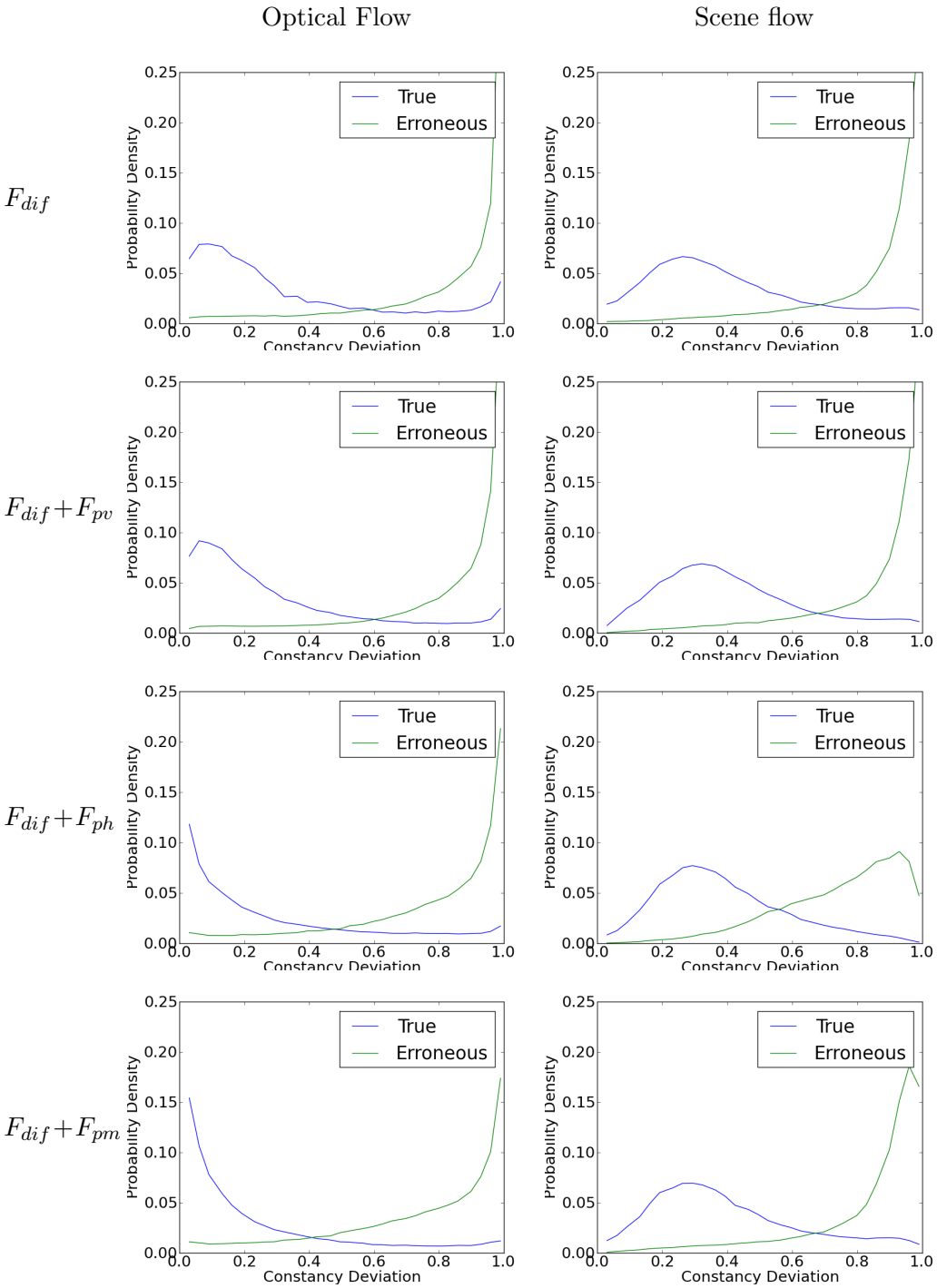


Figure 5.12: Contextual ITF response distributions. The distribution of responses for ITFs based on F_{dif}, including various types of contextual information.

the original brightness constancy functions (note that the scale of the y axis had to be modified from that used in figure 5.10, in order to make the new curves visible). This suggests that ITFs provide much more rapid convergence during optimization for motion estimation, in addition to the expected gains in accuracy. The inclusion of contextual information also yields improved convergence rates, in addition to the improvements in separation.

5.7.4 Motion Estimation with Intelligent Transfer Functions

Although ITFs display more robustness to the appearance constancy assumption, most motion estimation techniques, including the scene particles algorithm, already contain mechanisms (such as multiscale estimation and iterative warping schemes) designed to mitigate standard functions shortcomings. As such, it is important to quantify whether the gains in robustness translate to improved accuracy in motion estimation.

In the original scene particles formulation, the sampling of the likelihood distribution used to weight each scene particle, was achieved by analysing equation 4.7. Instead, the supporting pixel set for each scene particle may be calculated, and an ITF may be employed. Note however, that the scene particles algorithm is formulated in a probabilistic manner, rather than than the more standard energy minimisation formulation. As such, 1 minus the response of the ITF was used (i.e. high values likely relate to true motions, while low values relate to erroneous motions).

Motion estimation accuracy using a number of ITFs was compared to the original SQ formulation. Testing was performed under both the multi-view (table 5.4) and hybrid (table 5.3) scenarios, the latter of which equates to an optical flow analysis, for the purposes of brightness constancy sensitivity.

In addition to motion estimation accuracy, the computational complexity of each function is shown in table 5.3. These complexities are described in terms of the number of supporting pixels (i.e. the cardinality of Ψ, which is 6 for the hybrid scenario and 24 for the multi view scenario, due to the use of 3 independent colour channels) and the number of pixels used for local context (cardinality of Ω, which is 3×3 in these experiments). The ITFs also have an additional cost, for the RDF to analyse the features.

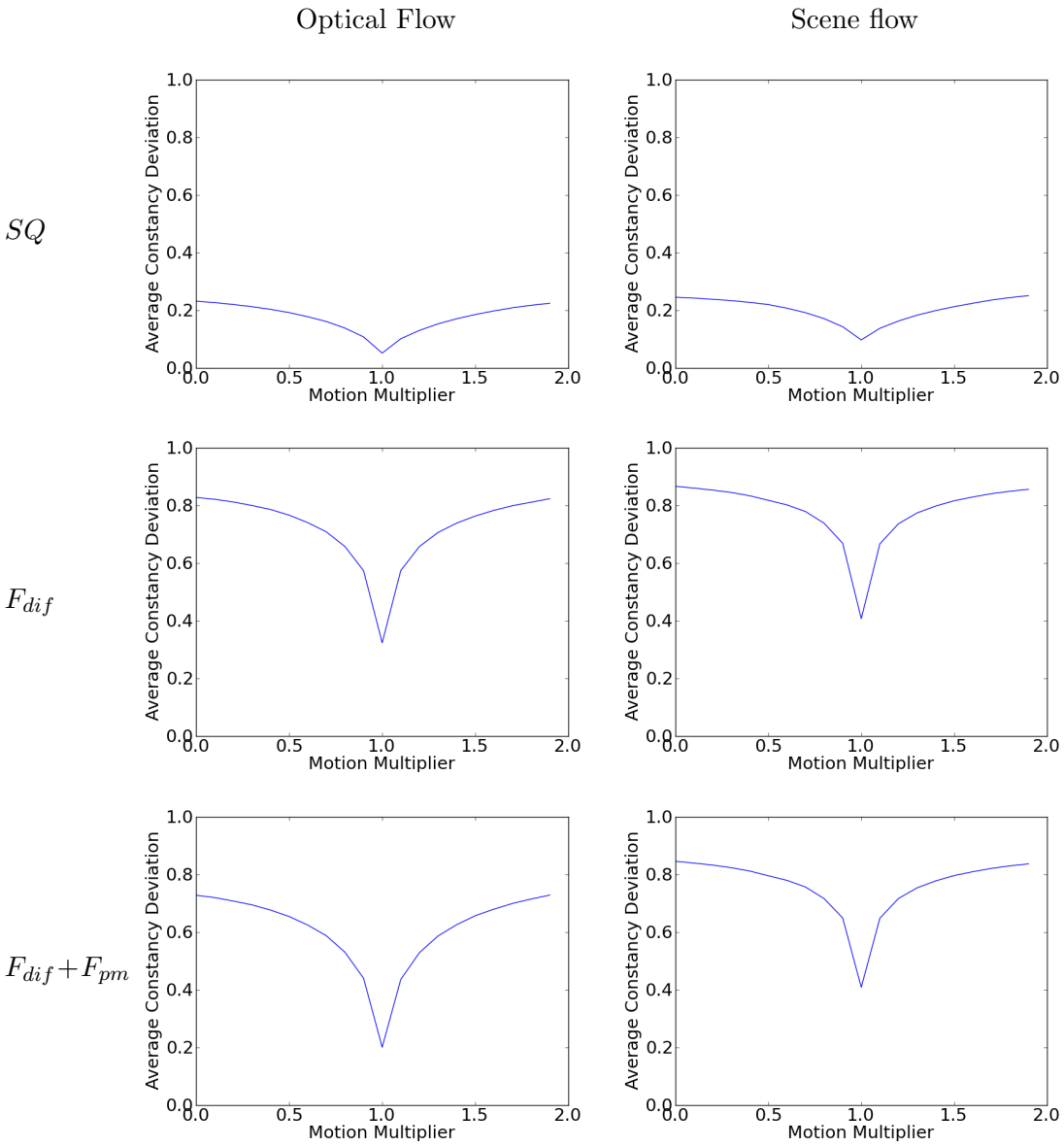


Figure 5.13: ITF convergence performance. The average function response for varying levels of motion error, for the best ITFs with and without contextual information. The previous SQ function is also shown, rescaled to the same y axis, for comparison purposes.

Function	err^n_{of}	err_{ae}	Complexity
SQ	0.095	5.02	$O(3\|\Psi\|)$
F_{var}	0.046	4.06	$O(3\|\Psi\| + \zeta log(\phi))$
F_{dif}	0.048	4.11	$O(2\|\Psi\| + \zeta log(\phi))$
F_{pix}	**0.036**	**3.22**	$O(\zeta log(\phi))$
$F_{dif} + F_{pv}$	0.051	4.39	$O(3\|\Psi\|\|\Omega\| + 2\|\Psi\| + \zeta log(\phi))$
$F_{dif} + F_{ph}$	0.046	3.95	$O(\|\Psi\|\|\Omega\| + 2\|\Psi\| + \zeta log(\phi))$
$F_{dif} + F_{pm}$	0.041	3.50	$O(\|\Psi\|\|\Omega\| + 2\|\Psi\| + \zeta log(\phi))$

Table 5.3: Single view motion estimation performance with ITFs. Performance for hybrid scene flow estimation, based on the standard SQ function, and a range of ITFs. Also shown is the computational complexity for analyzing each function.

For a linear implementation (not exploiting the independence of trees within the RDF), this cost is equal to the number of trees ζ (20 in these experiments), times the log of the tree depth ϕ (which averaged at 20). The complexity in relation to each parameter is linear or better, for all functions (note however that the neighbourhood cardinality $|\Omega|$ would often be incremented in a quadratic manner). This means that the runtime cost of using each function should be roughly the same order of magnitude. In these experiments these complexity relationships lead to the simpler ITFs having roughly the same computation time as SQ for the multiview scenario, with computation for the hybrid scenario roughly doubled. ITFs utilising contextual features have computations times increased by roughly a factor of 2. It is interesting to note, that it is often faster to use the F_{pix} ITF, rather than to simply calculate the variance (i.e. the SQ function) particularly when estimating multiview scene flow. It should also be noted that these changes in computational costs only affect one stage of the scene particle algorithm.

The motion estimation results, reflect the behaviour seen in the initial sensitivity analysis. This demonstrates that despite the multiscale estimation scheme and other mitigating factors, the scene particle algorithm is still sensitive to the brightness constancy assumption, and that by improving robustness to the assumption, improved accuracy may be obtained. For single view motion estimation, even the worst performing of

Function	err^n_{of}	err^n_{sf}	err^n_{st}	err_{ae}
SQ	0.062	0.001	2.90	4.08
F_{var}	0.059	0.002	2.91	4.03
F_{dif}	0.040	0.001	**1.98**	3.47
F_{pix}	0.051	0.001	2.24	3.62
$F_{dif} + F_{pv}$	0.036	**0.000**	2.10	3.68
$F_{dif} + F_{ph}$	0.048	0.001	2.17	3.91
$F_{dif} + F_{pm}$	**0.035**	0.001	2.02	**3.03**

Table 5.4: Multi view motion estimation performance with ITFs. Performance for scene flow estimation, based on the original SQ function, and a range of ITFs.

the ITFs, leads to almost a 50% reduction in motion magnitude errors. The raw pixel value features F_{pix} prove the most effective (65% reduced err^n_{of} and 35% improvement directional accuracy), closely followed by the combination of difference and context features ($F_{dif} + F_{pm}$). In the multi-view tests, F_{var} offers marginal improvement over the original formulation, confirming that a simple nonlinear mapping is insufficient for this more complex task. However, the other ITFs all provide some performance gain, with the best results coming from the difference features coupled with context $F_{dif} + F_{pm}$, which provides a more modest 44% improvement in magnitude accuracy, and 20% improvement in directional accuracy, coupled with a 30% reduction in structural error.

5.8 Conclusions

In this chapter, the fundamental assumptions of the scene particle algorithm have been explored. Initially the inclusion of additional constraints, embodying new assumptions, was explored. Specifically, the constant velocity assumption was formalised, and various approaches were employed to incorporate the assumption of a smooth motion field. It was found that a traditional, local, approach to smoothness, leads to motion estimation similar to previous state of the art performance (i.e. improved directional accuracy, and degraded performance for all other error measures). Suggesting that previous formula-

tions of smoothness constraints may be a significant limiting factor on the performance of contemporary scene flow estimation. The constant velocity assumption was shown to be valuable, and to incur very little computational cost. However, further analysis demonstrated that the sensitivity to the assumption makes it impractical for real vision tasks.

Motivated by this, the sensitivity of the original brightness constancy assumption was also analysed, in the context of a number of previously proposed matching functions. Each function was shown to be broken by different artefacts of real data. More importantly, the standard functions were shown to be violated to the same degree by both erroneous estimates, and scene artefacts. Some of the functions were also seen to exhibit poor convergence behaviour, with the optical flow constraint always favouring smaller motions, while the gradient constancy assumption required accurate initialisation.

An ITF was proposed, using machine learning techniques to create a more general and robust non-linear version of the brightness constancy assumption. Various ITFs were examined, based on a range of inputs including a number of types of contextual information. Such functions were shown to offer a more robust representation of the behaviour of real data, with scene artefacts penalised less heavily than erroneous motions, while also offering improved convergence rates. By incorporating these ITFs into the scene particles algorithm, significant improvements in accuracy were obtained in both the hybrid and multiview scenarios, for negligible computational cost.

In the future it would be useful to confirm that the application of ITFs, leads to improved performance in alternative motion estimation techniques, in addition to the probabilistic framework examined here. Additionally a similar approach may be used for the smoothness assumption, or for the depth constancy assumption used in range flow and hybrid scene flow estimation. It may also prove valuable to learn joint ITFs, such that for example, a single function would be used to embody both the brightness constancy and smoothness functions. Finally it would be useful to investigate the application of ITFs to other areas such as near duplicate image retrieval and image registration tasks.

Chapter 6

3D Motion Features

The scene particles algorithm allows the generation of accurate 3D motion fields, with a speed that is tractable for real vision applications. As such, it is now possible to return to the Hollywood 3D action recognition dataset, introduced in chapter 3. The following chapter explores techniques for encoding 3D motions, as a feature descriptor for recognition, in much the same way that 3D structure was explored in section 3.2.3.

In section 3.2.3 the ω operator was introduced, to create a joint histogram of the values from two images, over a local neighbourhood. A HOG descriptor, was defined as the application of this operator to the gradient images $\boldsymbol{I}_{\triangledown x}, \boldsymbol{I}_{\triangledown y}$. Additionally, HOF descriptors were defined as the result, when the operator was applied to optical flow images ($\boldsymbol{I}_{\frac{dx}{d\tau}}$ and $\boldsymbol{I}_{\frac{dy}{d\tau}}$). Here, a more detailed description of the HOF case is presented, followed by a formulation for scene flow fields.

6.1 Histograms of Oriented Optical Flow

The HOF descriptor (sometimes also referred to as "Histogram of Oriented Optical Flow" or HOOF [33]) encodes local motion information, around a salient point (x, y). The descriptor is calculated over a neighbourhood, defined by the set Ω of 2D pixel offsets. To preserve some degree of spatial information, the neighbourhood is split into a grid of ϵ by ϵ cells. A separate histogram ω_{ij} is computed for each cell, and then

concatenated, and the result normalised as in equations 6.1 and 6.2. By normalising at the neighbourhood level, the descriptor gains scale invariance, as well as invariance to changes in the speed the action is performed, while differences in the relative motion between cells are maintained.

$$\omega_u = \{\omega_{11}, \omega_{12}, ..., \omega_{\epsilon\epsilon}\} \tag{6.1}$$

$$\omega = \frac{\omega_u}{\sum_{b \in \omega_u} b} \tag{6.2}$$

The histogram for each spatial cell, is computed using the flow magnitude and orientation images, defined in equations 6.3 and 6.4 respectively. If B is the number of bins used to quantise the flow orientations, then the value of bin b in cell histogram $\omega_{\epsilon\epsilon}$ is determined as in equation 6.5, where $\Omega_{\epsilon\epsilon}$ is the 2D set of pixel locations for the cell in question. Intuitively, this relates to each pixel casting a vote, weighted by its magnitude, into the bin determined by its orientation.

$$\boldsymbol{I}_{mag}(x,y) = \sqrt{\boldsymbol{I}_{\frac{dx}{d\tau}}(x,y)^2 + \boldsymbol{I}_{\frac{dy}{d\tau}}(x,y)^2} \tag{6.3}$$

$$\boldsymbol{I}_{ori}(x,y) = \arctan\left(\frac{\boldsymbol{I}_{\frac{dx}{d\tau}}(x,y)}{\boldsymbol{I}_{\frac{dy}{d\tau}}(x,y)}\right) \tag{6.4}$$

$$\omega_{\epsilon\epsilon}[b] = \sum_{\boldsymbol{o} \in \Omega_{\epsilon\epsilon}} \begin{cases} \boldsymbol{I}_{mag}((x,y)+\boldsymbol{o}) & \text{if } \frac{2\pi b}{B} \leq \boldsymbol{I}_{ori}((x,y)+\boldsymbol{o}) < \frac{2\pi(b+1)}{B} \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

As an example, if the number of orientation bins is set to four ($B = 4$), then the first bin ($\omega_{\epsilon\epsilon}[0]$) would contain votes from all pixels with an orientation between 0 and $\frac{\pi}{2}$ radians. Likewise the second bin ($\omega_{\epsilon\epsilon}[1]$) would be filled by the magnitude of all pixels, whose orientation lay between $\frac{\pi}{2}$ and π. The result is that every pixel in the cell, is entered into exactly 1 bin of the histogram.

6.2 Histograms of Oriented Scene Flow

A similar approach can easily be employed to encode scene flow, using the output image $\boldsymbol{I}_o$ defined in section 4.3.4. This image contains 4 channels, the first two of which encode the 2D image plane motion v_x, v_y (relating to images $\boldsymbol{I}_{\frac{dx}{d\tau}}$ and $\boldsymbol{I}_{\frac{dy}{d\tau}}$ above), while the third channel encodes the out of plane motion v_d (the fourth channel contains the disparity of the point, and is not used here).

For three dimensional vectors, a single orientation value is not sufficient. Instead a pair of spherical co-ordinates, azimuth and elevation, are defined in equations 6.6 and 6.7. Figure 6.1 illustrates this binning of this 2D spherical orientation space, if the start point for a unit magnitude flow vector is at the origin of the co-ordinate system (i.e. the centre of the circle) then its end point will lie somewhere on the surface of the sphere, which is divided into discrete regions. The Azimuth specifies rotation within the xy plane (around the z axis) taking values from 0 to 360 degrees. In the diagram this relates to motion of the flow end point along the concentric rings. Azimuth is analogous to the in plane orientation used in HOFs. Elevation specifies the rotation within the yz (around the x axis) and takes values from -90 to +90 degrees. The reduced range prevents ambiguity and ensures every point on the sphere is defined by a single unique pair of orientations. As an example, an azimuth of 0 and elevation of 10 would specify (if allowed) the same end point as an azimuth of 180 and elevation of 350. The magnitude of a flow controls its voting weight into the relevant bin, and is defined in equation 6.8.

$$\boldsymbol{I}_{azi\text{-}p}(x,y) = \arctan\left(\frac{\boldsymbol{I}_o[1]\,(x,y)}{\boldsymbol{I}_o[0]\,(x,y)}\right) \tag{6.6}$$

$$\boldsymbol{I}_{ele\text{-}p}(x,y) = \arctan\left(\frac{\boldsymbol{I}_o[2]\,(x,y)}{\boldsymbol{I}_o[1]\,(x,y)}\right) \tag{6.7}$$

$$\boldsymbol{I}_{mag\text{-}p}(x,y) = \sqrt{\boldsymbol{I}_o[0]\,(x,y)^2 + \boldsymbol{I}_o[1]\,(x,y)^2 + \boldsymbol{I}_o[2]\,(x,y)^2} \tag{6.8}$$

Given these new definitions of orientation and magnitude, the joint histogram (where each combination of azimuth bin b_1 and elevation bin b_2 is stored separately) is defined

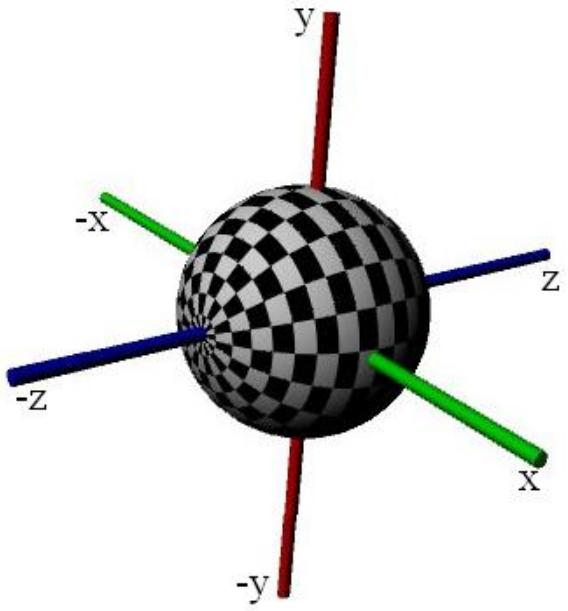


Figure 6.1: Azimuth and Elevation bins on a 2D spherical space. Azimuth is rotation around the z axis, relating to moving the flows end point around the concentric rings. Elevation is rotation around the x axis (i.e. moving between different concentric rings). This example has 20 bins for both the azimuth and elevation dimensions.

for cell (ϵ,ϵ) in equation 6.9. As for the optical flow case, normalising and concatenating each cell histogram produces the full HOS descriptor ω^p.

$$\omega^p_{\epsilon\epsilon}[b_1][b_2] = \sum_{\boldsymbol{o}\in\Omega_{\epsilon\epsilon}} \begin{cases} \boldsymbol{I}_{mag\text{-}p}((x,y)+\boldsymbol{o}) & \text{if } \frac{2\pi b_1}{B_{azi}} \leq \boldsymbol{I}_{azi\text{-}p}((x,y)+\boldsymbol{o}) < \frac{2\pi(b_1+1)}{B_{azi}} \\ & \text{and } \frac{-\pi b_2}{B_{ele}} \leq \boldsymbol{I}_{ele\text{-}p}((x,y)+\boldsymbol{o}) < \frac{\pi(b_2+1)}{B_{ele}} \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

6.2.1 Undistorted HOS

The ω^p descriptor extends the original HOF descriptor to encode additional information about out of plane motions. However, the formulation may comply too closely with the

original. Using the v_x, v_y and v_d velocities (i.e. after projection), introduces projective distortions into the scene flow field. Additionally, as the velocities are in different units, the computed orientation and magnitude is less meaningful.

As an alternative to using $\boldsymbol{I}_o$ it is possible to use the weighted average, of the set of scene particles $\Theta_{0,}x, y$ which lie along ray x, y of sensor 0. This weighted average scene particle $\bar{\boldsymbol{P}}_{0,}x, y$ is defined in equation 6.10, where $\bar{w}_n$ is the normalised scene particle weight, such that the total weight along the ray is 1 (see section 4.3.1). Elements 3,4 and 5 of the scene particle relate to the world velocities $v_{\dot{x}}, v_{\dot{y}}, v_{\dot{z}}$ respectively.

$$\bar{\boldsymbol{P}}_{0,}x, y = \sum_{\boldsymbol{P}n \in \Theta_{0,}x, y} \bar{w}_n \boldsymbol{P}_n \tag{6.10}$$

It is now possible to redefine the orientation (azimuth and elevation) and magnitude images as in equations 6.11 to 6.13. Substituting these images into equation 6.9 to replace their projected counterparts, produces the undistorted HOS descriptor ω^{3d}.

$$\boldsymbol{I}_{azi\text{-}3d}(x, y) = \arctan\left(\frac{\bar{\boldsymbol{P}}_{0,}x, y[4]}{\bar{\boldsymbol{P}}_{0,}x, y[3]}\right) \tag{6.11}$$

$$\boldsymbol{I}_{ele\text{-}3d}(x, y) = \arctan\left(\frac{\bar{\boldsymbol{P}}_{0,}x, y[5]}{\bar{\boldsymbol{P}}_{0,}x, y[4]}\right) \tag{6.12}$$

$$\boldsymbol{I}_{mag\text{-}3d}(x, y) = \sqrt{\bar{\boldsymbol{P}}_{0,}x, y[3]^2 + \bar{\boldsymbol{P}}_{0,}x, y[4]^2 + \bar{\boldsymbol{P}}_{0,}x, y[5]^2} \tag{6.13}$$

6.2.2 Independent Encoding

By moving to a 2 dimensional joint histogram, the descriptiveness of both ω^p and ω^{3d} features, is far greater than the original, optical flow based, HOF descriptor ω. However, if the neighbourhood size and cell sizes are left unchanged, this shift may also lead to sparsity problems. The ratio of pixels within a cell, to the number of orientation bins, will be greatly reduced, possible causing the descriptors to be unreliable. One possible way to combat this, is to use independent (rather than joint) encoding of each

orientation co-ordinate. This allows information about both the azimuth and elevation to be encoded in a more representative way, using all the pixels from the cell. The cost, is that any information contained in the correlation of the two orientations is lost. This approach also leads to smaller descriptors ($2B$ bins per cell rather than B^2), improving train and test speeds.

A simple modification to the cell histogram equation 6.9 leads to the independent histogram for azimuth orientations ω^{azi}, as in equation 6.14. A similar definition can also be made for the distribution of elevation orientations ω^{alt}. The overall descriptor is then formed by the concatenation and normalisation of these histograms, from each cell, as in equation 6.16.

$$\omega_{\epsilon\epsilon}^{3d\text{-}azi}[b] = \sum_{\boldsymbol{o}\in\Omega_{\epsilon\epsilon}} \begin{cases} \boldsymbol{I}_{mag\text{-}3d}((x,y)+\boldsymbol{o}) & \text{if } \frac{2\pi b}{B} \leq \boldsymbol{I}_{azi\text{-}3d}((x,y)+\boldsymbol{o}) < \frac{2\pi(b+1)}{B} \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

$$\omega_u^{3d\text{-}ind} = \left\{\omega_{11}^{3d\text{-}azi}, \omega_{11}^{3d\text{-}alt}, \omega_{12}^{3d\text{-}azi}, \omega_{12}^{3d\text{-}alt}, ..., \omega_{\epsilon\epsilon}^{3d\text{-}azi}, \omega_{\epsilon\epsilon}^{3d\text{-}alt}\right\} \tag{6.15}$$

$$\omega^{3d\text{-}ind} = \frac{\omega_u^{3d\text{-}ind}}{\sum_{b\in\omega_u^{3d\text{-}ind}} b} \tag{6.16}$$

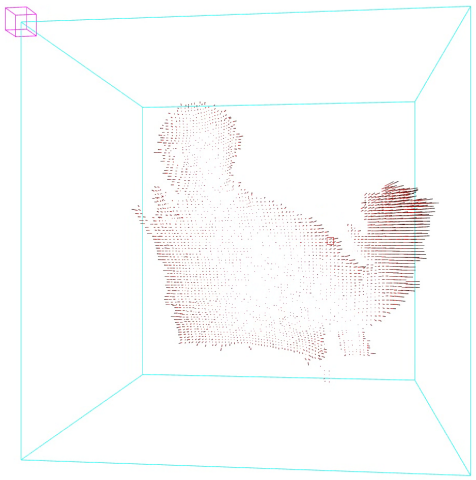
6.3 Histogram of Scene-flow Evaluation

The applicability of the scene particles algorithm and HOS feature encoding is examined below, both in the context of 3D action recognition, and more generally. The scene flow field for all results was calculated with 20 particles per ray, 3 image scales and 2 diffusion iterations. With these settings, the entire Hollywood 3D dataset may be processed in around 2 days, as opposed to thousands of years with traditional scene flow techniques. For HOS encoding four azimuth bins and four elevation bins were used (B_{azi}=B_{ele}=4), unless specified otherwise.

The scene particles algorithm requires estimates of the intrinsic parameters for each camera, and their relative configuration, in order to operate. To this end it was determined that eight of the fourteen films in the Hollywood 3D dataset were recorded using the Fusion camera system, while two used products from 3ality technica (the remaining films did not specify the camera technology employed). Rigs from 3ality are extremely flexible and capable of operating with any camera and lens, thus the parameters obtained from the fusion camera system are taken as a rough approximation of the setup for all films. The system contains a 2/3" CCD, with an aspect ratio of 16:9, implying pixel dimensions of around 0.0053mm (this is subject to some error, as manufacturer specified CCD sizes are often unreliable). Fujinon lenses are used, with focal lengths being of the order of 4mm. The relative configuration of the two cameras may be modified, angling them together or apart to create varying perceptions of the depth in a scene. For these experiments it is assumed that the cameras are close to parallel, with a separation equal to the inter-ocular distance (taken as 65mm). Naturally these parameters are rough estimates only, as the camera may be zoomed, changing its focal length, and the lenses used may differ between shots or films. However, these effects may be mitigated to an extent by the scale invariance of the HOS features, which preserve only the relative distribution of motion strength and not its absolute value.

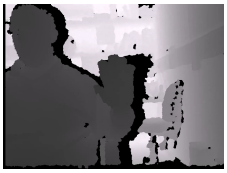
6.3.1 Qualitative

Figure 6.2 illustrates the estimated scene flow field (6.2a), and subsequent HOS encoding (6.2d to 6.2g), of a simple Kinect sequence, showing a person waving their hand and flexing their fingers. Histogram slices relating to differing levels of out-of-plane motion are shown, overlaid on the RGB image, in figures 6.2d to 6.2g. These correspond to slices of the 2D histogram, for elevation bins of (-90 to -45), (-45 to 0), (0 to 45) and (45 to 90) degrees respectively.



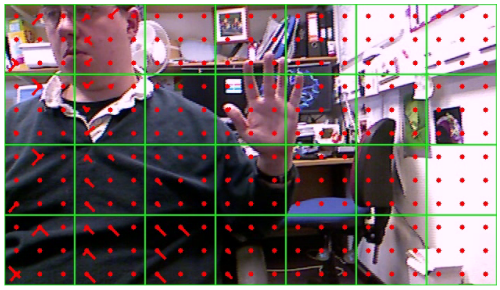
(a) Scene flow

(b) RGB



(c) Depth

(d) HOS encoding (elevation bin 1)



(e) HOS encoding (elevation bin 2)

(f) HOS encoding (elevation bin 3)

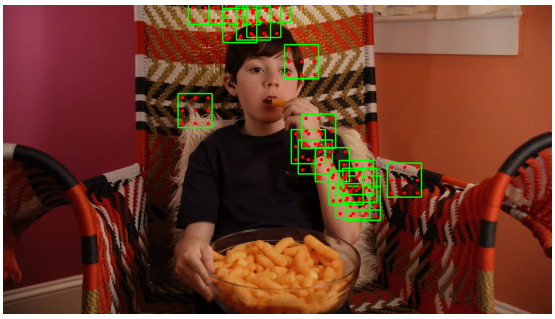
(g) HOS encoding (elevation bin 4)

Figure 6.2: Scene flow and HOS encoding with Kinect. The RGB and depth images from a single frame of the sequence are shown in 6.2b and 6.2c respectively. The estimated scene flow field (with background removed for clarity) is shown in 6.2a. The remaining images show different elevation slices of the subsequent HOS feature encoding, performed densely across the scene (including background motion). Green squares are "blocks" of normalised motions.

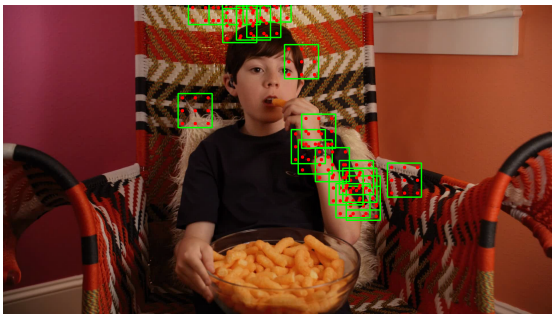
Observing the scene flow field of figure 6.2a it can be seen that the majority of the motion occurs at the hand, with some on the arm and shoulder, and a small amount at the head, all of which are moving to the right. The HOS encoding images reflect this, with the vast majority of motion falling into the first and fourth azimuth bins (i.e. angles from (0 to 90) and (270 to 360)), which are the bins oriented along the positive x axis (towards the right of the image). It is interesting to note that blocks which completely overlap the torso, are encoded with some amount of motion, despite appearing near stationary in the motion field. This is due to the scale invariance of the HOS descriptor, which amplifies the small amount of motion present within the block. If a faster moving region overlapped the block, these low level background motions would become insignificant. However, in the Hollywood 3D task, the use of spatio-temporal saliency measures make it less likely for HOS features to be extracted in stationary regions.

Contrasting the 4 HOS images (i.e. the elevation bins) provides information about the out of plane motion in the scene. For example, motion on the hand, shoulder and arm regions tends to occur in figures 6.2f and 6.2g which relate to motion away from the camera (to different degrees). This reflects the contents of the scene. The amplified motions on the torso and head appear mostly in images 6.2e and 6.2f indicating small amounts of in/out-of-plane motion (of the same order as the in plane motion). Image 6.2d is mostly empty, correctly indicating that nothing in the scene is moving towards the camera.

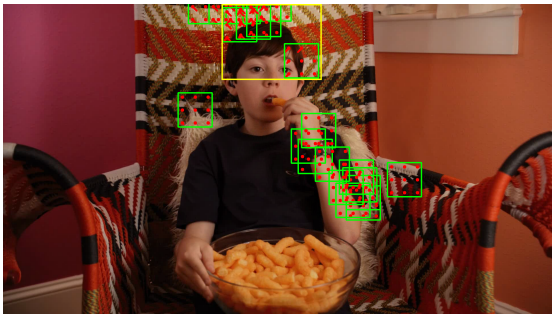
Figure 6.3 shows an example of HOS encoding for an *Eat* sequence, taken from the Hollywood 3D dataset. Figures 6.3a and 6.3b indicate very little motion is occuring towards the camera, while figure 6.3d (i.e. elevations of 45 to 90 degrees, relating to motions strongly away from the camera) displays significant motion at the arm interest points. This motion is uniformly down and to the left, within the image plane, correctly reflecting the motion of the actor, who is moving his hand towards his face (away from the camera) and slightly to the left. Again, regions with small but nonzero motion, such as the areas above the actors head, display some motion due to scale invariance. In particular in figure 6.3e there is some motion towards the bottom of the image, due to small head motions.



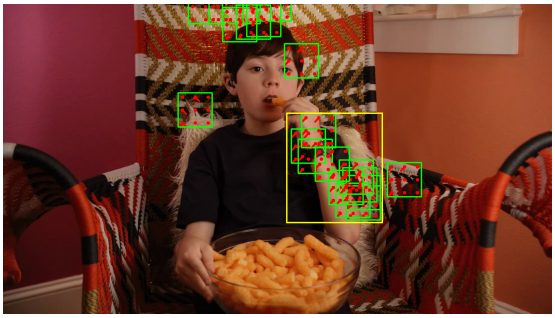
(a) HOS encoding (elevation bin 1)



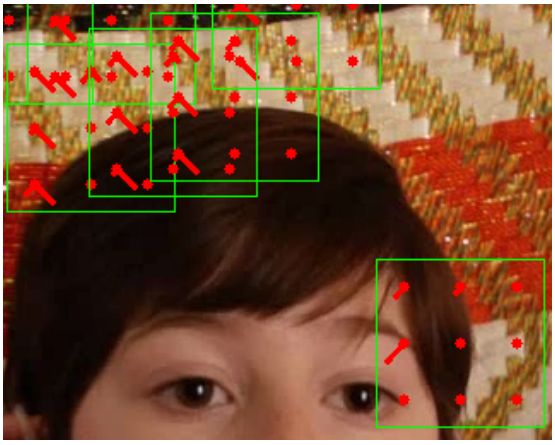
(b) HOS encoding (elevation bin 2)



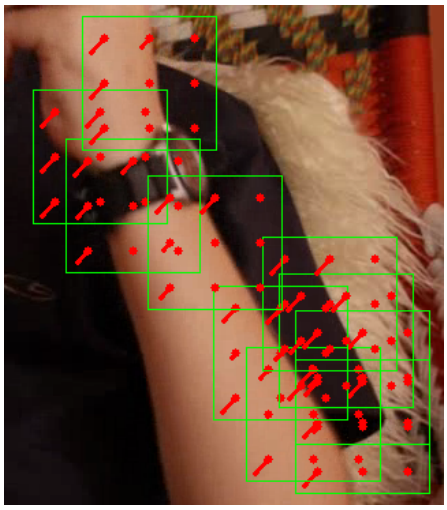
(c) HOS encoding (elevation bin 3)



(d) HOS encoding (elevation bin 4)



(e) Zoomed head region for elevation bin 3



(f) Zoomed arm region for elevation bin 4

Figure 6.3: HOS encoding for Hollywood 3D *Eat* example. A small number interest points have been detected, with partial overlaps of the local features. Green squares represent blocks, with each red dot being the centre of a cell. Images 6.3e and 6.3f show zoomed regions of interest, indicated by yellow boxes in the original images.

6.3.2 Action Recognition

Next, the value of the HOS features is analysed quantitatively on the Hollywood 3D dataset, using the action recognition approach of chapter 3. The 3.5D Harris saliency measure ($F_{3.5D\text{-}Ha}$) and BoVW-4D descriptors are explored, as they were previously shown to offer the best overall performance (see table 3.5), while facilitating comparison to optical flow features. To integrate the new 3D motion features, the HOF elements of the ρ_{4D} descriptors from equation 3.19 are replaced with a selection of HOS based features. Other stages of the pipeline, such as interest point detection, codebook generation, holistic encoding, and SVM classification, remain unchanged, including the appearance (HOG) and structure (HODG) elements of the ρ_{4D} feature vector.

Where relevant, parameters remain unchanged from the previous Hollywood 3D experiments of chapter 3. A codebook size of 4000 is again used, in conjunction with a grid of 3×3 cells, where each cell consists of 11×11 pixels.

Figure 6.4 shows the performance for each type of motion feature, in multiclass and 1 vs 1 (soft voting) detection modes, which were shown in section 3.3.4 to be the most scalable and most accurate, respectively. Feature ω^{2d} is the same as ω^{3d} described in section 6.2.1 with the number of elevation bins B_{ele} set to 1, so as to eliminate information about out of plane motions and approximate a HOF. Feature $\omega^{3d\text{-}dep}$ is a standard HOS, but using the precomputed depth maps provided with the Hollywood 3D dataset as a hard constraint, rather than re-estimating the structure. Feature $\omega^{3d\text{-}int}$ is also the same as the ω^{3d} encoding scheme, but employing an ITF during scene flow estimation (based on F_{dif} features and trained on the cones, teddy and venus middlebury sequences).

The previously noted relationship between detection modes is again observed, with the less scalable 1 vs 1 approach generally providing improved performance over the standard Multiclass classification mode. It is particularly interesting to note that motion features based on scene flow do not always provide improved performance over optical flow based features. When using the scene particles algorithm to simulate a HOF (ω^{2d}) performance is significantly worse than the standard ρ_{4D} approach. This implies that the estimated scene flow field contains systemic errors, such as shearing of the motion

field in some sequences, and temporal inconsistencies within zoom shots. However, the full scene flow histogram ω^{3d} does offer significant gains over its 2D counterpart ω^{2d}, confirming the hypothesis that out of plane motions contain valuable information for action recognition.

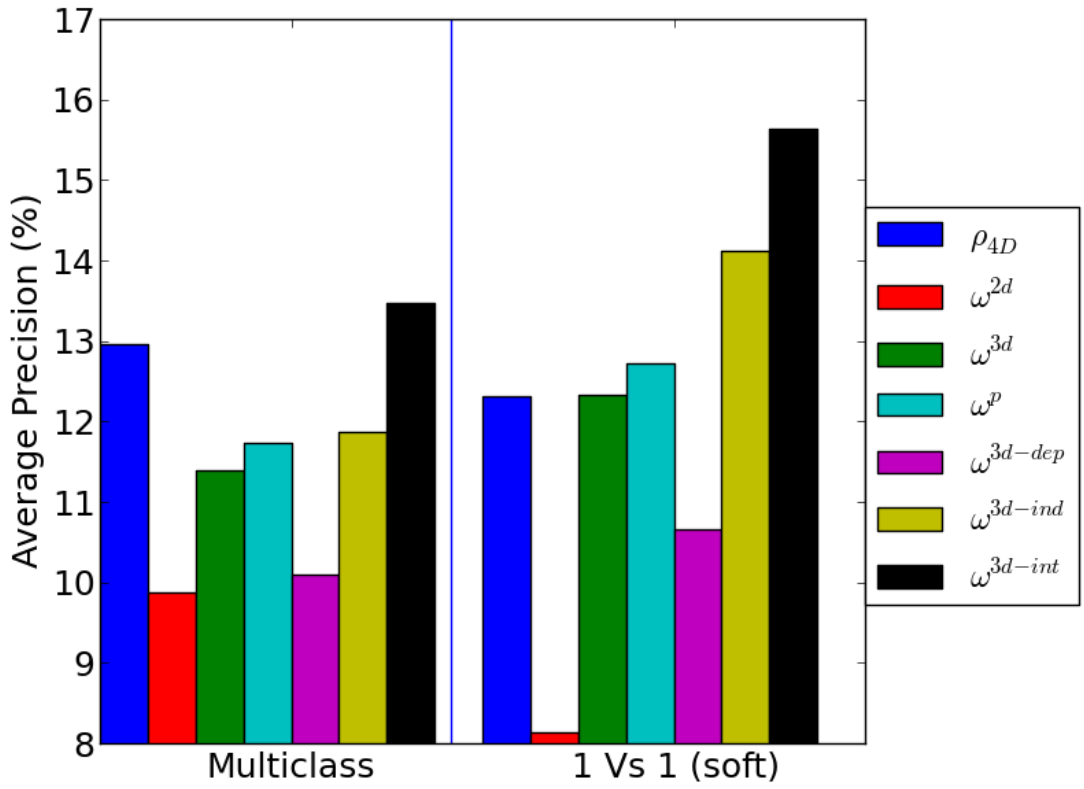


Figure 6.4: Motion feature performance. Standard HOF features compared to a range of scene-flow encoding schemes.

The effects of projective distortion in ω^{p} do not significantly impact performance compared to the undistorted ω^{3d} features, and the independent encoding scheme of $\omega^{3d\text{-}ind}$ also makes little difference within multiclass systems. In the 1 vs 1 detection mode, independent encoding does lead to some performance improvement. This is reasonable, as the instability of sparsely populated histograms may be mitigated to some extent when learning from a large number of training samples. However the simpler sub tasks of the 1 vs 1 scheme have fewer samples available, increasing sensitivity to sample reliability. It is also interesting to note that the success of the independent encoding scheme

implies that little information is contained in the correlation between angles (i.e. the mutual information is low). This is useful as encoding the orientations independently leads to a greatly reduced feature vector and more efficient computation, without a reduction in performance.

The use of precomputed depth maps during scene flow estimation is detrimental to recognition rates. This is likely due to the sparsity of the depth maps, combined with the formulation of the scene particles being based on the dense output of depth sensors. This suggests that in the future, a combined scheme may prove valuable, where structure is estimated in some regions by the scene particles, and in other regions is provided by an external input.

By far the largest gain in performance comes from employing ITFs during scene flow estimation ($\omega^{3d\text{-}int}$). This allows reasonable motion fields to be estimated, even when calibration issues prevent perfect estimation, leading to a significant increase in performance over the optical flow based features ρ_{4D}. This implies that the improved accuracy and convergence properties of ITFs, also lead to increased robustness to camera alignment errors. Further, this demonstrates that ITFs trained on the middlebury datasets are able to generalise very well to more varied data, including indoor and outdoor scenes with a huge variety of illumination types.

The breakdown of performance for each class, using each type of motion feature, is given in table 6.1. Note that the performance of the original ρ_{4D} features is taken from figure 3.6d, and not directly comparable to the multiclass results of table 3.4.

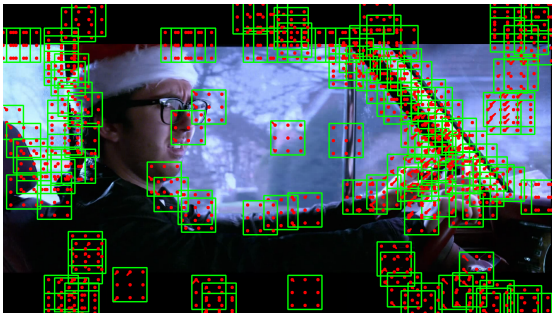
It can be seen that certain types of action such as *Run*, *Drive* and *Swim* lend themselves to scene-flow features, leading to doubled average precisions. An example of a *Drive* sequence is shown in figure 6.5. The most obvious difference to the *Eat* sequence of figure 6.3, is that as the scene is less static, many more interest points are detected. The second observation is that the majority of the motion in the scene occurs in elevation bins 2 and 3 (figures 6.5b and 6.5c) relating to small amounts of out-of-plane motion. Unsurprisingly, most of the motion occurs on the moving background, particularly at the borders with the car windows. The majority of this motion falls into the same azimuth bin, which relates to motion towards the left of the image. This dominant global

Action	ρ_{4D}	ω^{p}	ω^{3d}	ω^{2d}	$\omega^{3d\text{-}ind}$	$\omega^{3d\text{-}int}$
NoAction	12.5	13.8	13.0	10.0	**14.2**	12.0
Run	18.0	23.4	21.5	**47.4**	37.9	26.8
Punch	2.9	11.7	10.9	8.0	8.9	**15.0**
Kick	3.6	10.4	8.1	9.0	9.1	**11.3**
Shoot	16.3	16.8	**24.4**	8.4	17.6	21.0
Eat	3.6	4.0	5.5	**7.5**	6.6	4.2
Drive	35.1	46.2	45.4	20.1	49.0	**51.1**
UsePhone	8.1	8.2	7.8	**9.0**	**9.0**	8.7
Kiss	6.7	6.8	7.0	**8.6**	7.1	7.0
Hug	2.6	3.2	3.5	6.0	4.6	**8.1**
StandUp	8.8	7.1	7.1	10.1	7.6	**10.4**
SitDown	4.3	4.6	4.8	**7.5**	6.0	4.8
Swim	6.4	9.0	**14.0**	13.2	11.3	13.8
Dance	2.8	3.7	3.7	**7.1**	2.8	4.4
Overall	12.3	12.7	12.3	8.1	14.1	**15.6**

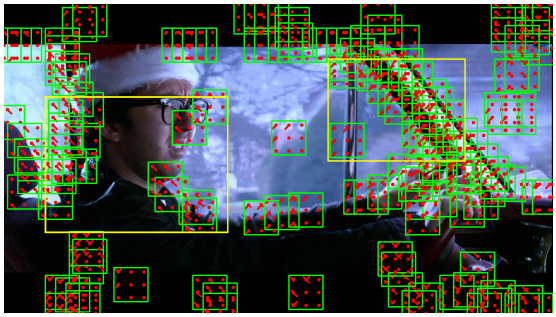
Table 6.1: Motion feature class performance. The Average Precision for each class using each type of motion feature in the 1 vs 1 (soft) scheme. The best performing feature for each class is shown in bold. The previous ρ_{4D} features are taken from figure 3.6d.

translation is also likely to occur in *Run* and *Swim* actions, and depending on camera orientation, may occur towards or away from the camera (e.g. a driving sequence, with the camera on the dashboard or in the back seat). This explains why out of plane motions are so useful for recognising these particular actions, as without knowledge of these strong global translations, there is little discriminative motion information (in this particular example, only the motion of the head and steering wheel, both turning to the right and towards the camera, would be left in image 6.5a).

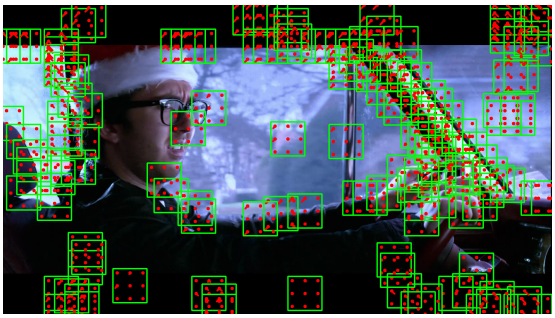
It is particularly interesting to note that the performance of the 2D scene-flow features ω^{2d} differs significantly from the HOF based ρ_{4D} features, despite the fact that they



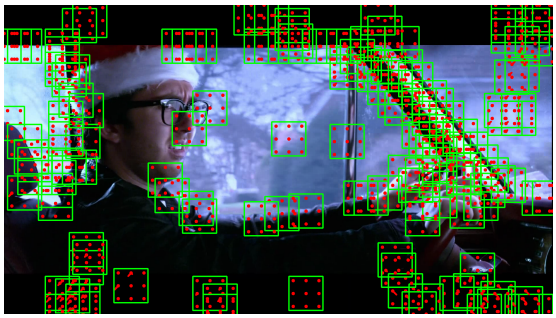
(a) HOS encoding (elevation bin 1)



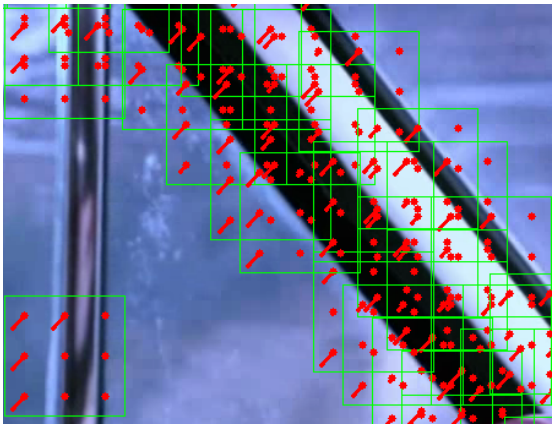
(b) HOS encoding (elevation bin 2)



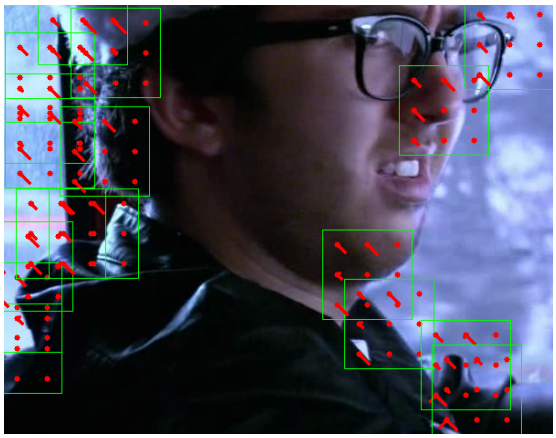
(c) HOS encoding (elevation bin 3)



(d) HOS encoding (elevation bin 4)



(e) Zoomed window region for elevation bin 2



(f) Zoomed head region for elevation bin 2

Figure 6.5: HOS encoding for Hollywood 3D *Drive* example. A large number interest points have been detected, with some overlaps in the local features. Green squares represent blocks, with each red dot being the centre of a cell. Images 6.5e and 6.5f show zoomed regions of interest, indicated by yellow boxes in the original image.

theoretically encode the same information. In general the ω^{2d} features appear to be less effective for classes where ρ_{4D} features work well (such as *Shoot* and *Drive*), while improving classes for which ρ_{4D} fails (such as *Eat*, *Dance* and *Shoot*). These improved actions are often characterised by stationary actors, exhibiting a small amount of more complex motion at their extremities (i.e. hands, arms and legs) as seen in the *Eat* sequence of figure 6.3. It is likely that traditional optical flow schemes are less able to accurately resolve this motion due to smoothing issues, particularly as the interest points occur at object boundaries. In contrast, the lack of smoothing in the scene particles algorithm allows these motions to be recovered more effectively. The 3D $\omega^{3d\text{-}int}$ features also benefit from this lack of smoothing, and exhibit similar or better performance on these particular actions, while also improving more global actions via exploitation of out of plane motion as noted previously.

6.4 Conclusions

In this chapter techniques were presented which allowed the motion features of chapter 3 to be extended to exploit the 3D information in the Hollywood 3D dataset, using the scene flow estimation techniques of chapters 4 and 5.

It was shown that the use of scene flow rather than optical flow doesn't always lead to improved performance, particularly for unreliable flow fields due to inexact camera alignment. However, the use of ITFs during scene-flow estimation provides robustness to this, leading to significantly improved recognition rates. It was also shown that little information is encoded in the correlation of in-plane and out-of-plane motion orientations allowing a shorter and more efficient independent encoding scheme to be employed. Indeed this scheme was actually shown to provide improved performance in cases where the number of training samples is low (such as 1 vs 1 schemes) by reducing the sparsity of the histograms, but this may change with an increased number of training samples.

Further, it was found that the lack of smoothing in the scene particles algorithm was particularly beneficial for recognising certain classes of action, involving small amounts

of complicated motion, such as *Eat*, *Dance* and *Shoot*, even when out of plane information is not exploited. Incorporating the out of plane motion led to greatly improved performance for other actions such as *Run*, *Drive* and *Swim*, which are characterised by large amounts of simple motion (often towards or away from the camera).

As future work, it would be useful to improve the quality of the estimated scene flow fields, either by utilising a larger number of hypotheses or more iterations. Both of these would necessitate improved efficiency such as a parallel GPU implementation, in order to remain tractable for datasets the size of Hollywood 3D. An alternative approach would be to employ automatic estimation schemes for the camera parameters, such as performing a bundle adjustment, or including the parameters within the motion estimation framework [122]. This may lead to more reliable motion fields, however the short length and extreme non-rigidity of the sequences may render such approaches infeasible.

The motion feature encoding may also be developed further in future work. Sparsity issues in the 1 vs 1 scheme may be addressed by allowing motions to cast (weighted) votes into multiple neighbouring orientation bins, or by extending the independent encoding scheme to include some redundancy (i.e. calculate the distribution of orientations on additional non-orthogonal axes). A particularly promising avenue of exploration is the inclusion of rotational invariance in the descriptor. Different orientations of the observing camera, lead to the same action being described as a wide range of different feature vectors, while the distribution in an actor-centred co-ordinate frame remains unchanged. Rotational invariance can eliminate this effect, simplifying the task of action classification by reducing intra-class variations. Also a compensation scheme to remove camera motions, such as that used by Uemura *et al.* [58], could greatly reduce the noise of the HOS features, simplifying recognition.

Chapter 7

Discussion

This thesis set out to explore methods for the effective use of 3D information, in the challenging task of recognising natural human actions. The work was motivated by the recent rise in commercially available 3D footage and sensors, and the eminent suitability of highly invariant depth data, to a task which is characterised by complex intra-class variations.

The first challenge addressed was the lack of any suitable dataset or benchmarks for 3D action recognition "in the wild". To this end, the Hollywood 3D dataset was compiled and a wide range of existing action recognition techniques were extended to make use of the additional 3D information, including 5 interest point detectors and two feature descriptors. The results confirmed that 3D information is indeed well suited to action recognition tasks, with improvements in performance in all cases. Further, it was found that a synergy exists between depth aware interest points and features encoding structural information. Employing both in combination served to focus estimation into areas where structural features are most valuable, leading to a rise in average precision of 30%.

When analysing the structural features, it was noted that varying the saliency threshold lead to differing behaviour for each feature type. RMD descriptors became more accurate with higher thresholds, while bag of words features decreased in performance. This implies that the RMD is more sensitive to noisy features, while BoVW is better able to

extract useful information from a large set of noisy features. In addition, this finding highlighted the fact that sparser features can sometimes lead to improved accuracy in addition to increased computation speed. It also shows that comparisons between feature descriptors may lead to different conclusions at different saliency thresholds.

The next challenge was the incorporation of 3D information into the motion features. Existing techniques for 3D motion estimation were far too computationally expensive to be used in conjunction with a dataset the size of Hollywood 3D. Thus a new formulation for scene flow estimation based on particle filtering was introduced, maintaining multiple hypotheses and exploiting the accumulation of constraints over time. This approach was far more applicable to real computer vision tasks than the previous state of the art, operating orders of magnitude faster while removing over smoothing artefacts which damage motion estimation in the most salient regions of the scene. The technique was also shown to be robust to various types of measurement noise, and to offer considerably improved performance on standard scene flow benchmarks. The developed formulation was flexible enough to operate with any combination of appearance and/or depth sensors in any configuration. To highlight this wide applicability, an alternative example application was demonstrated, tracking hands in 3D during sign language, over 31 hours of HD video.

The scene particles algorithm was then further developed by exploring the inclusion of additional constraints. Occlusion awareness and motion smoothness assumptions were shown to provide large gains in performance at the cost of a significant reduction in speed. It was also shown that when using a traditional, local, approach to smoothness, the performance of the scene particle algorithm is similar to that of previous state of the art techniques (i.e. directional accuracy is improved while all other performance measures are reduced). This suggests that the formulations of smoothness may be a significant limiting factor in contemporary algorithms.

The inclusion of the constant velocity assumption initially appeared to be useful, with no additional computational cost. However, further analysis demonstrated extreme sensitivity to violations of the assumption, making it impractical. This discovery motivated a similar sensitivity analysis for the original brightness constancy assumption,

which underlies the scene particles algorithm (and, in some form, all other motion estimation techniques). Interestingly it was found that all the common formulations of brightness constancy are violated to some extent in real data. Further, these violations were found to be indistinguishable to violations from erroneous motion estimates, implying an inability to distinguish between poorly estimated motions and artefacts of real data. Thus an "intelligent transfer function" was proposed, employing machine learning techniques to create a more general and robust matching criteria. Such functions were found to not only better represent real scenes, but to also translate to significantly improved motion estimation accuracy in both optical flow and scene flow scenarios, with negligible computational cost. ITFs also appear to be valuable for variational approaches, demonstrating superior convergence properties.

Finally the estimated 3D motion fields were exploited to generate more advanced features for action recognition on the Hollywood 3D dataset. The use of ITFs was found to offer robustness to errors in camera alignment, translating to less noisy motion fields and greatly improved action recognition rates. This also demonstrated the generalisation of the ITFs to untrained location types. Results also showed that little information was contained in the correlation of in plane and out of plane orientations, which facilitates a more efficient and compact independent encoding scheme. Indeed this was found to improve performance in certain cases where the unreliability of sparse histograms is an issue. A more detailed breakdown of the performance revealed two categories of actions which benefit from the 3D motion features in different ways. The first category comprises actions characterised by large amounts of strong motion, along a single direction, such as *Run*, *Drive* and *Swim*. These actions benefited significantly from the out of plane motion information, which aided the detection of actions when performed directly towards or away from the camera. The second category of actions involved small amounts of complex motion, such as *Eat*, *Shoot* and *Dance*. These actions were more easily recognised due to the lack of smoothing in the scene particles algorithm, while traditional optical flow techniques often failed to recover the motions.

In summary the contributions made during the course of this thesis include:

1. A large dataset of 3D actions "in the wild", with a wide range of benchmark

results and publicly available source code.

2. Five new saliency functions based on spatio-temporal appearance and structure.

3. A range of depth-aware feature extraction schemes for action recognition (including 2 extensions of previous work, and a number of schemes based on 3D motion information).

4. A novel particle based approach to scene flow estimation, which avoids smoothing errors while operating several orders of magnitude faster than previous techniques.

5. An intelligent transfer function approach, which greatly improves both the accuracy and robustness of motion estimation, allowing 3D motion features to provide significant benefits to action recognition tasks, despite inaccuracies in camera alignment.

7.1 Future Work

There is excellent scope for further development of the work in this thesis. The field of 3D natural action recognition is still in it's infancy, but is certain to become more prevalent in the future. It is hoped that the release of the Hollywood 3D dataset and benchmarks will stimulate the community to explore additional techniques for fully exploiting this information. Some promising avenues for development include more sophisticated feature encoding schemes such as Motion Boundary Histograms [39, 128]) or Volumetric Moments [25, 57]. Development of the depth-aware saliency detectors is also possible by formulating scale invariant versions, which may significantly improve performance for high resolution data such as Hollywood 3D. Integral volumes and parallel processing may also be exploited to drastically increase saliency estimation runtimes enabling a wider range of applications.

The classification stage of the action recognition pipeline was not addressed in this thesis, however dedicated multi-modal learning schemes such as Multiple Kernel classifiers [61] may be better able to deal with the increased sparsity of high dimensional feature spaces, and more effectively exploit the new 3D information. It may also prove

interesting to explore the use of this information in voting based classification schemes [54, 58] which are able to localise actions in time and space. It would be reasonable to assume that structure and scene flow information would enable a full 3D localisation, in addition to improving the accuracy 2D localisation.

The scene particles algorithm also contains significant scope for further development. A particularly promising avenue of exploration is the use of state space smoothing schemes such as Filter Forward Backward Smoothing (FFBS) and the Two-Filter formula. These schemes were developed in the field of statistical modelling, to allow particle filters to approximate joint distributions over time, without degeneracy due to repeated resampling. Put simply these schemes allow the particle filter to use all previous observations, to accurately re-estimate *all previous states* rather than providing only an estimate of the most recent state. In the context of the scene particles algorithm this is extremely useful, as it allows the temporal accumulation of motion constraints to apply both forwards and backwards. As an example, currently if the motion of a point is particularly ambiguous there may be 5 motion hypotheses with significant probability, leading to a poor estimate. At the next frame, 4 of these hypotheses may be disproved leading to a clean and accurate motion field, however there is currently no principled way for the previous noisy estimate to be refined by this new information. The tradeoff is that estimating the joint motion field over multiple frames is naturally more costly, however a balance may be struck using a sliding window approach.

An alternative and simpler approach to handling ambiguous motions would be to develop more sophisticated versions of the interpretation stage (section 4.3.4) in which the particle cloud is converted to an estimated motion field. Recently Basha *et al.* [18] utilised a weighted multi-modal scheme, which may be easily adapted to scene particles. A more efficient GPU implementation of the scene particles algorithm would also be extremely valuable, allowing the application of scene flow for an even wider range of vision tasks, including real time applications. Considerable effort has already been invested in this area, and a publicly available GPU implementation is planned for the near future

A great deal of work remains to be done on the ITFs of section 5.7.3. It would be

particularly interesting to explore their integration into both local and variational motion estimation schemes. In addition it is likely that the approach will yield benefits in other areas of computer vision such as image registration and retrieval.

Finally the exploitation of scene flow as a feature descriptor may be developed further, via the inclusion of rotational invariance and techniques to reduce histogram sparsity. Additional elements could also be incorporated into the estimation pipeline, such as the automatic camera parameters estimation, improving robustness in video categorisation tasks such as Hollywood 3D.

Appendix A

Pose Recognition Using 3D Structure

The work in this appendix explores the use of coarse depth data during instantaneous pose estimation. This serves as a precursor to it's use in the more complex temporal domain of action recognition.

A.1 Texture Features

As a descriptor, Local Binary Pattern (LBP) texture features were used in both the appearance and structure domains. These features provide invariance to monotonic value changes, translating to resistance to illumination changes in appearance and object distance in depth. The features are highly customisable, with the possibility for rotational invariance[91], tunable accuracy and multiple scales.

LBPs describe an image in terms of a histogram of micro-texture components (edges, corners, dark points and light points in the intensity channel, ridges, contours, peaks and depressions in the depth channel). For basic LBP features, every pixel in the image is labelled by taking a 3×3 neighbourhood and thresholding each point by the value of the centre pixel. The result is an 8 bit long binary number labelling the pixel.

$$LBP = \sum_{i=0}^{7} \begin{cases} 2^i & f_i \geq f_c \\ 0 & otherwise \end{cases} \tag{A.1}$$

LBP features were extended to capture texture components at different scales, and also to allow for variable accuracy. The operator LBP(P,R) indicates that, rather than a 3×3 neighbourhood, P points are sampled uniformly around the centre, at a radius R. So R controls feature scale detected, and P controls the length of the output label (and so the size of the feature vector). However there is a limit on the detail possible in the features, dependant on the scale. If P is greater than the number of distinct pixels falling along a circle of radius R, then the new bins being added to the feature histogram are redundant

It was also shown that for most images, 90% of the LBP labels tend to belong to a small subset of the 2^P possible patterns. These patterns were termed "uniform" LBPs and are characterised by having at most two transitions between 0 and 1 in their binary representation. Ojala et al. claim that the removal of these unstable histogram bins also improves classification performance, however our experiments show that if the dataset is large enough, their removal decreases performance.

Another variant of the LBP operator is to add rotational invariance. In order to achieve this, the LBP for every pixel is bit-shifted until the minimum value is found, and this minimum value is used as a label. Equation A.2 defines this conversion, where $shift_i$ represents a binary shift of i bits.

$$LBP^{ri} = min_{i=0}^{P} \left(shift_i \left(LBP^{(P,R)} \right) \right) \tag{A.2}$$

This gives an even greater reduction in feature vector size than uniform LBPs. It is also possible to apply both variants, and use rotationally invariant, uniform LBPs. Histograms of LBP features, for a single LBP variant v, are labelled LBP_v. Several different variant histograms may be concatenated, to provide additional features. These multi-variant histograms may be computed across a subregion r of the object, providing a description of the local texture in that region labelled HR_r. Concatenating these region histograms together forms the feature vector HI_i for the image i. Finally

concatenating image histograms for both the depth and appearance images gives the objects feature representation H.

$$\begin{aligned} HR_{ir} &= \{LBP_0, \ldots, LBP_v\} \\ HI_i &= \{HR_0, \ldots, HR_r\} \\ H &= \{HI_0, HI_1\} \end{aligned} \tag{A.3}$$

In section A.7, the exact effects of the specific feature variants on performance in different tasks is demonstrated. Additionally, by normalising the histogram of textures, the features become invariant to the scale of the detected object.

A.2 Pose Recognition Framework

For classification, a random decision forest is used. This is an ensemble classifier where a large number of decision trees are grown, each based on different random subset of the data. This allows each of the trees to capture different aspects of class separability. The outputs of these weak classifiers are then combined to act as a strong classifier. For the experiments in this appendix a forest of 100 trees with a random sample ratio of 0.6 was used.

One useful feature of random forest classifiers is that it provides a likelihood distribution L over all classes c, given the input observations H. This allows likelihoods to be estimated between classes, somewhat mitigating the drawback of a classification based approach. This likelihood distribution also proves to be an advantage in section A.2.1 where it is used in a particle filtering framework.

$$L(c) = P(H|c) \tag{A.4}$$

A.2.1 Particle Filtering

The particle filter takes the output of the classification stage as an observation likelihood, and combines it with the prior probability of the class $P(c)$, based on the

previous system state and system dynamics. From Bayes theorem, the probability of each class given the new observation, is given by:

$$P(c|H) \propto L(c)\, P(c) \tag{A.5}$$

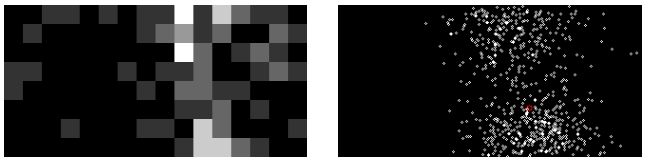
The particle filter approximates $P(c)$ with a number of weighted hypotheses, which are modified from the previous state based on the dynamics of the system with some stochastic diffusion. A resampling step is used to ensure that the higher probability portions of the distribution are more accurately estimated at the next iteration, using a larger number of hypotheses. Each hypothesis in the previous iteration generates a number of new hypotheses, based on it's normalised weight. Equation A.6 illustrates the resampling technique, where $Quant_i$ represents the ith quantile of a distribution. W is the function of normalised hypothesis weights, n is the total number of hypotheses, and S_t is the set of hypotheses at time t.

$$S_{t+1}(i) = S_t\left(Quant_{i/n}\left(\int W\right)\right) \tag{A.6}$$

Figure A.1 shows an example output from the pose classification system (a), being applied to the particle filter. Initially the particles are uniformly distributed. After the classification output is applied, the particles converge towards the peaks of the distribution (b), with more particles centred around higher peaks. This pose tracking allows the pose estimate to be continuously valued, despite initially using a discrete classification methodology.

A.3 Pose Data Collection

The depth information is extracted via stereo point correspondence, from a PointGrey Bumblebee2 stereo camera system. The mask size used causes an unfortunate trade-off between sparsity and accuracy. Smaller masks are harder to match, but provide finer details. In order to provide a more dense depth image, stereo reconstruction is performed with various mask sizes. The images are then combined, using the smallest



(a) (b)

Figure A.1: Likelihood and tracking hypotheses over the pose space. The likelihood distribution (a) across the pose classes, is applied to the pose tracker. The positions of the hypotheses after application of the new likelihoods is shown in (b).

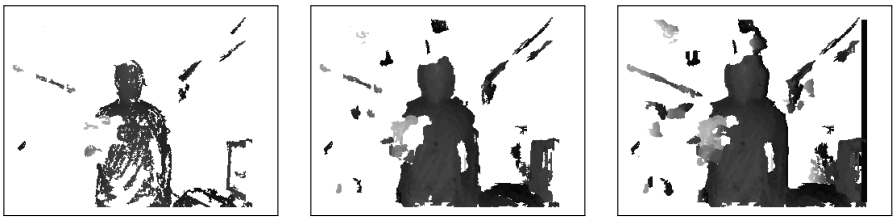


Figure A.2: Combining multiple depth maps. Each successive image is the previous image, combined with an of higher mask size.

mask size wherever possible. Figure A.2 demonstrates this idea, showing a sequence of images each of which has had unmatched pixels from the previous image, filled in by a depth map captured at larger mask size.

A set of depth maps D was generated from the set of stereo masks S by performing stereo point matching on the left and right images (L and R respectively). Where $S = \{15 \times 15, 7 \times 7, 5 \times 5, 3 \times 3\}$ and $stereo_{S_i}$ represents stereo matching with the ith mask.

$$D_i = stereo_{S_i}(L, R) \tag{A.7}$$

The output depth map O is then created by selecting each pixel value $O(p)$ from the corresponding pixel values $D_i(p)$, where D_i is the depth map from the ith stereo mask.

$$O(p) = \begin{cases} D_i(p) & D_i(p) \neq NULL \\ D_{i+1}(p) & otherwise \end{cases} \tag{A.8}$$

A.4 Discrete Head Pose

If the object whose pose is to be estimated, is a subregion of a larger image, then initially the object must be detected. This step is task specific. In the example experiments, head location is extracted using the well known, cascade of boosted haar-feature classifiers technique [127].

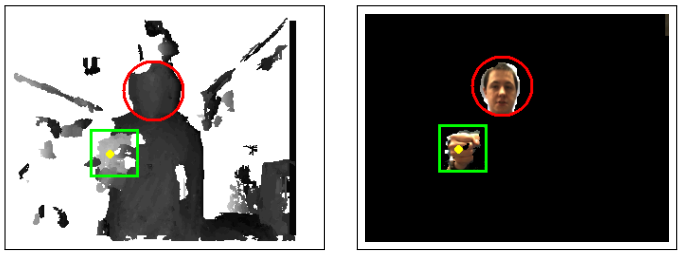
A.5 Semantic Hand Classes

For hand detection a similar detector could be used, however due to the variability possible in human hands it requires large amounts of data to train, and performs significantly worse than with faces [92]. Many other hand detectors simplify the problem, by using segmentation techniques. Segmentation can be performed using background suppression, coloured gloves, motion detection, or skin segmentation [86]. In every case this imposes a restriction on general applicability. Instead, in this work depth images are used to segment the hand, utilising the heuristic that when gesturing at a system, the hand is extended in front of the body.

Using the weak perspective camera model the scale (S) of the object in the image plane stretches between two depths (z_2 and z_1). Thus the resultant scale of an object in the image plane, can be determined by the distance in depth, from an object of known image scale, if their base scale ratio (B) is known, as in equation A.9, where f is the focal length of the camera. In this case the base scale ratio from the face to the hand is taken as 1.2, based on the measurements of the "Vitruvian Man".

$$S = B\left(1 + f\left(\frac{1}{z_2} - \frac{1}{z_1}\right)\right) \tag{A.9}$$

In both tasks, the depth is then used for background suppression. After an object is detected, the median depth of that object is taken. Every image point, with a depth distance further from the median than the expected object size, is suppressed in both the intensity and depth images. This simple heuristic allows operation in noisy and



(a) (b)

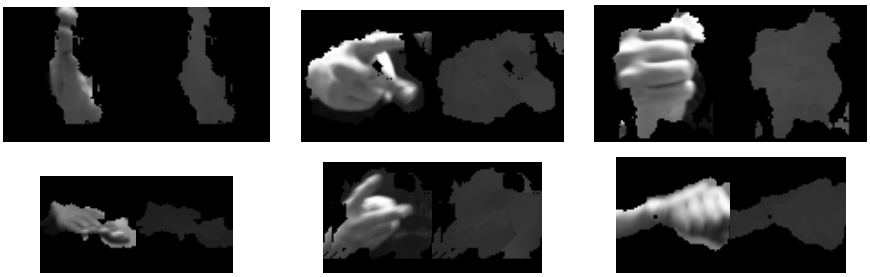
Figure A.3: Head and Hand detection and segmentation. (a) Unsegmented depth image showing face detection (red circle) and nearest point detection (yellow dot), with estimated hand scale (green box). (b) Hand and face appearance after background suppression via depth.

cluttered scenes, without the need for more complicated detection strategies. Background clutter of similar depth to the object is not suppressed by this method, however the objects location and scale have already been estimated, so there is generally little clutter within the small region of interest. See section A.8 for the specific performance increase using background suppression.

Figure A.3 illustrates the hand detection and segmentation. In the first image, the face and closest region of depth are detected, represented by the red circle and yellow dot respectively. The scale of the hand is estimated from the depth difference, and represented by the green box. The second image shows the intensity after background suppression is performed on the 2 objects.

A.6 Pose Recognition Evaluation

Datasets were captured for each task, as there were few pre-existing pose datasets containing appearance and depth information. Both datasets are comprised of subjects from various ethnicities and genders. Performance was measured using 5 fold cross validation, with a random split of 70% training, 30% test images. The training set in each case was enriched by adding small amounts of scale and translation variation to each image. Specifically, each image was translated in all 4 directions by 5% and 10%



(a) (b) (c)

Figure A.4: Hand pose examples. Two randomly selected appearance and depth image pairs from the dataset for (a) Paper, (b) Scissors and (c) Stone. The scale variation between images of the dataset is apparent here.

of it's size, creating 8 additional images, and then the image was enlarged and shrunk by 5% and 10% producing an additional 4 images.

A.7 Hand Pose Classification Results

A test situation for hand pose was required, where the lexicon consisted of a small number of static gestures. A Rock, Paper, Scissors game was determined as a suitable candidate for the trial (see section A.8.2). A dataset of depth and appearance images was created for each of the 3 poses. Seven subjects, including male and female subjects of a wide range of nationalities were asked to create the specific gesture at different orientations and positions. In total 2100 appearance and depth image pairs were captured per symbol (before enrichment). A random selection of image pairs from this dataset is shown in figure A.4. Performance was measured with a number of different feature variants, as shown in table A.1

The first 3 rows of the table illustrate the value of depth. Testing entirely without the influence of depth is impossible in this task, as it is required for object detection, however structural features may be removed from the classification stage. Classification based on depth and appearance features both achieve respectable performance levels, while the combination of the two improves over either alone.

Feature type	Average correct classification	Standard deviation
Un-enriched, Greyscale channel	0.8929	0.0079
Un-enriched, Depth channel	0.8623	0.0054
Un-enriched, Both channels	0.9083	0.0040
LBP(8,1)	0.9689	0.0006
LBP^{U}(8,1)	0.9656	0.0013
LBP^{R}(8,1)	0.8865	0.0018
LBP^{UR}(8,1)	0.8593	0.0014
LBP^{U}(8,1) and LBP^{U}(8,2)	0.9693	0.0043
LBP^{R}(8,1) and LBP^{R}(8,2)	0.8932	0.0022

Table A.1: Hand pose classification performance. Operating with different variants of LBP features. LBP^{U} are uniform, and LBP^{R} are rotationally invariant LBPs.

	Rock	Paper	Scissors
Rock	0.9740	0.0102	0.0188
Paper	0.0200	0.9822	0.0315
Scissors	0.0060	0.0076	0.9497

Table A.2: Hand classification confusion matrix. Uniform, multi-scale (8,1) (8,2) LBPs were used. Rows are predicted classes and columns are true classes.

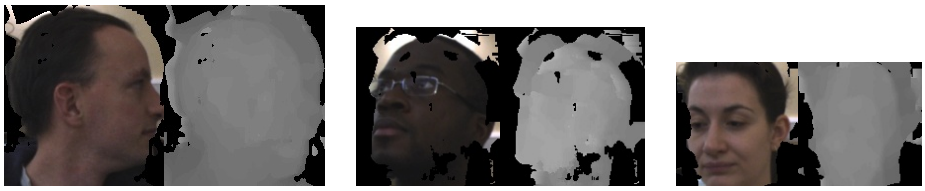
Standard LBP features provide excellent performance. Utilising features across scale does provide slightly improved performance, however in this task, class discrimination is based upon finger location, which may be poorly represented at higher scales. Using Uniform LBPs caused little change, implying that micro-texture components useful for determining finger positions are mostly uniform patterns. This is useful, as removing these patterns means a smaller feature vector, improving both training and test times for the system.

Rotationally invariant LBPs perform significantly worse in all cases, compared to their rotationally variant counterparts. This is likely because rotational variations are so well represented in the dataset, that implementing the invariance within the features is unnecessary.

The confusion matrix is shown in table A.2. The performance on the rock and paper class is significantly higher than on the scissors class. Although scissors examples suffer from higher class confusion, few rock or paper images are classified as scissors. The most prominent features of the scissors class are the two extended fingers. Due to pose, often only the tips of these fingers are visible. So the number of image points useful for identifying a scissors shape may be low.

A.8 Head Orientation Results

The head pose parameters affecting pose direction are pan angle and tilt angle (roll is ignored), these 2 dimensions were segmented into a series of classes at 10 degree intervals. Five subjects, again including male and female subjects from a range of nationalities, were required to sit in a fixed position and look at markers placed at the relevant angle for each class. Haar feature cascades picked out the faces and the background was suppressed using depth. This dataset was far sparser than the hand data, with 153 different classes, and only 1-3 images per subject, per class (2200 pairs of appearance and depth images in total). This sparse dataset makes the task far more difficult, and reinforces the need for a classification based method, capable of operating with little training. As discussed above, situations with sparse datasets such as this,



(a) (b) (c)

Figure A.5: Head Pose examples. Three randomly selected appearance and depth image pairs from the head orientation dataset. (a) -90 degrees pan, -10 degrees tilt. (b) +20 degrees pan, +30 degrees tilt. (c) +10 degrees pan, -20 degrees tilt. Note that scale variations are included in the dataset.

may use feature customisation to incorporate some invariances which are not in the dataset, directly into the feature representation.

The other difficulty with this dataset is the inconsistency of the data. Ten degrees rotation is difficult to capture accurately for the human head, as subjects naturally tend to move their eyes, rather than their heads when looking at close, new objects. This means the dataset tends to have movement between classes of anywhere from 0 to 10 degrees, with the remainder made up by eye motion. Randomly selected example images from the dataset are shown in figure A.5.

Tests were initially performed on isolated images, using a range of feature variants (Table A.3). Classification performance is listed for classifying within 10 degrees of the listed value, reflecting the probable range within the data, as mentioned previously. Using depth to suppress the background from detected objects improves performance by 1%-2% by removing clutter from the images. Using depth as the only feature channel, is more accurate than the standard appearance channel features. However the most effective system utilizes the combination of both feature channels to provide 4% improved performance.

Standard LBP features achieve a respectable 74% classification rate. As expected, the sparse dataset is unable to cover the variations in the classes. Customising the features to suit the task, yields improved results, with uniform LBPs providing the best performance. Due to the sparseness of the dataset, non-uniform feature bins are

Test mode	Average exact classification	Classification within 10 degrees	Standard deviation
No seg. colour features	0.1464	0.6584	0.0202
No seg. depth and colour features	0.1911	0.7225	0.0069
Seg. colour features	0.1691	0.6801	0.0145
Seg. depth features	0.2052	0.7064	0.0114
Seg. depth and colour features	0.2010	0.7398	0.0113
LBP(8,1)	0.2010	0.7398	0.0113
LBP^U(8,1)	0.2845	0.8364	0.0052
LBP^R(8,1)	0.1981	0.6957	0.0103
LBP^{UR}(8,1)	0.2817	0.8107	0.0062
LBP^U(8,1) and LBP^U(8,2)	0.2870	0.8362	0.0094
LBP^R(8,1) and LBP^R(8,2)	0.2043	0.7017	0.0156

Table A.3: Head pose estimation on isolated images, using different types of LBP features and with different usage of depth. LBP^U are uniform, and LBP^R are rotationally invariant LBPs.

Mode	Exact classification	Classification within 10 degrees	Average pan error	Average tilt error
Per frame classification	0.0885	0.4712	N/A	N/A
Pose tracking	0.1081	0.6414	10.0	10.6

Table A.4: Continuous head pose estimation. Head pose estimation on a continuous sequence with and without pose tracking.

unstable, and when present, are mistakenly chosen as discriminatory.

As in the hand pose tests, the results show only a marginal improvement when using features from multiple scale, while using rotationally invariant LBPs causes a considerable drop in performance. This is to be expected as the test dataset does not contain roll variation, and so the rotational invariance is unnecessary.

A.8.1 Pose Tracking Framework

Head pose estimation was also performed on a continuous sequence, rather than a set of isolated images. For this test the particle filtering framework was enabled. The sequence contains partial and complete occlusions of the subjects face, and also frequent, sudden, changes in direction. The results are shown in table A.4.

As expected, applying temporal constraints is useful when determining the current pose. As a result, 15% more examples were classified correctly over isolated classification. In both dimensions the average error angle is roughly one class. Coupled with the fact that 64% of frames are classified within 10 degrees, it can be inferred that most missclassified examples lie within two classes.

Figure A.6 shows the confusion matrices before (a) and after (b) the pose tracking framework was used. The two dimensional arrangement of pan and tilt classes has been flattened into a vector. The tilt angle changes most rapidly, with the pan angle changing every 9 classes. This means that points which are 9 classes apart in the

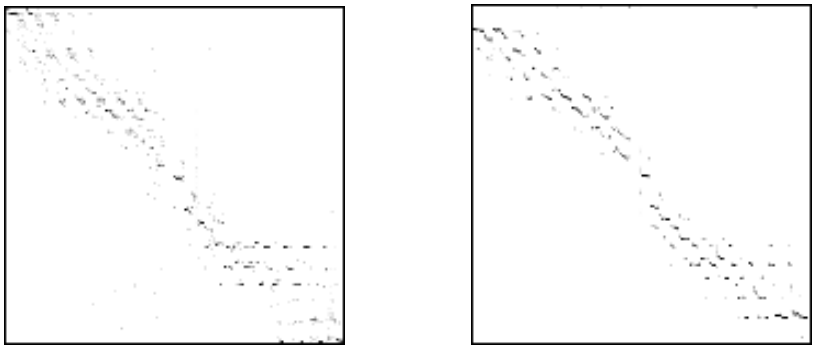


Figure A.6: Head Pose Confusion matrices. (a) without and (b) with pose tracking, for the 153 class head pose task. Darker pixels indicate greater classification rates. The average correct classification rates (within 10 degrees) are 47%, and 64% respectively

confusion matrix, are in reality only 10 degrees apart. This can be observed in the confusion matrix by the multiple diagonal lines, at 9 class intervals.

In the first image (without tracking) there are fewer diagonals visible, and each diagonal is more sharply defined. These two features relate to lower average confusion in tilt and pan respectively. In both cases there are very few extreme outliers, meaning the classification system is able to accurately find the correct region of pose space. A prominent feature of the confusion matrices is the increased number of diagonals present at extreme classes, compared to the central classes. From this it can be deduced that tilt angle is easily determined for a frontal face, but for profile faces (high pan angles) there is greater confusion in the tilt dimension.

A.8.2 Interactive System

In order to demonstrate the systems real-time performance, an interactive demonstration system was built around the hand pose task. This demonstration uses an animated avatar as an opponent for a user to play Paper, Scissors, Stone against. Figure A.7 shows an image of the demonstration system in use. A video of the system is also available at http://www.youtube.com/watch?v=SRfQFOMSH3A.

Figure A.7: Interactive demonstration of hand pose estimation in a Paper, Scissors, Stone.

Appendix B

Additional Motion Function Results

In this appendix, additional results for the brightness constancy and ITF evaluations from chapter 5 are presented for alternative sequences from the middlebury stereo benchmark.

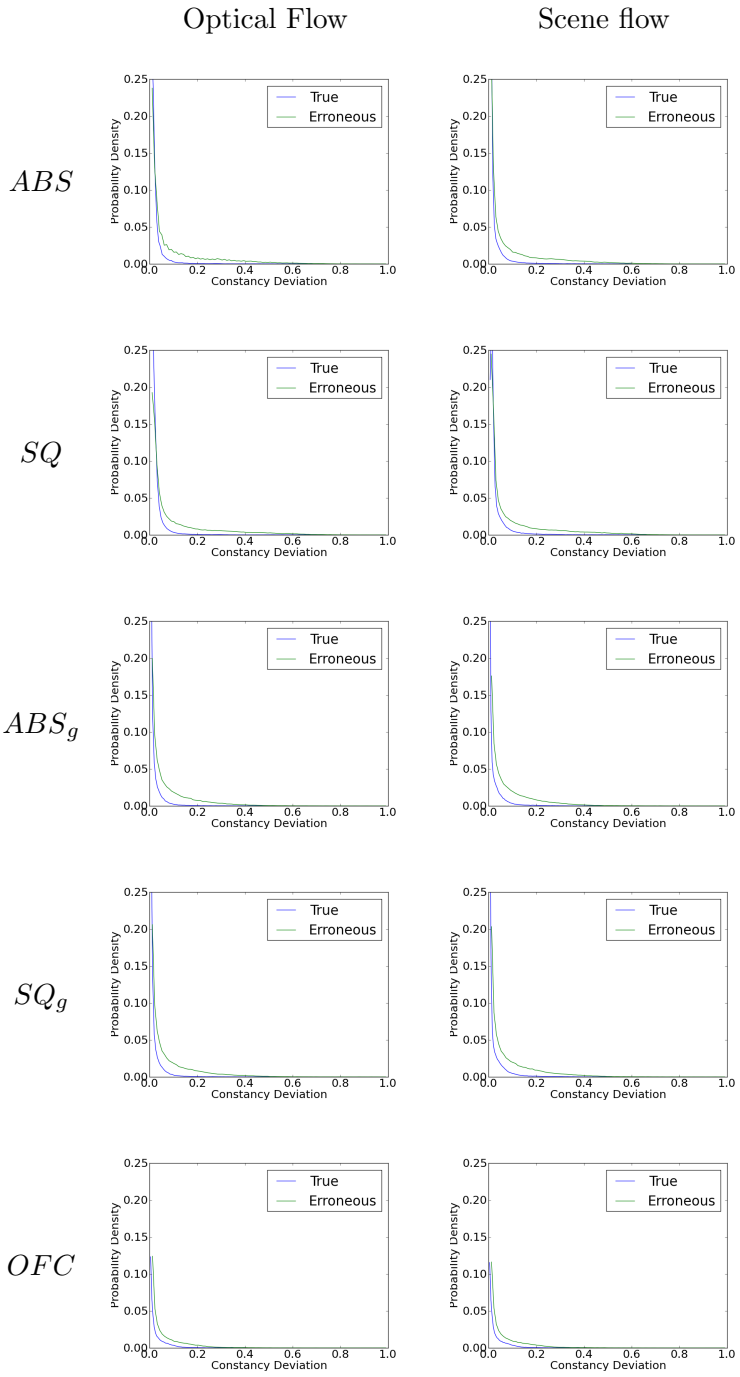


Figure B.1: Response distributions. Distribution of responses for various motion estimation functions, applied to both ground truth and error motions of the venus dataset. Responses are normalized in the range 0 to 1.

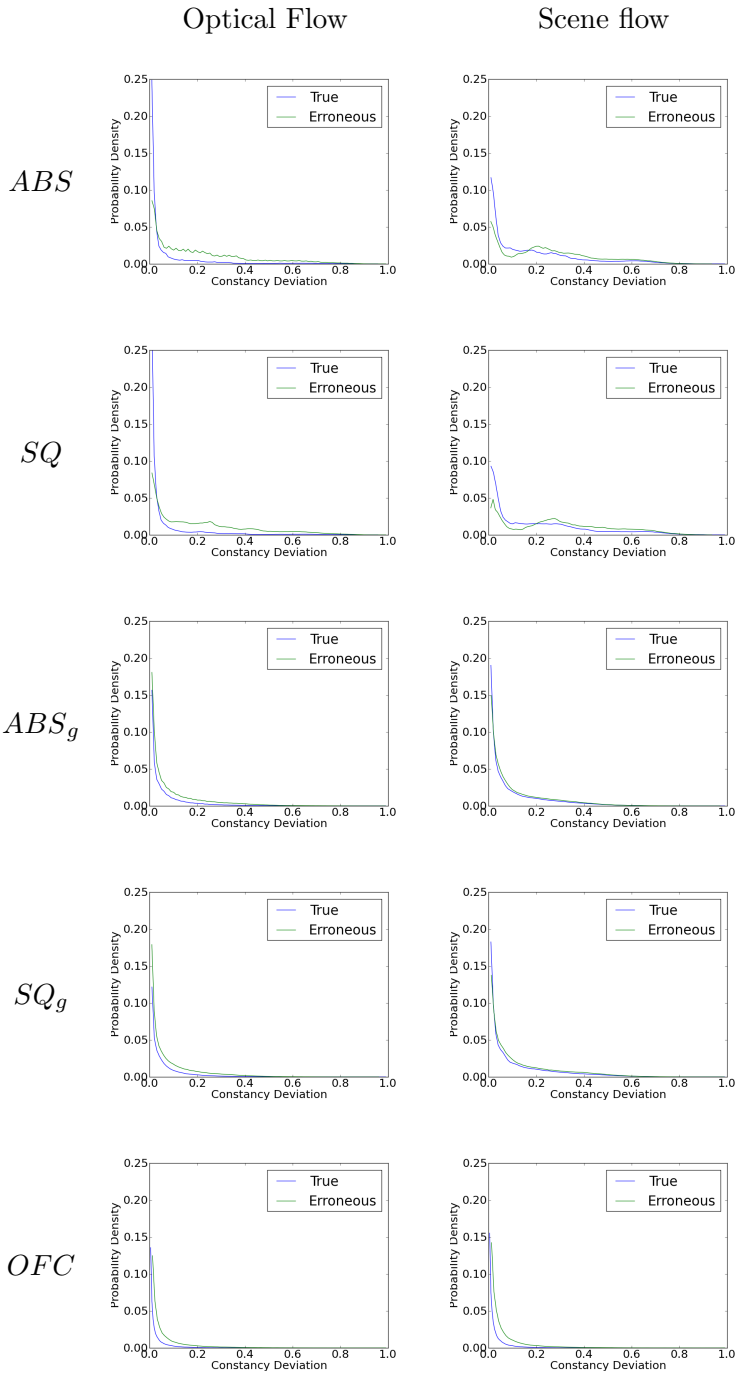


Figure B.2: Response distributions. Distribution of responses for various motion estimation functions, applied to both ground truth and error motions of the laundry dataset. Responses are normalized in the range 0 to 1.

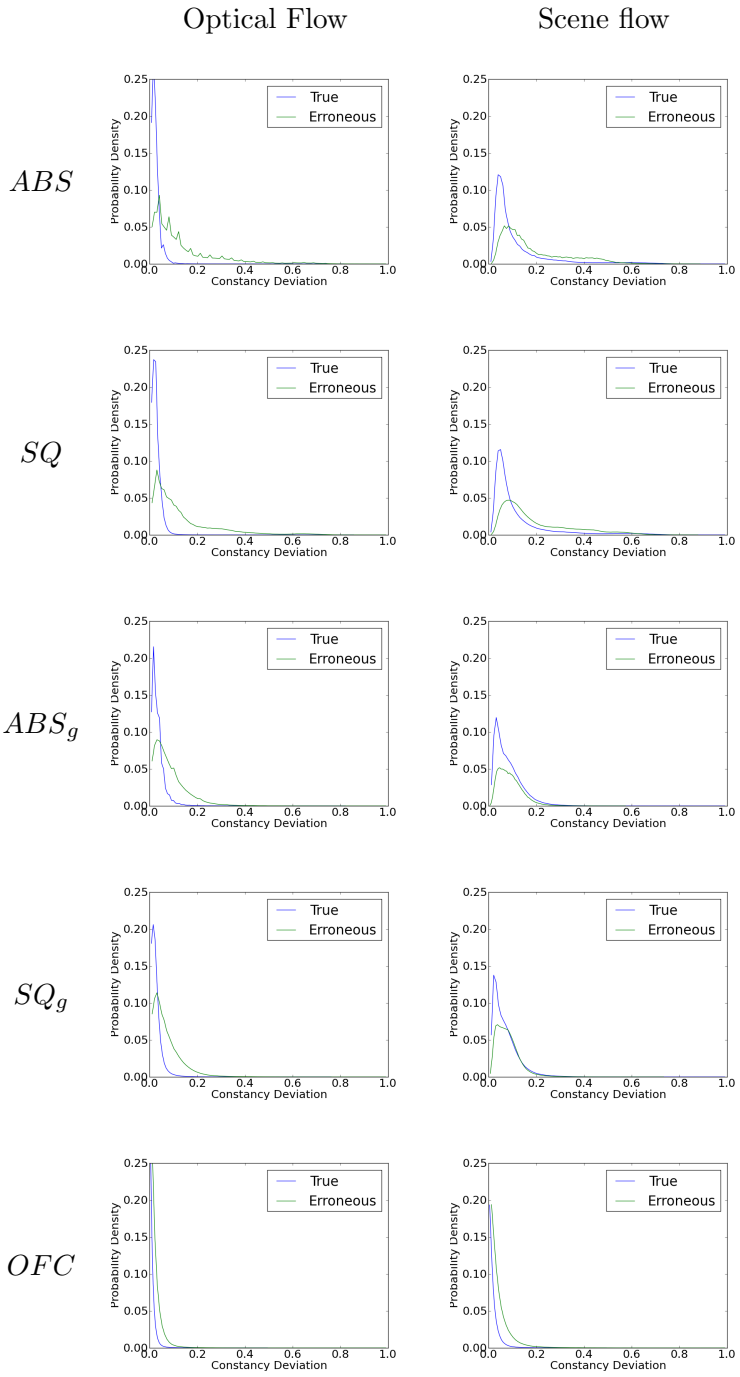


Figure B.3: Response distributions. Distribution of responses for various motion estimation functions, applied to both ground truth and error motions of the wood1 dataset. Responses are normalized in the range 0 to 1.

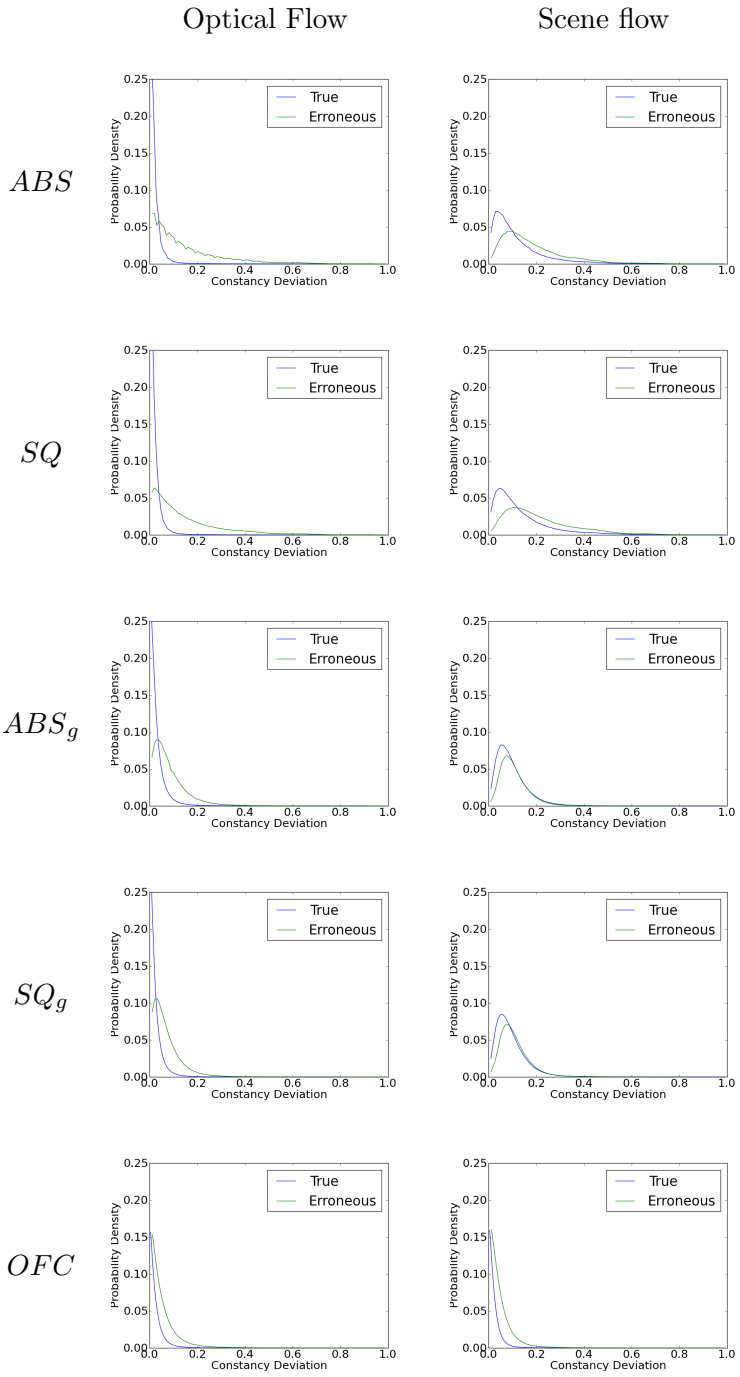


Figure B.4: Response distributions. Distribution of responses for various motion estimation functions, applied to both ground truth and error motions of the rocks1 dataset. Responses are normalized in the range 0 to 1.

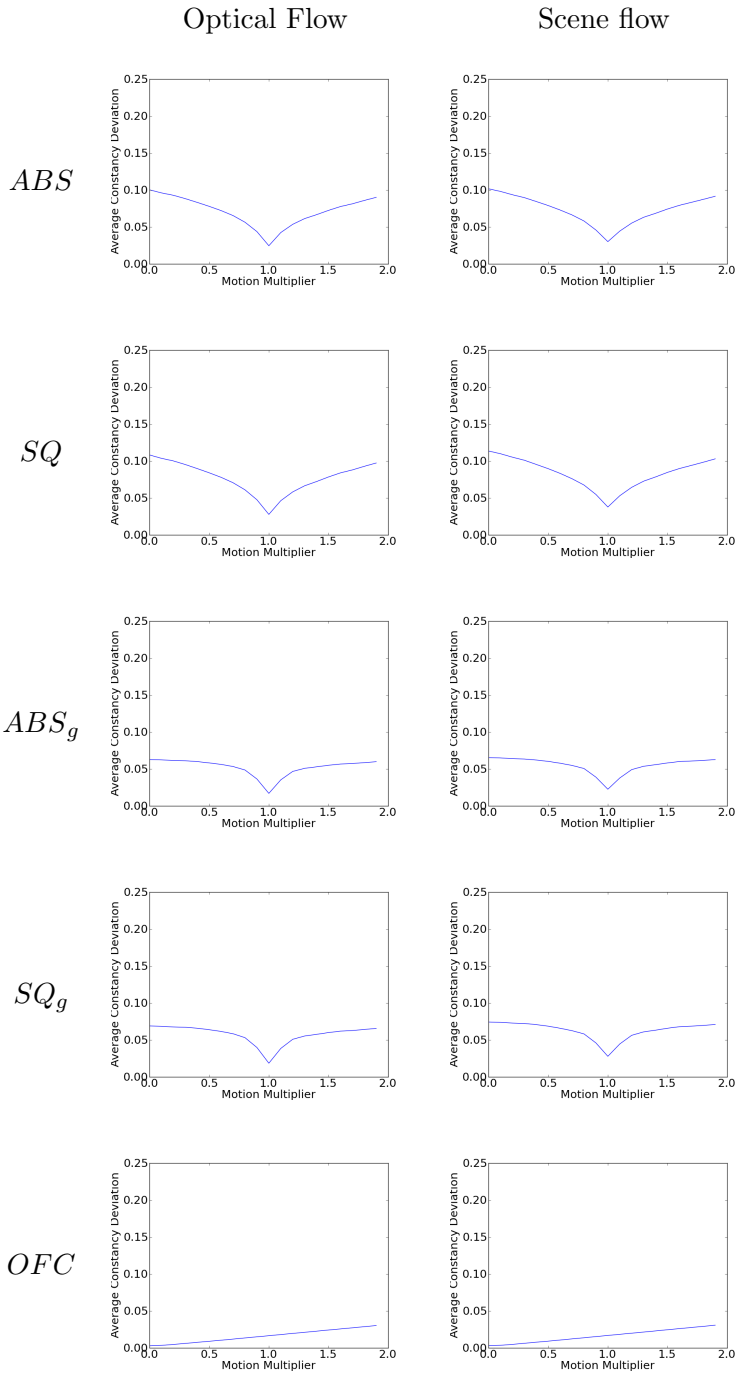


Figure B.5: ITF convergence performance - venus dataset. The average function response for varying levels of motion error, for the best ITFs with and without contextual information. The previous SQ function is also shown, rescaled to the same y axis, for comparison purposes.

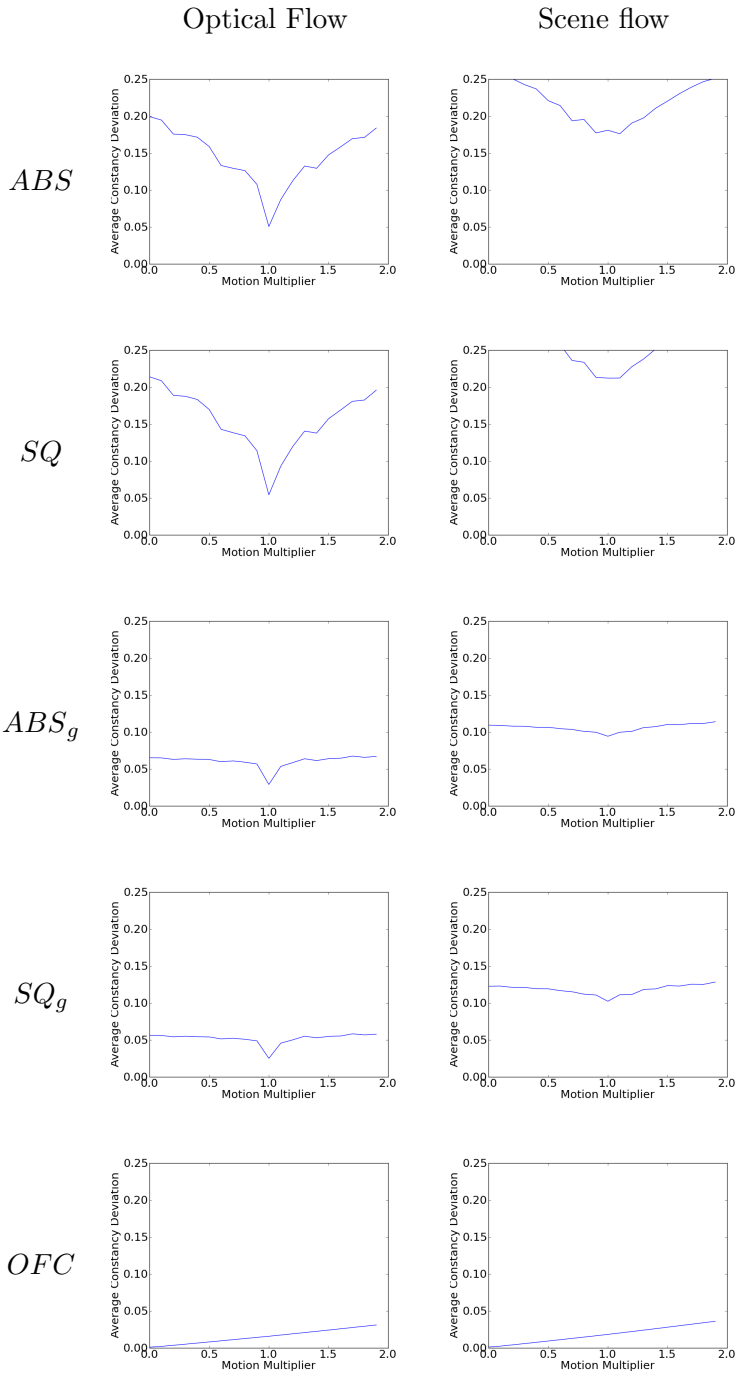


Figure B.6: ITF convergence performance - laundry dataset. The average function response for varying levels of motion error, for the best ITFs with and without contextual information. The previous SQ function is also shown, rescaled to the same y axis, for comparison purposes.

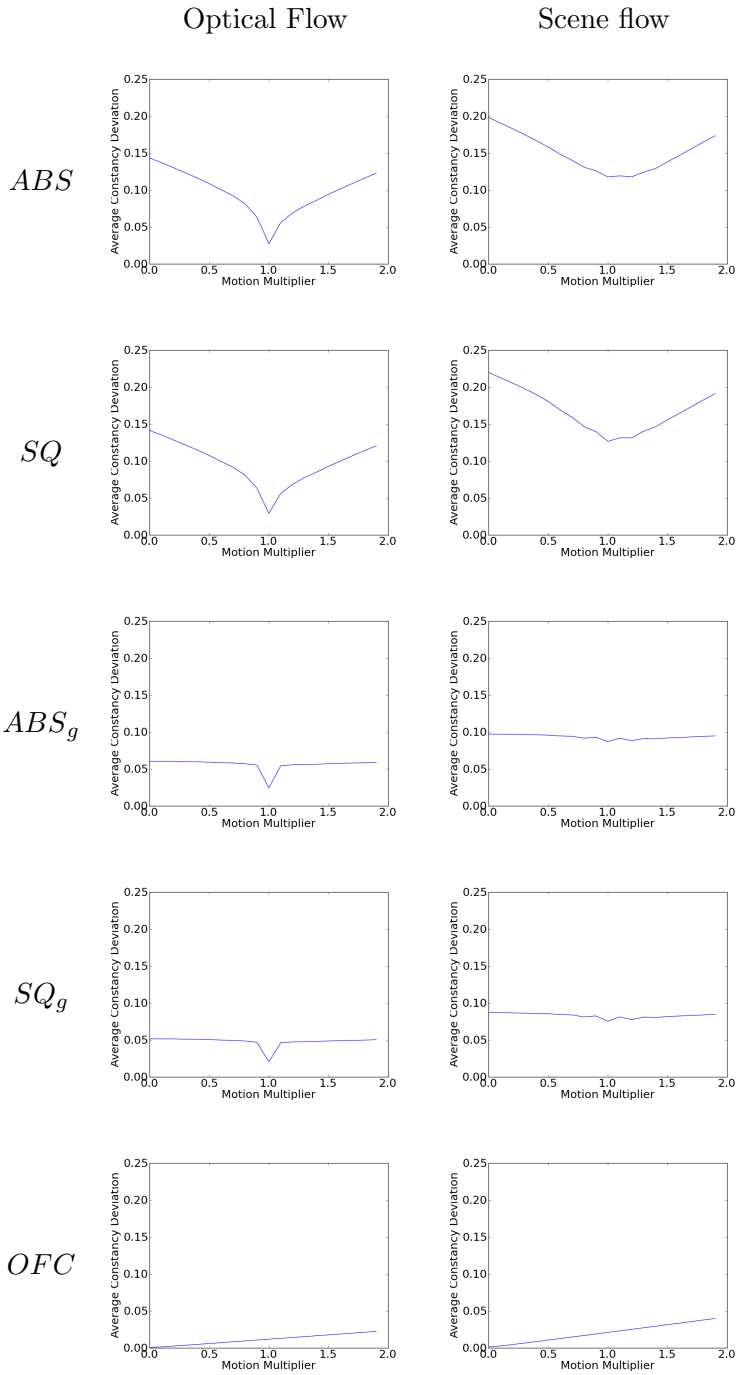


Figure B.7: ITF convergence performance - wood1 dataset. The average function response for varying levels of motion error, for the best ITFs with and without contextual information. The previous SQ function is also shown, rescaled to the same y axis, for comparison purposes.

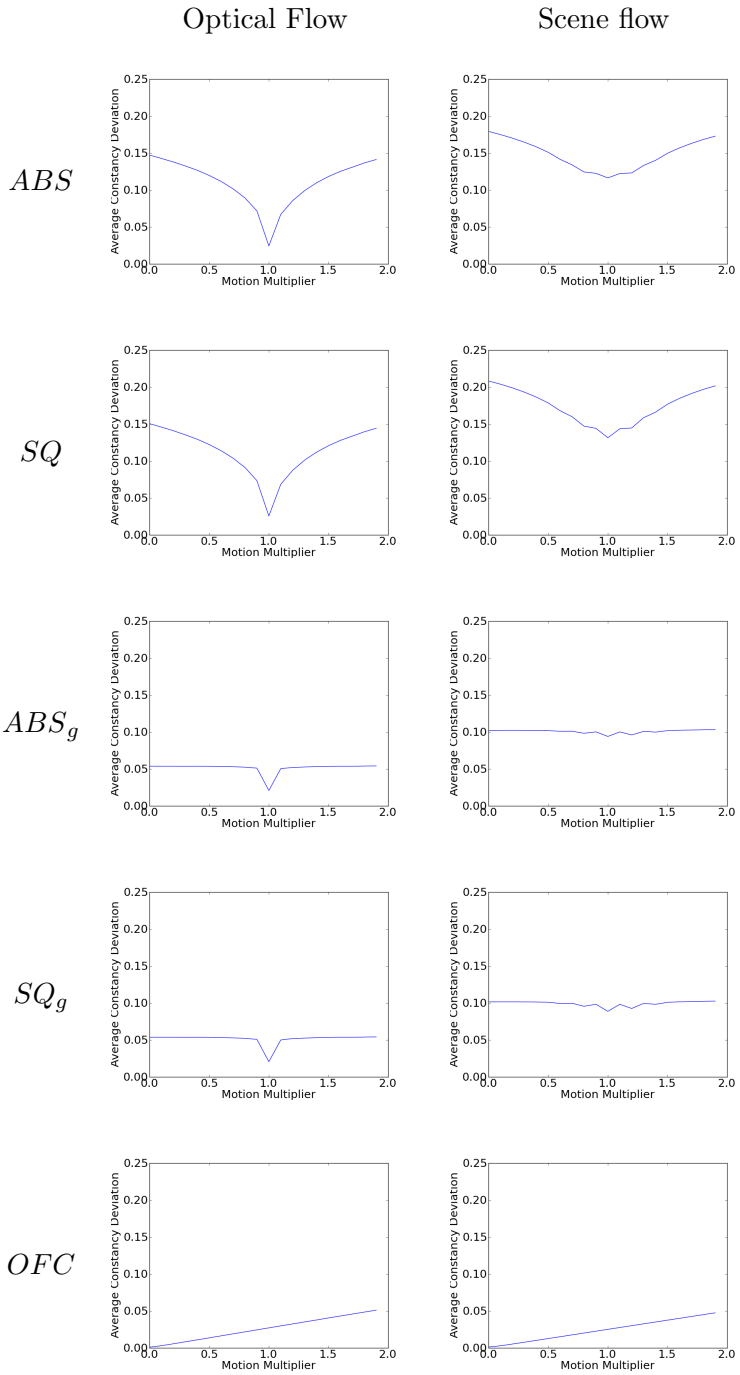


Figure B.8: ITF convergence performance - rocks1 dataset. The average function response for varying levels of motion error, for the best ITFs with and without contextual information. The previous SQ function is also shown, rescaled to the same y axis, for comparison purposes.

Bibliography

[6] *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, USA, June 13 – 15 2000.

[7] *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA, December 2001.

[8] *IEEE International Conference on Computer Vision*, Nice, France, October 14 – 18 2003.

[9] *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005.

[10] *European Conference on Computer Vision*, Graz, Austria, May 7 – 13 2006.

[11] *IEEE International Conference on Computer Vision*, Rio de Janario, Brazil, October 16 – 19 2007.

[12] *European Conference on Computer Vision*, Heraklion, Crete, September 5 – 11 2010.

[13] *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, June 13 – 18 2010.

[14] *ChaLearn Gesture Dataset (gesture.chalearn.org)*, 2011.

[15] *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Colorado, USA, June 21 – 23 2011.

[16] *International Conference on Image Analysis and Recognition*, Aveiro, Portugal, June25-27 2012.

[17] H. Ando. Dynamic reconstruction of 3D structure and 3d motion. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 101 –110, oct 1991.

[18] T. Basha, S. Avidan, A. Hornung, and W. Matusik. Structure and motion from scene registration. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1426 –1433, june 2012.

[19] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [13], pages 1506–1513.

[20] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Leonardis et al. [10], pages 404–417.

[21] P.A. Beardsley, I.D. Reid, A. Zisserman, and D.W. Murray. Active visual navigation using non-metric structure. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 58 –64, jun 1995.

[22] P. Beaudet. Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, pages 579–583, 1978.

[23] C. Berzuini and W. Gilks. Resample-move filtering with cross-model jumps. *Sequential Monte Carlo Methods in Practice*, pages 117–138, 2001.

[24] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 231–236. IEEE, 1993.

[25] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. Tenth IEEE Int. Conf. Computer Vision ICCV 2005*, volume 2, pages 1395–1402, 2005.

[26] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Computer Vision and Pattern Recognition, 1997.*

Proceedings., 1997 IEEE Computer Society Conference on, pages 994 –999, jun 1997.

[27] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[28] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *In Proc European Conference on Computer Vision*, pages 25–36, Prague, Czech Republic, May 11 – 14 2004. Springer.

[29] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Free-form mesh tracking: a patch-based approach. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [13], pages 1339–1346.

[30] R. L. Carceroni and K. N. Kutulakos. Multi-view scene capture by surfel sampling: from video streams to non-rigid 3D motion, shape and reflectance. In *In Proc IEEE International Conference on Computer Vision* [65], pages 60–67.

[31] R. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *Proc. 12th IEEE Int. Symp. Wearable Computers ISWC 2008*, pages 15–22, 2008.

[32] Jan Cech, Jordi Sanchez-Riera, and Radu Horaud. Scene flow estimation by growing correspondence seeds. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [15].

[33] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1932–1939, Miami, FL, USA, June 20 – 26 2009. IEEE.

[34] Zhongwei Cheng, Lei Qin, Yituo Ye, Qingming Huang, and Qi Tian. Human daily action analysis with multi-view and color-depth data. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 52–61. Springer, 2012.

[35] Olivier Chomat, Vincent de Verdiere, Daniela Hall, and James Crowley. Local scale selection for gaussian based description techniques. In *In Proc European*

Conference on Computer Vision, volume 1842 of *Lecture Notes in Computer Science*, pages 117–134, Dublin, Ireland, June 27 – 30 2000.

[36] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 374 –382, may 1993.

[37] J. Courchay, J.P. Pons, P. Monasse, and R. Keriven. Dense and accurate spatio-temporal multi-view stereovision. In *In Proc Asian Conference of Computer Vision*, pages 11–22, Xi'an, China, September23-27 2009. Springer.

[38] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [9], pages 886 –893 vol. 1.

[39] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In Leonardis et al. [10], pages 428–441.

[40] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *In Proc IEEE International Conference on Computer Vision* [8], pages 1403–1410.

[41] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1052–1067, 2007.

[42] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using non-parametric tracking and non-linear optimization. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, page 8, june 2004.

[43] P. Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer, 2004.

[44] F. Devernay, D. Mateus, and M. Guilbert. Multi-camera scene flow by tracking 3-d points and surfels. In *In Proc IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition, volume 2, pages 2203–2212, New York, NY, USA, June 2006.

[45] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. 2nd Joint IEEE Int Visual Surveillance and Performance Evaluation of Tracking and Surveillance Workshop*, pages 65–72, 2005.

[46] R. Douc and O. Cappe. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64 – 69, sept. 2005.

[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[48] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *In Proc IEEE International Conference on Computer Vision* [65], pages 171–177.

[49] Mihai Fgdar-Cosma, Vladimir-Ioan Creu, and Mihai Victor Micea. Dense and sparse optic flows aggregation for accurate motion segmentation in monocular video sequences. In *In Proc International Conference on Image Analysis and Recognition* [16].

[50] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008.

[51] B.R. Fransen, E.V. Herbst, A. Harrison, W. Adams, and J.G. Trafton. Real-time face and object tracking. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2483–2488. IEEE, 2009.

[52] Y. Furukawa and J. Ponce. Dense 3D motion capture from synchronized video streams. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, USA, June 23 – 28 2008.

[53] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Proc. IEEE 12th Int Computer Vision Conf*, pages 925–931, 2009.

[54] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):883 –897, may 2011.

[55] Minglun Gong and Yee-Hong Yang. Disparity flow estimation using orthogonal reliability-based dynamic programming. In *In Proc IAPR International Conference on Pattern Recognition*, volume 2, pages 70–73, Hong Kong, China, August 2006.

[56] N.J. Gordon, D.J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings, Radar and Signal Processing*, 140:107–113, Apr 1993.

[57] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[58] S. Ishikawa K. Mikolajczyk H. Uemura. Feature tracking and motion compensation for action recognition. In *In BMVC*, 2008.

[59] Simon Hadfield and Richard Bowden. Kinecting the dots: Particle based scene flow from depth sensors. In *In Proceedings, International Conference on Computer Vision*, Barcelona, Spain, 6-13 Nov 2011.

[60] Simon Hadfield and Richard Bowden. Hollywood 3d: Recognizing actions in 3d natural scenes. In *In Proceedings, Conference on Computer Vision and Pattern Recognition (CVPR)*, Oregon, USA, June 22 – 28 2013.

[61] Dong Han, Liefeng Bo, and C. Sminchisescu. Selection and context for action recognition. In *Proc. IEEE 12th Int Computer Vision Conf*, pages 1933–1940, 2009.

[62] C. Harris and M.J. Stephens. A combined corner and edge detector. In *Alvey Vision conference*, pages 147–152, 1988.

[63] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[64] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *In Proc IEEE International Conference on Computer Vision* [11], pages 1–7.

[65] IEEE. *IEEE International Conference on Computer Vision*, Vancouver, BC, July 9 – 12 2001.

[66] IEEE. *European Conference on Computer Vision*, Marseille, France, October 12 – 18 2008.

[67] M. Isard and J. MacCormick. Dense motion and disparity estimation via loopy belief propagation. In *In Proc Asian Conference of Computer Vision*, pages 32–41, Hyderabad, India, January13-16 2006. Springer.

[68] Yan Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *In Proc IEEE International Conference on Computer Vision*, volume 1, pages 166 – 173 Vol. 1, Beijing, China, oct. 2005.

[69] Alexander Klaser, Marcin Marszalek, and Cordelia Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, Leeds, Royaume-Uni, September 2008.

[70] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225 –234, nov. 2007.

[71] I. Laptev and T. Lindeberg. Space-time interest points. In *Proc. Ninth IEEE Int Computer Vision Conf*, pages 432–439, 2003.

[72] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008.

[73] I. Laptev and P. Perez. Retrieving actions in movies. In *Proc. IEEE 11th Int. Conf. Computer Vision ICCV 2007*, pages 1–8, 2007.

[74] R. Li and S. Sclaroff. Multi-scale 3D scene flow from binocular stereo sequences. *Computer Vision and Image Understanding*, 110(1):75–90, 2008.

[75] Rui Li and Stan Sclaroff. Multi-scale 3D scene flow from binocular stereo sequences. In *Proc. IEEE Workshop Motion and Video Computing WACV/MOTIONS '05 Volume 2*, volume 2, pages 147–153, 2005.

[76] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010.

[77] Tony Lindeberg. On automatic selection of temporal scales in time-causal scale-space. In *proc Algebraic Frames for the Perception-Action cycle*, 1997.

[78] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–116, 1998.

[79] Jingen Liu, Jiebo Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2009*, pages 1996–2003, 2009.

[80] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, November 2004.

[81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[82] T.C. Lukins and R.B. Fisher. Colour constrained 4d flow. In *In Proc BMVA British Machine Vision Conference*, pages 340–348, Oxford, UK, September 6 – 8 2005. Citeseer.

[83] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2009*, pages 2929–2936, 2009.

[84] S.J. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and vision computing*, 17(3-4):225–231, 1999.

[85] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525 –531 vol.1, 2001.

[86] A. Mitome and R. Ishii. A comparison of hand shape recognition algorithms. In *In Proc Anual Conference of the IEEE Industrial Electronics Society*, Virginia, USA, November2-6 2003.

[87] J. Neumann and Y. Aloimonos. Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision*, 47(1):181–193, 2002.

[88] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [13], pages 1498–1505.

[89] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320 –2327, nov. 2011.

[90] Richard A. Newcombe, Andrew J. Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 127 –136, oct. 2011.

[91] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[92] Eng-Jon Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *In Proc IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17 – 19 2004.

[93] O. Oshin, A. Gilbert, and R. Bowden. Capturing the relative distribution of features for action recognition. In *Proc. IEEE Int Automatic Face & Gesture Recognition and Workshops (FG 2011) Conf*, pages 111–116, 2011.

[94] Olusegun Oshin, Andrew Gilbert, and Richard Bowden. There is more than one way to get out of a car: Automatic mode finding for action recognition in the wild. In Jordi Vitri, Joo Sanches, and Mario Hernndez, editors, *Pattern Recognition and Image Analysis*, volume 6669 of *Lecture Notes in Computer Science*, pages 41–48. Springer Berlin / Heidelberg, 2011.

[95] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006.

[96] V. Pitsikalis, S. Theodorakis, C. Vogler, RC Athena, and P. Maragos. Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *Workshop on Gesture Recognition*, 2011.

[97] J.-P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [9], pages 822–827.

[98] J.-P. Pons, R. Keriven, O. Faugeras, and G. Hermosillo. Variational stereovision and 3D scene flow estimation with statistical similarity measures. In *In Proc IEEE International Conference on Computer Vision* [8], pages 597–602.

[99] J.P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007.

[100] Clemens Rabe, Thomas Müller, Andreas Wedel, and Uwe Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *In Proc European Conference on Computer Vision* [12].

[101] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. Dodgson. Real-time spa-

tiotemporal stereo matching using the dual-cross-bilateral grid. In *In Proc European Conference on Computer Vision* [12].

[102] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Leonardis et al. [10], pages 430–443.

[103] J. Ruttle, M. Manzke, and R. Dahyot. Estimating 3D scene flow from multiple 2d optical flows. In *In Proc International Machine Vision and Image Processing Conference*, pages 1–6, Dublin, Ireland, September2-4 2009.

[104] Houssam Salmane, Yassine Ruichek, and Louahdi Khoudour. Gaussian propagation model based dense optical flow for objects tracking. In *In Proc International Conference on Image Analysis and Recognition* [16].

[105] Michael Sapienza, Fabio Cuzzolin, and Philip Torr. Learning discriminative space-time actions from weakly labelled videos. In *In Proc BMVA British Machine Vision Conference*, Surrey, UK, September 3 – 7 2012.

[106] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 1, 2003.

[107] T. Schuchert, T. Aach, and H. Scharr. Range flow for varying illumination. In *In Proc European Conference on Computer Vision* [66], pages 509–522.

[108] T. Schuchert, T. Aach, and H. Scharr. Range flow in varying illumination: Algorithms and comparisons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1646–1658, 2009.

[109] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proc. 17th Int. Conf. Pattern Recognition ICPR 2004*, volume 3, pages 32–36, 2004.

[110] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 357–360, New York, NY, USA, 2007. ACM.

[111] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35:151–173, 1999.

[112] Jianbo Shi and C. Tomasi. Good features to track. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593 –600, Seattle, WA, USA, jun 1994.

[113] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [15], pages 1297 –1304.

[114] Hedvig Sidenbladh. *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequence.* PhD thesis, Stockholm Royal Institute of Technology, 2001.

[115] H. Spies, B. Jahne, and J.L. Barron. Range flow estimation. *Computer Vision and Image Understanding*, 85:209–231, 2002.

[116] Jonathan Starck and Adrian Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *In Proc IEEE International Conference on Computer Vision* [11], pages 1–8.

[117] H. Strasdat, JMM Montiel, and A.J. Davison. Real-time monocular slam: Why filter? In *In Proc IEEE International Conference on Robotics and Automation*, pages 2657–2664, Anchorage, Alaska, May3 - 8 2010. IEEE.

[118] D. Sun, S. Roth, J. Lewis, and M. Black. Learning optical flow. In *In Proc European Conference on Computer Vision* [66], pages 83–97.

[119] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[120] W. Trobin, T. Pock, D. Cremers, and H. Bischof. An unbiased second-order prior for high-accuracy motion estimation. *Pattern Recognition*, pages 396–405, 2008.

[121] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors - a survey. *Foundations and Trends in Computer Graphics and Vision*, 2008.

[122] Levi Valgaerts, Andres Bruhn, Henning Zimmer, Joachim Weickert, Carsten Stoll, and Christian Theobalt. Joint estimation of motion, structure and geometry from stereo sequences. In *In Proc European Conference on Computer Vision*, 2010.

[123] Tobi Vaudrey, Clemens Rabe, Reinhard Klette, and James Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *23rd International Conference of Image and Vision Computing New Zealand (IVCNZ '08)*, pages 1–6, 2008.

[124] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *In Proc IEEE International Conference on Computer Vision*, volume 2, pages 722–729, Corfu, Greece, September 21 – 24 1999.

[125] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6d. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [6], pages 592–598.

[126] S. Vedula, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):475–480, 2005.

[127] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [7], pages 511–518.

[128] Heng Wang, A. Klaser, C. Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169 –3176, june 2011.

[129] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009 - British Machine Vision Conference*, London, Royaume-Uni, September 2009. CLASS.

[130] Yaming Wang, Junbao Zheng, and Wenqing Huang. Non-rigid structure from motion based on mrf. In *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, volume 1, pages 181 –184, march 2011.

[131] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision*, 95(1):29–51, 2011.

[132] A. Wedel and D. Cremers. *Stereoscopic Scene Flow for 3D Motion Analysis*. Springer, 2011.

[133] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In Forsyth et al. [66], pages 739–751.

[134] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In Forsyth et al. [66], pages 650–663.

[135] Y. Zhang and C. Kambhamettu. Integrated 3D scene flow and structure recovery from multiview image sequences. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [6], pages 674–681.

[136] Ye Zhang and Chandra Kambhamettu. On 3D scene flow and structure estimation. In *In Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [7].

[137] Ye Zhang and Chandra Kambhamettu. On 3-d scene flow and structure recovery from multiview image sequences. *IEEE Transactions on Systems, Man and Cybernetics*, 33:592–606, 2003.