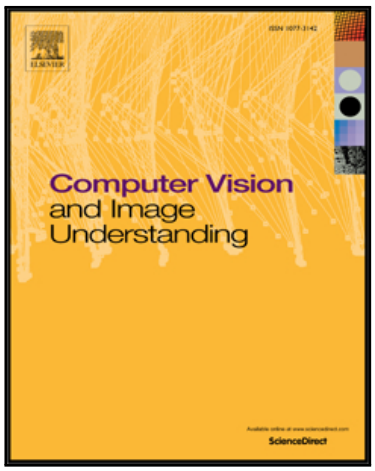


Guided Optimisation through Classification and Regression for Hand Pose Estimation

Philip Krejov, Andrew Gilbert, Richard Bowden

PII: S1077-3142(16)30193-X
DOI: 10.1016/j.cviu.2016.11.005
Reference: YCVIU 2512

To appear in: *Computer Vision and Image Understanding*

Received date: 2 February 2016
Revised date: 10 August 2016
Accepted date: 28 November 2016

Please cite this article as: Philip Krejov, Andrew Gilbert, Richard Bowden, Guided Optimisation through Classification and Regression for Hand Pose Estimation, *Computer Vision and Image Understanding* (2016), doi: 10.1016/j.cviu.2016.11.005

Highlights

- Proposes hand pose estimation using a combination of model optimisation and discriminative methods which allows tracking to be performed at over 40 frames per second using a single CPU thread.
- Introduces a residual error regression for hand pose estimation, learning from mistakes in model optimisation.
- A method of training, which captures system response and user variance, allowing supervised feedback for joint refinement.
- Extensive quantitative and qualitative evaluation including additional datasets and comparison against multiple state of the art approaches.

Guided Optimisation through Classification and Regression for Hand Pose Estimation

Philip Krejov, Andrew Gilbert, Richard Bowden, Centre for Vision, Speech and Signal Processing University of Surrey, United Kingdom

Abstract

This paper presents an approach to hand pose estimation that combines discriminative and model-based methods to leverage the advantages of both. Randomised Decision Forests are trained using real data to provide fast coarse segmentation of the hand. The segmentation then forms the basis of constraints applied in model fitting, using an efficient projected Gauss-Seidel solver, which enforces temporal continuity and kinematic limitations. However, when fitting a generic model to multiple users with varying hand shape, there is likely to be residual errors between the model and their hand. Also, local minima can lead to failures in tracking that are difficult to recover from. Therefore, we introduce an error regression stage that learns to correct these instances of optimisation failure. The approach provides improved accuracy over the current state of the art methods, through the inclusion of temporal cohesion and by learning to correct from failure cases. Using discriminative learning, our approach performs guided optimisation, greatly reducing model fitting complexity and radically improves efficiency. This allows tracking to be performed at over 40 frames per second using a single CPU thread.

Keywords: Hand Pose Estimation, Human Computer Interaction, Hand Tracking, Finger Tracking, Model Optimisation, Random Decision Forest, Discriminative Learning, Regression

1. Introduction

There has been an increasing need for new innovative methods of interaction, providing natural interfaces that can facilitate collaborative computing. The mouse and keyboard are the typical devices used for Human Computer Interaction (HCI), however, they are only suited to conventional computing where the users are sat at a desk. Hand pose estimation offers user control without contact with

the computer as shown in Figure 1 and provides the possibility for future Multi-touchless interfaces [1]. Its use spans many applications, aiding design, remote surgery, robotics, home entertainment and communication. This demand can be seen in Virtual and Augmented reality which provide immersive visualisation with limited interaction.

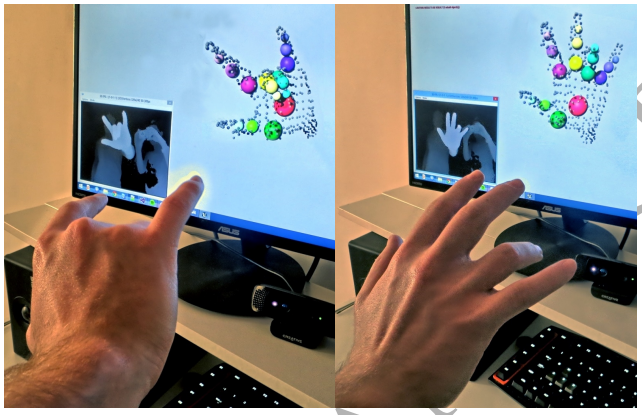


Figure 1: Demonstration of our hand pose estimation method.

Hand pose estimation is a very challenging problem as the hand is an articulated body with many degrees of freedom. This is similar to the challenge of body pose estimation, but with a number of additional challenges. The hand is capable of rapid motion that can break conventional methods of tracking. The hand is also highly dexterous allowing a large number of poses, which varies from the subtle folding of fingers to rapid global rotations. Ideally, an approach must be able to accommodate the large variety in pose while maintaining accuracy for subtle interaction. This requires an understanding of the complex kinematic relationships while being able to identify the different regions of the hand. This is extremely difficult as there is little variation between the appearance of each finger, with ambiguity of the finger arrangement being common. Self-occlusion is also prevalent when using a single camera.

Given this ambiguity, several approaches using gloves and markers have been employed with computer vision to discern similar poses, however, these devices limit adoption in applications outside of constrained environments. For bare hands, existing discriminative approaches utilise large datasets to capture the variety of poses [2, 3, 4] during training. Often, these approaches evaluate the joint positions using a single frame because it does not drift. However, this discards prior information and temporal cohesion, which could be used to reduce the likely

pose space and eliminate jitter. Alternatively, model based approaches [5, 6] depend on temporal information but can become trapped due to local minima in the optimisation process. This has led to the introduction of hybrid methods such as Tagliasacchi [7] where a PCA pose space is used to regularize Iterative Closest Point (ICP) based tracking. Sharp [8], and the Constraint Driven Optimisation (CDO) discussed in this paper also utilise hybrid optimisation. As we will demonstrate, combining both approaches allows a reduction of these mutually exclusive failure cases, improving robustness and accuracy.

This paper provides a combined method for real-time pose estimation, designed to be computationally efficient while reducing failure cases. Global approximation is performed using a Randomised Decision Forest (RDF) which provides a robust, coarse estimate. This is followed by a local optimisation which uses an ICP based framework, refining a parametric model to the hand's depth appearance. Constraint Driven Optimisation (CDO) allows the explicit modelling of kinematic constraints while preventing self-intersection. The combination also provides smooth realistic tracking of the joints. The forest estimate provides implicit reinitialisation, preventing the model becoming trapped in local minima for extended periods.

This paper provides a significant extension to the method and analysis of [9]. We extend the approach to account for discrepancies between a generic model and unseen users. The proposed Residual Error Regressor (RER) provides a correction vector, used to improve tracking capable of learning from the failure cases of prior stages and greatly improves accuracy for unseen users. Our detailed evaluation directly compares against state-of-the-art real-time approaches and demonstrates the strengths and the operational failure cases of our approach, providing state of the art performance on benchmark data.

This performance stems from 3 main contributions. First, the approach to model fitting which uses a combination of model optimisation and discriminative methods. Secondly, improved accuracy with the introduction of residual error regression, which we believe is the first use of such an approach. Finally, our method of training, which captures system response and user variance, allowing supervised feedback for joint refinement.

2. Related Work

A variety of methods have been proposed to resolve hand pose estimation, the following section highlights notable methods and discusses their relevance to our approach.

A number of approaches have been proposed to utilise appearance acquired from RGB cameras. Such methods need to contend with varying lighting conditions and challenging background segmentation. Rehg and Stenger [10, 11] proposed the use of contour based analysis using a framework which considered kinematic limitations, however, the dependency on strong contours proves challenging for a cluttered background. Stenger [12] later utilised a tree structure trained to carve the search space to avoid unlikely poses. The leaf nodes of the tree also offered model parameters that could then be refined through optimisation and allowed tracking against skin coloured backgrounds. De La Gorce [13] removed the need for training using generative modelling of texture and illumination. A comprehensive review of appearance-based approaches was conducted by Erol [14].

Appearance-based estimation can also be assisted using augmentation of the hand with colour markers or a glove. Such approaches offer the best accuracy making them suitable for film production and product testing but at an increased cost with reduced usability. Using LED markers situated across the hand, Aristidou [15] performed motion capture through Inverse Kinematics. As with any active system, there is increased cost and engineering complexity. A lower cost passive solution that used a coloured glove [16] offered less impeded tracking but required a bespoke pattern. An extended review of glove based approaches was conducted by Dipietro [17].

Ideally, hand pose estimation should aim for augmentation free interaction, offering reduced cost of production and allowing widespread adoption and has lead to progress in vision based methods. Hamer [18] was an early approach introducing the use of depth for hand pose estimation. Using structured light, the depth information reduced ambiguity. In combination with appearance, Hamer's approach could handle strong occlusion from grasped objects using part based tracking. However, the complexity of optimisation meant a frame was processed in several seconds. Oikonomidis [19] developed an optimisation approach to resolving the hand using Particle Swarm Optimisation (PSO), which allowed a tractable search of the pose space with near real-time performance using a PSO. Multiple cameras also enabled tracking of object manipulation [20]. Multiple cameras also allow depth to be determined, with Sridhar optimising against both depth and five RGB viewpoints [21].

With generative approaches there is a large overhead in rendering candidate poses, Qian [22] reduced this complexity through sub-sampling the observation and using spheres to model the hand. Melax [5] introduced the use of a rigid body simulation for optimisations which we discuss in greater detail in section 5.1. Using point to surface constraints, the optimisation process operated similarly to

ICP. Alone, ICP would quickly fall into local minima, as such, Melax presented several heuristics that guided optimisation. When using heuristics methods can become trapped in local minima which had not been considered previously, which is why learning methods are preferred. Schmidt [23] instead performed model optimisation using a Signed Distance Function, offering a general approach for body, hand and robot tracking. Similarly to our approach, Sharp [8] proposed using Fern/Jungle based discriminative learning to provide several candidate model parameters and incorporates temporal information using a model, optimised using PSO. This per frame detection provides failure recovery and reinitialisation. Our novel combined approach instead guides optimisation using a single discriminative hypothesis, preforming guided Gauss-Seidel updates to the models parameters, rather than stochastic optimisation steps as with PSO. This reduces computational requirements and as such, operates in real time without multi-threading or a GPU.

With the increase in availability in consumer depth sensors, a number of discriminative approaches have been developed for both hand and body pose estimation. Shotton [24] proposed the use of RDFs as a means of segmenting the body. Mean-shift then localised the joint positions. This was then adapted to hand pose estimation by Keskin [25], trained with synthetic images. Due to the problems associated with high pose variation, Keskin [3] then introduced clustered training to improve classification performance. Training can be performed entirely using synthetic data which is challenging to create, requiring an anatomically accurate hand model. Synthetic data also lacks noise which is characteristic of depth. Noise synthesis was introduced by Xu [26] in an attempt to improve training data. Xu also incorporated kinematic limitations forcing the joint positions to only valid poses. Tang [27] performed knowledge transfer to incorporate real data in training, learning features from both real and synthetic data. Optimising against different objective functions down the depth of the trees also allowed Tang to partition the pose space similar to the clustering performing by Keskin.

Forests were later extended to perform offset regression of the bodies joints using Hough style voting [28]. Kinematic limitations would be implicitly modelled, but would break for unseen examples, hence the need for large datasets. Tang [4] improved the efficiency of forest-based regression using a hierarchical approach, regressing all joints in a single pass of the forest per frame. However, this meant that the approach was vulnerable to the propagation of error and lacked kinematic refinement, leading to invalid poses.

Alternative machine learning approaches have also been used for discriminative modelling. Convolutional Neural Networks (CNNs) can be used to partition the hand regions [29] and provide estimated joint locations [2], although CNNs

are slower to evaluate at runtime than RDFs, impacting real-time performance. Cascaded linear regression has also been proposed for estimating the joints, directly from depth [30]. This method presented both holistic and hierarchical propagation of joint regression. Image retrieval can also be used to resolve the joint positions, inferring the position of joints [31].

Structural techniques also exist, using the shape and contour to localise the fingertips. Krejov [32] and Liang [33] identify the fingertips by considering them geodesic maxima from the centre of the hand. Such approaches are very fast to compute, and in the case of [32], four hands can be tracked simultaneously. However, when fingers are touching, the assumption of geodesic maxima breaks. Despite this, such approaches can be used to assist model optimisation [34].

3. Method Overview

Given a parametric representation, estimating the pose of the hand can be conducted through optimisation, finding the parameters of a hand model that minimises the error between the observation and the models appearance. In the case of optimising against depth, the models appearance is typically rendered as a depth map, which is then compared against the observation. Optimising a hand model using depth is challenging due to the hand's high Degree of Freedom (DoF). Its complex structure and range of local deformation as well as global transformations, mean that many different pose configurations can have similar appearance, leading to local minima that breaks optimisation.

The proposed approach differs in that optimisation is heavily constrained using discriminative segmentation of the hand. A hand model, constructed using 3D bodies was designed to imitate both the shape and limits of real hands. Then optimisation is performed against this physics model, which uses constraints to determine the position and orientation of the hands parts. This provides realistic recreation of the hands motion and temporally smooth estimation.

As shown in Figure 2a, a depth stream is captured using a depth sensor. The depth is then projected using the cameras intrinsic properties to form a point cloud of the hand. This is segmented so that only the hand remains, seen in Figure 2b. The point cloud is then filtered to reduce its density and the amount of noise present (Figure 2c). The pose of the model is then determined using this depth observation through the attachment of spring based, point to surface constraints. These constraints pull the model into position acting similarly to ICP, which minimises the error between model and observation. Through the iterative application of impulse forces, the models position moves closer to the observation

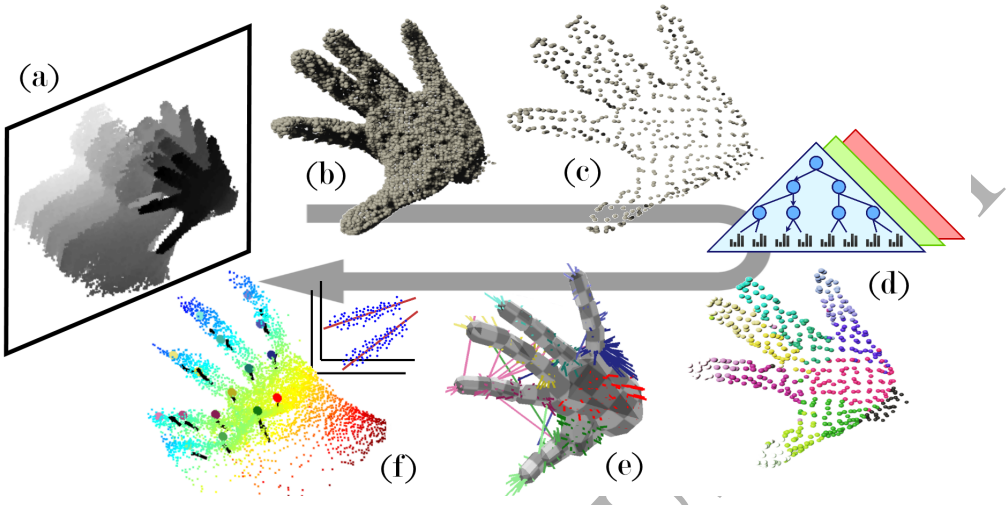


Figure 2: Method overview which shows processing of the depth camera stream (a). The depth is converted to its corresponding point cloud (b) for filtering and sub-sampling (c). Forest classification labels each point using depth (d), allowing constraint-driven model optimisation (e). Linear regression using depth sampled features then corrects model discrepancies(f).

Figure 2e. Unique to our approach, point correspondence for each constraint are determined using a Randomised Decision Forest seen in Figure 2d. This means that optimisation quickly converges to the global optimum solution. The progressive update of model position based on Newtonian dynamics incorporates the temporal information, ensuring realistic dynamic behaviour and reduces inter-frame jitter. For these reasons, the combination of both approaches improves the performance of either alone.

There are two sources of error inherent to this approach. Firstly, the use of a general model for all users without re-targeting, means that residual errors are likely to remain, which limit accuracy. Secondly, as the model fitting is gradient descent and subject to the segmentation of the forest, it is still susceptible to local minima or false minima due to errors in segmentation. For this reason, the final joint positions are refined through cascaded regression, seen in Figure 2f. The result of the earlier model-based optimisation serves as the initial estimate. The cascade itself operates using several tiers which model high-dimensional linear relationships. Each tier is trained to converge to the correct joint location through the regression of update offsets. High dimensional features are sampled from the prior stage capturing local context around each joint. These features are projected through each tier's linear model, iteratively refining the error between the estimate and the correct

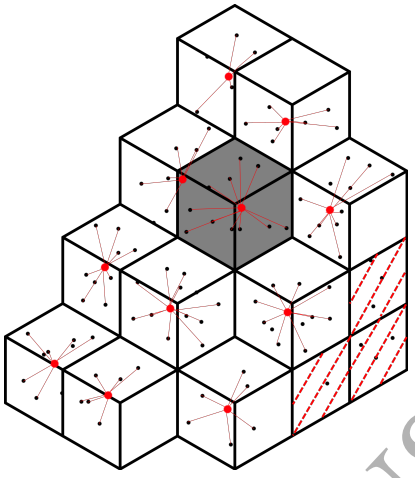


Figure 3: Example of voxel filtering, down sampling a point cloud (black), to its resulting centroids (red). Hatched voxels represent those points (outliers) rejected due to insufficient points.

joint location. This latter stage accounts for hand-model variances and improves accuracy, learning from failure cases previously seen in model optimisation.

4. Preprocessing Depth Data

The hand is first segmented from the image using depth. For the purposes of evaluation, the hand is assumed to be the closest object to the camera. We define the depth of the typical hand as θ_h learned empirically. Our estimate of θ_h includes the wrist as the RDF is trained to label the forearm which provides a robust segmentation. This depth threshold eliminates a large number of pixels belonging to the users head and torso which would add unnecessary computation through the RDF. Additionally, there is less variability in modelling the forearm vs a garbage/non-hand class.

For an image with coordinates Ω we index each pixel $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ and access the depth using $\mathbf{I}(\mathbf{x})$. The pixels of the hand form the subset $\mathbf{I}'$ which are found using the following pass through filter.

$$\mathbf{I}' = \{\mathbf{x} \in \Omega | \mathbf{I}(\mathbf{x}) < \min(\mathbf{I}) + \theta_h\} \tag{1}$$

The remaining points are then refined using the RDF which has an additional class for forearm, discussed in Section 5.4. Alternative methods capable of tracking the users skeleton could also provide robust segmentation.

Each of the hand's pixels are projected using the camera calibration $\mathbf{K}$ to a 3d point $\mathbf{p_x} = (x, y, z)$ constructing a point cloud of the hand $\mathcal{P} = \{\mathbf{p_x} | \mathbf{x} \in \mathbf{I}'\}$. This set of points is very dense as hands captured by short-range depth cameras produce upward of 5000 cloud points. Such a spatial resolution is much higher than required for robust tracking and leads to additional computational complexity. Camera noise around the contour of the hand is also observed. Such erroneous points are likely to impact accuracy.

Intelligently down sampling the cloud to representative points reduces complexity, improving performance of the subsequent stages and is a common practice for model based optimisation [5, 22]. To achieve this, the application of a Voxel grid filter provides both down sampling and outlier rejection and can be seen in Figure 3. Voxels with an insufficient number of points are removed, signified by the red hatches, rejecting outlier noise that forms around the contour. The points (drawn in black) within the remaining voxels are then down sampled. All points contained within the bounds of a voxel are used to compute a centroid. This centroid forms one sample in the less dense cloud shown in red. This process provides $\mathcal{P}'$, a representation of the original cloud that maintains detail while significantly reducing density. The strength of filtering can be tuned by changing the voxel size. The larger the voxel size, the less dense the point cloud, and vice versa. Consideration in choosing voxel size must be given to ensure that small features of the hand are sufficiently represented such as finger tips. The filtered point cloud is then used during optimisation in the following section.

5. Constraint Driven Optimisation

Constraint Driven Optimisation (CDO) is used for model fitting, which performs a combination of model optimisation and discriminative classification. This allows the incorporation of prior knowledge into model optimisation, reducing the complexity of model fitting.

5.1. Rigid Body Simulation

The aim of optimisation is to identify the pose configuration that minimises the error against the filtered point cloud. This is performed using Rigid Body Simulation (RBS) which is most commonly used in film and games [35] and is capable of solving in real time. The approach has been used previously for hand pose estimation [5], solving for constraints that were derived using nearest neighbour assignment. Our approach instead uses machine learning to resolve assignment.

RBS is comprised of a number of components that aim to solve the position and forces applied to bodies, simulating the impact of physics driven interactions between objects. Object collision is examined through a two stage process, with the aim of preventing self-intersection of the fingers. A broad-phase eliminates those bodies too distant to collide, while a narrow-phase confirms and localises the contact point. It is important to note that the simulation takes place in discrete time steps meaning colliding objects intersect. This intersection is computed efficiently for convex bodies using the Gilbert-Johnson-Keerthi distance algorithm (GJK) [36]. On collision, a repulsive pairwise constraint is applied pushing the bodies apart.

System constraints are resolved using a Projected Gauss-Seidel solver which is formulated to reproduce realistic Newtonian physics. This derivation from Newton's laws of motion enforces temporal cohesion and each body's motion state is modelled. Constraints are enforced through the application of impulse forces on a pairwise basis, the direction and magnitude of which, aim to reduce the constraint error. Several iterations are performed over each time step, minimising the error between successive frames. This jointly enforces the kinematic limitations of the joints and the collision constraints applied in the previous stage.

5.2. Kinematic Hand Model

The hand configuration is estimated through optimisation of a generic hand model against the filtered point cloud. Ideally a model must have similar shape and proportions to the real hand it is optimised against and be able to synthesise its dynamic behaviour, which includes realistic representation of joint flexibility. Kinematic constraints enforce the fact that only viable poses are generated, reducing the optimisation search space.

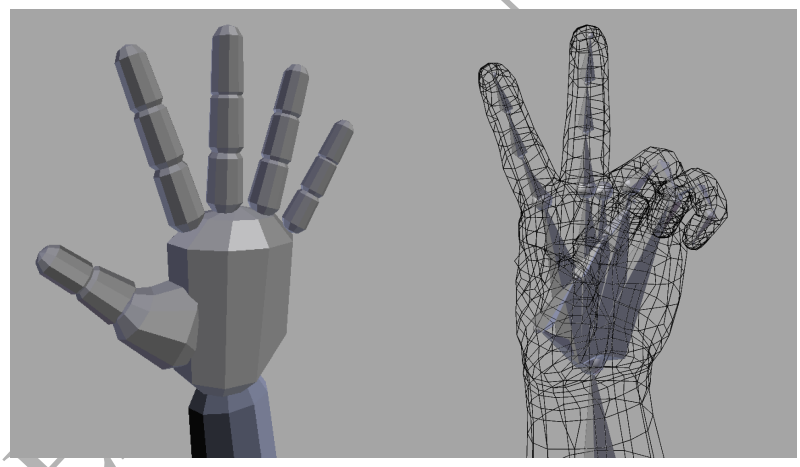
During flexion of the joints, the surface geometry of the hand changes. The inclusion of muscle and bone meshes would add realistic bulging and sliding of the skin [37]. However, our model optimisation omits such complexity as the proposed RERs aims to resolve surface variance in section 6.

Changes in global scale can be adjusted at runtime [5], however changing the proportions of the hand to match a user is more challenging and extends the DoF of the model. User specific models could be constructed [38], but would require a calibration stage. To avoid this, our approach optimises a generic model and refines the pose against such variance through RER.

The use of a mean hand allows tractable optimisation that generalises across users. Morphological surveys of the human hand [39] measure such variance. Using this information and multiple reference images across several users, a general hand model was constructed. The hand model $\mathcal{H} = b_1 \cup b_2 \cup ... \, b_n$ is

comprised of $n = 16$ bodies shown in Figure 4a, three capsules per finger and thumb while a single body models the palm. This use of a single mesh for the palm is suitable for optimisation but there is limited flexibility across the metacarpals. Each of the bodies is connected through a skeletal hierarchy rooted at the wrist using hinge and rotational constraints to reflect the anatomical structure. The skeletal structure can be seen in Figure 4b and can be attached to a weighted model for realistic animation. These kinematic limitations are applied with ranges that match those proposed in [40]. For the purpose of segmenting training data, the forearm is also modelled with realistic kinematics.

By performing the optimisation in an RBS framework, temporal tracking and prediction is implicitly modelled. The mass of each hand component impacts the acceleration during the application of constraining forces. Assuming constant density of the hand; the mass of each component is estimated using the volume, which is sufficiently accurate. This also accounts for the fact that fewer constraints pull the smaller parts of the hand, but should converge at a similar rate.



(a) (b)

Figure 4: Kinematic hand model constructed in Maya. (a) The hand is broken into separate rigid bodies, joined through constraints to reflect realistic limitations. (b) The model is also rigged with a poseable skeleton allowing the generation of training data, and use in 3D applications.

5.3. Point to Surface Constraints

Point to surface constraints ensure contact between a fixed point $\mathbf{p}$ and the surface $\mathbf{s}$ of a rigid body $\mathbf{b_i}$. The constraint solver determines the impulse forces

needed to minimize the constraint error. This error is calculated as the residual between $\mathbf{p}$ and $\mathbf{p}'$, where $\mathbf{p}'$ is the closest surface point.

$$\mathbf{p}' = \underset{\mathbf{p}_b \in b_i}{\arg\min}(||\mathbf{p} - \mathbf{p}_b||) \tag{2}$$

The closest point needs to be updated during the simulation to reflect changes in position and orientation of the constrained body. This again would be computational expensive to perform without the use of the GJK distance. GJK is an optimised method to determine intersection/closest points between two convex polygonal objects. The approach was modified to find the closest point on a body b_i to a point $\mathbf{p}$ in 3D space, which serves as the attachment point for the constraint. This location updates at each iteration, acting as a point to surface constraint.

The point cloud captured from a depth sensor represents the visible surface of the hand, as such, a sample $\mathbf{p}$ in the point cloud must correspond to a position on the camera facing side of the hand. It is challenging to determine where on the hand this point resides, due to the lack of textural information. An exhaustive search for model configuration that satisfies these constraints is not tractable. Searching locally to a prior estimate reduces the search, but during rapid motion, large transitions between frames lead to local minima. Such errors are difficult to detect and recover from. Instead, we estimate correspondence to the model using a Randomised Decision Forest (RDF) which utilises global spatial context. RDFs have been proven to be fast and accurate for region segmentation of the hand and body[25, 24]. This Constraint Driven Optimisation (CDO) allows the incorporation of a priori knowledge from training when optimising the hand model.

5.4. Assignment using Random Decision Forest

Conventional model fitting approaches especially those that employ global/local optimisation, require initialisation. This can be done by manual alignment of the hand with the model [20] or through near exhaustive searching [6]. Fingertip localisation, using structural or template based detection can also be used for initialisation [5]. However, this is dependent on the user forming a pose where all the fingertips are visible. During natural interaction, this is frustrating, particularity when tracking is already failing.

Our use of a Randomised Decision Forest (RDF) to segment the hand into regions allows pose recovery from a range of hand shapes. One of the major benefits of our approach is a fast classification of the hand's regions, allowing part based model correspondence.

A Randomised Decision Forest $\mathcal{F}$ is an ensemble classifier, comprised of multiple classification trees $t \in \mathcal{F}$. Each tree is trained using a random partition of the training data leading to an improved accuracy and robustness over that of a single tree. This improvement is attributed to stochastic discrimination, allowing the forest to generalise to unseen examples.

During forest training, discriminative features are identified that partition the training samples. The following splitting criteria $F_{\mathbf{u},\mathbf{v}}$, partitions the data based on depth comparisons that can identify surface and boundary features and is used in several previous depth based methods [24, 25, 4]. These features ϕ consist of two random offsets $\mathbf{u}, \mathbf{v}$ uniformly distributed over the range r_{max}, and used in combination with a random threshold $\theta < \theta_d$. The lengths of $\mathbf{u}$ and $\mathbf{v}$ are normalised by the samples depth, allowing scale invariance.

$$F_{\mathbf{u},\mathbf{v}}(\mathbf{I}, \mathbf{x}) = \mathbf{I}\left(\mathbf{x} + \frac{\mathbf{u}}{\mathbf{I}(\mathbf{x})}\right) - \mathbf{I}\left(\mathbf{x} + \frac{\mathbf{v}}{\mathbf{I}(\mathbf{x})}\right) \tag{3}$$

The parameter r_{max} can be modified to evaluate more global context for the features and its effect is explored in section 8.1. It should also be noted that the features are not rotationally invariant, as such the training data must encapsulate global rotation. This is achieved through synthetically rotating the dataset through 22.5° intervals. Alternatively the features offsets could be rotated by an in-plane rotation[26], but this requires the addition of an initial regression stage to determine the hand's orientation.

At runtime, each point $\mathbf{p}$ in the point cloud $\mathcal{P}'$ is projected on to $\mathbf{I}$. This point is then propagated down each tree and the class probabilities are aggregated across all the trees in the forest.

$$P(l|\mathbf{I}, \mathbf{p}) = \frac{1}{|\mathcal{F}|} \sum_{t \in \mathcal{F}} P_t(l|\mathbf{I}, \mathbf{p}) \tag{4}$$

The class with maximal likelihood is then assigned to the point.

$$L(\mathbf{p}) \stackrel{\text{def}}{=} \arg\max_{l \in 1..n} (P(l|\mathbf{I}, \mathbf{p})) \tag{5}$$

$$\mathbf{p}' = \arg\min_{\mathbf{p}_b \in b_i} (||\mathbf{p} - \mathbf{p}_b||), \text{where } i = L(\mathbf{p}) \tag{6}$$

Once each points label is found, point to surface constraints are assigned to their appropriate body. The rigid body simulation is then solved, using a projected

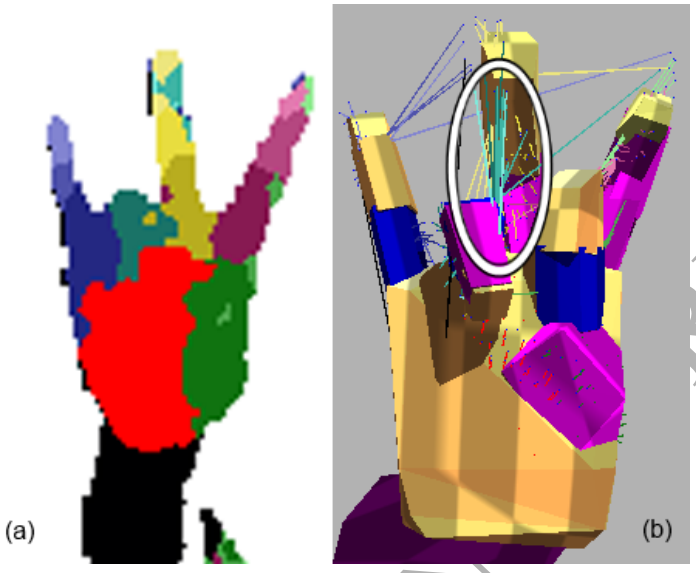


Figure 5: Constraint driven optimisation. (a) The hand's depth following segmentation of the Random Decision Forest. (b) The kinematic hand model after optimization using constraints. The ellipse highlights the finger being correctly estimated, despite erroneous constraints that are overpowered through optimisation.

Gauss-Seidel solver as described earlier. Optimisation iteratively applies impulse forces to reduce the error in equation 6 from all point to surface constraints and kinematic constraints.

Constraints which are incorrect are overpowered by the global consensus. This can be seen in Figure 5 where the middle finger tip is correctly identified, shown as yellow in 5(a), which pulls the model correctly into position, despite the misclassification of the middle section. Had the joint been determined using mean-shift or regression, it is likely the fingers would be self-intersecting, hence providing an invalid pose estimate.

6. Residual Error Regression

The following section discusses our Residual Error Regressor (RER) which aims to learn to recover from failures in optimisation due to local minima and residual errors in model fitting. One source of residual error comes from discrepancies between the user hand shape and that of a generic model. This residual error can be seen in Figure 6, which shows poor fitting around the palm and some parts of the fingers. The additional optimisation of hand proportions is not tractable for

real-time performance as this would significantly increase the degrees of freedom during optimisation. Instead, we propose the use of discriminative means to provide correction offsets that refine the pose estimate. Using local features around each joint, a cascaded linear regression is trained. Training samples are generated using pose estimates, computed using the CDO with perturbed initialisation.

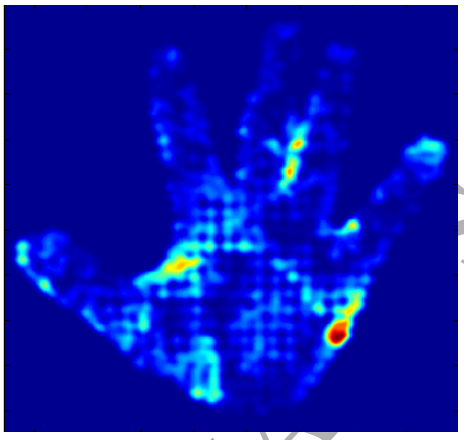


Figure 6: Residual constraint error from using a model with rigid components when converged using an ideal segmentation of depth. This error is synonymous with the use of a general hand model, and inter-person variation.

The pose estimated via CDO serves as the initial estimate and is defined as $\mathcal{H}^0$. The centroid of each body is defined as $\mathbf{c}_b$ and represents the joint during refinement. Information regarding the surface gradient and contour is captured using features that use pairwise offset $\mathbf{u}$ and $\mathbf{v}$ derived from 3 given as follows:

$$F_{\mathbf{u},\mathbf{v}}\left(\mathbf{I}, \mathbf{c}_b\right) = \mathbf{I}\left(\mathbf{x}_b + \frac{\mathbf{u}}{z}\right) - \mathbf{I}\left(\mathbf{x}_b + \frac{\mathbf{v}}{z}\right) \tag{7}$$

where z is the depth of the joint $\mathbf{c}_b$ rather than image depth and $\mathbf{x}_b$ is its projected image point. Features are sampled randomly around each part $\mathbf{c}_b \in \mathcal{H}$, providing local context about the depth surrounding its location. f features are sampled randomly around each joint, with the difference in depth between offsets being constrained to the range of the hand's depth.

The features for the hand's joints are then concatenated forming a high dimensional feature vector representing $\mathcal{H}$. Due to the sparse nature of the features, PCA is performed. Those dimensions containing 95% variance over our training set are preserved. We define the function $\phi(\mathbf{I}, \mathcal{H})$ to denote the PCA projected features.

Each tier of the cascade $k = (1, ..., K)$ estimates a joint update in pose $\Delta\mathcal{H}^k$, which aims to converge to the true joint positions. The following describes the application of the offset vector computed by each independent regressor $R(\phi)$

$$\begin{aligned} \Delta\mathcal{H}^k &= R^k(\phi(\mathbf{I}, \mathcal{H}^{k-1})) \\ \mathcal{H}^k &= \mathcal{H}^{k-1} + \Delta\mathcal{H}^k \end{aligned} \tag{8}$$

The pose refinement $\Delta\mathcal{H}^k$ predicted by $R(\phi)$ uses a linear projection of the high dimensional features ϕ. Training for each tier of regression is performed through minimisation of equation 9, solving for the projection matrix $\mathbf{R}_k$ and bias term $\mathbf{b}_k$ in the following:

$$\underset{\mathbf{I}, \mathcal{H}_k^i}{\arg\min} \sum_{\mathbf{I}^i} \sum_{\mathcal{H}_k^i} \left\| \Delta\mathcal{H}_k^i - \mathbf{W}_t \phi_k^i - \mathbf{b}_k \right\|^2 \tag{9}$$

Cascaded regression was previously used in facial landmark estimation by Xiong [41] where sampling during training used Gaussian noise added to landmark locations. However, this assumes a normal distribution. Instead, direct sampling of the CDO is performed. As rigid body simulation is deterministic, a random perturbation is added to the initial tracking state, and the CDO can then be used to generate many training samples.

For such a non-linear problem, cascaded regression accuracy can be improved by limiting the offset distance used in testing. Rather than applying each regression tier for the full residual error $\Delta\mathcal{H}$, a fractional update is used. Each regressor evaluates half $\Delta\mathcal{H}$. This prevents the first regressor attempting to model the complete error. This increases the number of steps required but reduces instability.

7. Preparing Training Data

The RDF learns discriminative features which allow the region labels to be determined but training of the forest requires segmented hand images. These regions were initially found using nearest neighbour assignment of each of the hand pixels to its nearest joint in 3d space. However, this was found to be unreliable as the assignment did not consider the boundaries between fingers and lacked an understanding of occlusion. This can be seen in Figure 7b with noise present around each of fingers. A better solution was to use the hand model $\mathcal{H}$ to label the pixels as belonging to each part of the hand. For each example pose, the model $\mathcal{H}$ pose was estimated through optimisation to match the ground truth, constraining

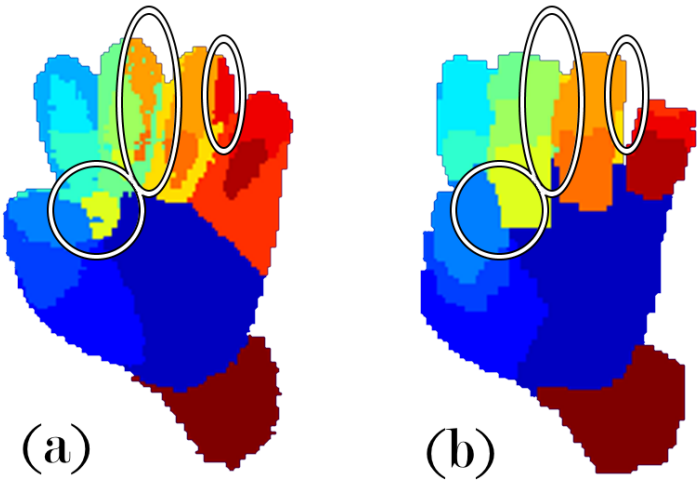


Figure 7: Figure demonstrating labelling of the hand using ground truth landmarks. (a) An example pose labelled by assigning each pixel the index of its closest joint. (b) using a ray test through our model, which is constrained using the ground truth

each joint position to its corresponding landmark. Each point in the depth image was then back projected using $\mathbf{K}$ to its 3D position. A ray was then traced from the camera to each point and beyond, thus the correct hand part could be identified at the intersection between ray and hand model. This provided robust labelling and offers two distinct benefits, firstly the segmentation is occlusion aware, recognising closely interacting fingers, and secondly the segmentation is accurate to the model, with consistent labelling at the boundary of neighbouring joints. This improved method of label assignment can be seen in Figure 7b under each of the ellipses. In addition to labelling the hand, the forearm was also identified to improve the localisation of the palm when δ_d is overestimated.

Given that the model is general it is likely that some rays may miss intersection with it's bodies, in which case they are not used in training. This did not cause issues during training as the model is sufficiently accurate, with only some boundary examples not being used. The RDF also generalises to pixels of similar neighbourhood appearance, accounting for those boundaries missed in training.

Table 1: The number of training and test examples seen in the NYU and ICVL datasets. The ICVL data covers a range of users while NYU has limited users but covers a larger depth range with increased noise.

Dataset:	N Training	N Test	N Train Users	N Test Users	Label Type	Range
NYU [2]	72,000	8200	1	2 (1 Seen)	2D	0.5m - 1.2m
IVCL [4]	330,000	1600	12	2 (2 Unseen)	3D	0.2m -0.5m

8. Experiments

The following section details the experimental validation of parameters used and the evaluation of the pose estimation. The evaluation includes qualitative and quantitative evaluation comparing our approach to existing state of the art approaches. A comparison of runtime performance is also conducted, highlighting the highly efficient performance using only a single threaded CPU.

The data used in the following experiments are from the ICVL [4], Dextar1 [21] and NYU [2] datasets and were used in the evaluation of Latent Regression Forests, model fitting and CNN approaches. Information regarding the size of these datasets can be seen in Table 1. An implementation of Keskin [25] is also used to demonstrate the improvement of the combined model over forest segmentation alone. A qualitative evaluation is then conducted against the approach of Melax [5].

8.1. Parameter Selection

The following section discusses the parameters used for pose estimation and optimisation over unseen examples from the ICVL [4] dataset. A validation set of 10000 images were excluded from training for optimisation. The RDF is configured using similar parameters as discussed in [24, 25, 9] Those parameters are summarised in the following. During training 2000 different splitting criteria were evaluated at each node and a random sample of 1000 pixels was used from each image. The ideal forest depth was d = 20, with three trees having been used. Deeper forests increase evaluation time, with a limited gain in performance. The addition of model fitting and the constraint of real-time processing limited the forests classification to 1000 points. The optimal maximal radius of was found to be 60 pixel meters, allowing the features to sample across the hands width. The remaining parameters optimised in this section are unique to the RER and are selected as to minimise the accumulated mean joint error over the validation examples.

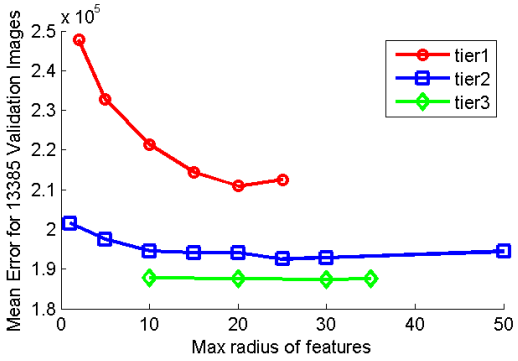


Figure 8: Parameter tuning of maximum feature radius. The reduction in error based on the number of tiers can also be seen.

Maximal Feature Radius

The maximum radius of the features used in RER can be adjusted. The results for adjusting the radius for each tier k of regression can be seen in Figure 8. The optimal feature radius is 20 for each tier and is more stable at each subsequent tier. The number of tiers also impacts performance and their convergence rate can be seen to lessen after the second tier, hence we only use two tiers at runtime.

This decrease in radius over that used by the forest is likely attributed to the local nature of the joint updates. The RER is intended to recover inaccuracies in the model fitting rather than estimate the entirety of the hand. This limited radius also has an impact on the number of features that are sampled. With a smaller radius, fewer features should be required to maintain the sampling density.

Number of Features

The cascaded regression model uses a number of features captured at each joint. The graph shown in Figure 9 demonstrates the change in performance versus the number of features captured at each joint. It can be seen that performance plateaus at 500 features. The reduced number of features over those needed in the forest is attributed to the proximity of the features.

8.2. NYU Dataset Evaluation

The evaluation dataset of NYU is labelled using a optimised hand model. There are 36 landmarks, that corresponds to a number of locations over there model. The results of NYU's CNN are provided for a subset of these positions. This subset differs from those which correspond to the positions available from the CDO

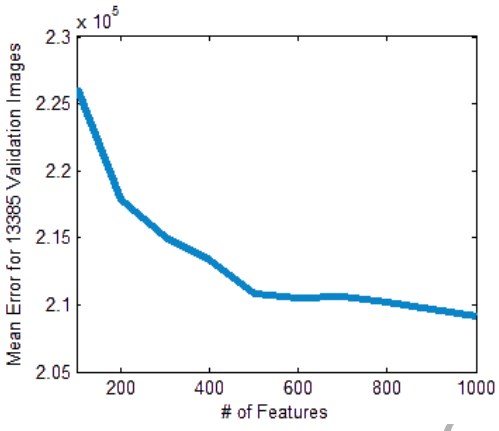


Figure 9: The impact on performance when changing the number of features. The performance begins to plateau when the number of features increases above 500.

model. This does not allow direct comparison against all of the joints, but rather the overlapping subset. This includes the centre of the hand, three points on the thumb and a point at the end of each finger (PALM_3, TH_KNU3_A, TH_KNU3_B, TH_KNU2_B, F1_KNU3_A, F2_KNU3_A, F3_KNU3_A, F4_KNU3_A).

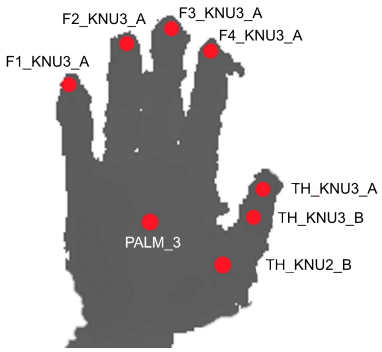


Figure 10: Joints used in the evaluation comparing CDO against NYU's CNN [2] and Mean Shift.

The result of Tompson et al CNN approach do not produce an estimate for depth, due to their approach regressing to an image location rather than a 3D position. In order to compute error in 3D, the depth for each of their joints is inferred from the depth of the hand. This is also the case for regressed locations that do not reside on the hand (e.g. missing depth). The graph presented in figure 13 demonstrates the performance of CDO without RER, compared against NYU's

CNN [2] and an implementation of Mean Shift[25]. RER is excluded as regression methods can overfit to the user when seen in training and test.

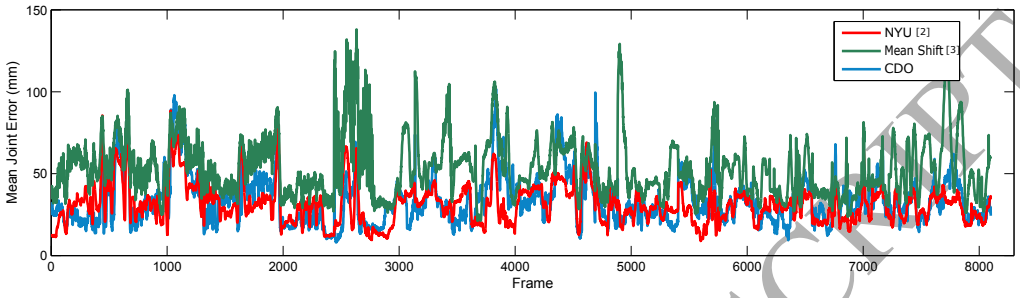


Figure 11: The joint error over NYU's test set, demonstrating similar performance to the CNN approach. The graph also shows the result of a mean shift method, which shows a large error demonstrating the improvement in using CDO.

The performance of the CDO is considerably improved over the Mean Shift method. This demonstrates the importance of explicitly modelling the Kinematic constraints. The variance between consecutive frames is also reduced, due to the incorporation of temporal modelling. The performance is comparable with that of NYU's approach [2], which performs direct regression but lacks information regarding the depth of the joints, which the CDO provides.

8.3. Dexter1 Dataset Evaluation

The Dexter1 dataset is multiple viewpoint dataset that includes captures from a ToF and structured light camera. The dataset is annotated with landmark position for each fingertip and the palm centre. There are seven sequences presented in the dataset which demonstrate several actions. Since the release of the dataset, a number of approaches have used it as a benchmark, which allows us to perform a direct comparison against the methods proposed by Sharp [8], Sridhar [21] and Tagliasacchi [7]. The graph in Figure 12 demonstrates the performance of the proposed method when compared against other model-based methods.

We also demonstrate our per frame performance over the course of the best and worst sequence in Figure 13. The worst performance is seen in sequence Adbadd which demonstrates abduction-adduction of the hand. It can be seen that there is a failure in tracking which we attribute to global motion that is present during that part of the sequence. For the remainder of the sequence, performance is comparable with the best performing sequence, demonstrating the model's ability to recover quickly from failure.

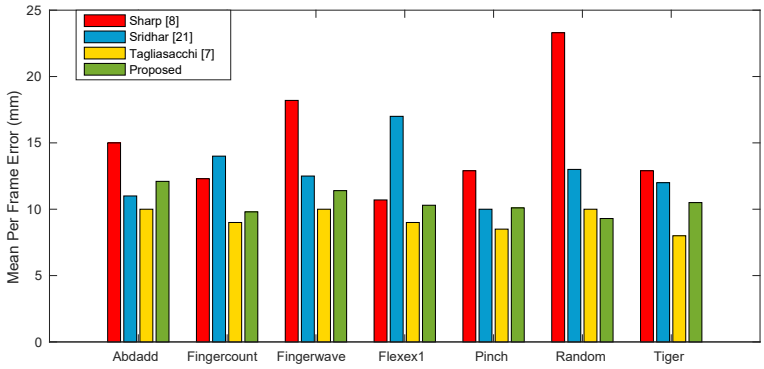


Figure 12: We quantitatively evaluate our algorithm using the Dexter1 dataset. The graph presents the mean per frame error for each sequence against the methods of Sharp [8], Sridhar [21] and Tagliasacchi [7]. Sridhar is the only approach of these to use the multiple RGB viewpoints. We show comparative results with Tagliasacchi which is a current state-of-the-art method on this dataset.

8.4. ICVL Dataset Evaluation

Ground truth labels were provided using automatic means, using the approach of Melax. These landmarks were then manually corrected, with entirely erroneous frames rejected. There are over 20,000 labelled frames captured across 12 users which are synthetically rotated, providing 330,000 training examples. There are two evaluation sequences of unseen users exhibiting challenging interaction poses and transitions. Frames were sampled at every third frame, both in the training and test sequences. This increases the difficulty for model-based approaches such as that proposed and Melax's as there is a larger transition between frames. There is also noise present demonstrated in figure 14 where it is quantified with a naive measure by calculating the number of ground truth labelled joints that are positioned outside of hand's contour on a per frame basis. There is also error observed for labelled points inside the contour shown in figure 15. Consistent errors in landmark accuracy are also likely to be of benefit to direct regression methods, which can learn the errors present in the training data. During evaluation, this can lead to inaccurate localisation which is consistent with the ground truth.

We evaluate the accuracy of our approach with the state of the art method of Tang [4] which performs the regression of joint locations through a hierarchical tree structure. The metrics used are the mean joint error for the hand for each frame, and the cumulative mean error as seen in Figure 16. The approach exhibits considerably less inter-frame noise, which is again attributed to the use of a model

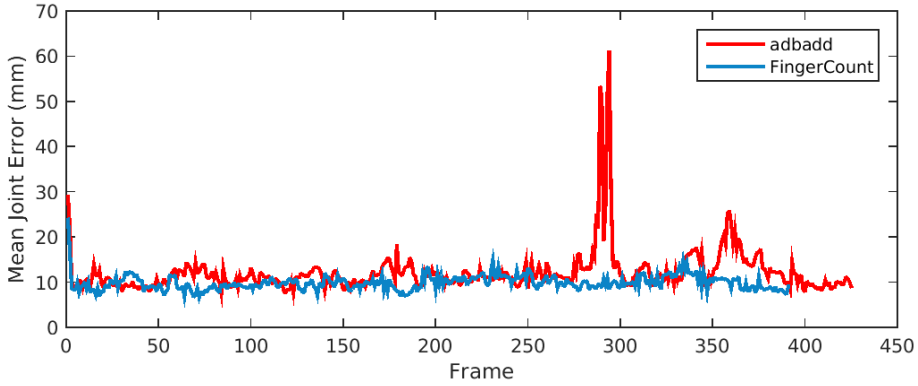


Figure 13: This graph presents the joint error for each frame for two sequences of the Dexter1 dataset. These two sequences illustrate the best and worst performance. A loss of tracking can be seen around frame 240 which is recovered from quickly due to the method being a hybrid approach, performing reinitialisation.

which is temporally coherent, and improves on simply smoothing the resulting detection. This offers a more realistic users experience allowing fine control with less jitter, which is important for a natural interface.

The cumulative mean error shows an improvement in accuracy across sequence 1 with a 24% improvement in joint localisation which we attribute to the use of cascaded regression. This is confirmed when comparing performance with and without the residual error regression. The graph in Figure 17 demonstrates the impact of residual error regression and shows both the inter-frame and cumulative error over sequence 1. The reduction in cumulative error demonstrates consistent improvement on the unseen data, on average reducing the mean joint error by 4.94 mm across the sequence which is a 32 % decrease in error. Closer inspection of frame wise error in Sequence 1 (Figure 17) shows there are limited instances where regression deteriorates performance.

The second sequence in Figure 18 shows similar performance to that of Tangs. We attribute this to the presence of faster gestures, which are harder to track due to the temporal sampling.

8.5. Qualitative Study

The following discusses the qualitative comparison with the approach of Melax [5] which is the first approach to utilise RBS. The inclusion of heuristics allows Melax's approach to track a range of poses. Tracking is fast with real-time frame rates and suffers little lag. However in instances of failure, the

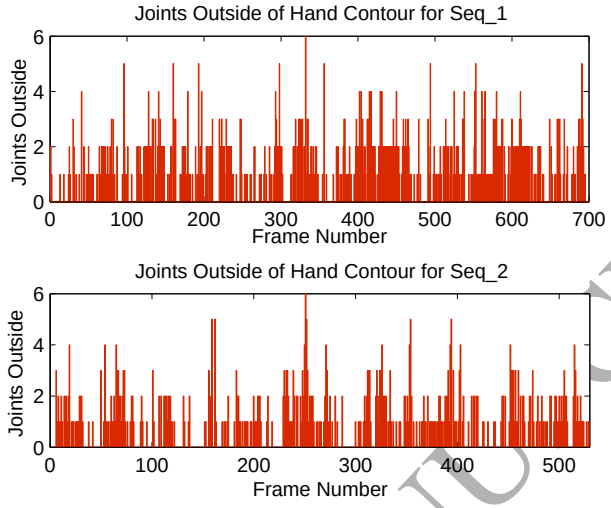
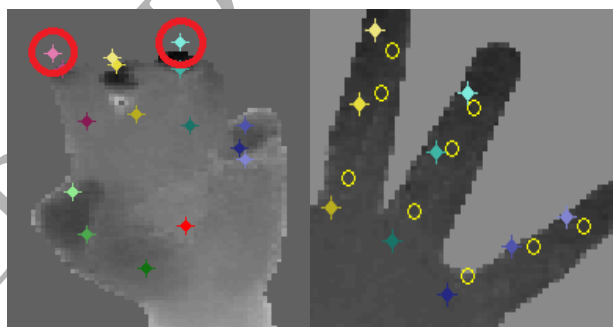


Figure 14: Graph showing labelling error in ground truth (GT) of test data. Measured using joints outside of the hand contour in sequence 1 and sequence 2. This error is due to the test sequence having been labelled using automatic means.



(a) (b)

Figure 15: The coloured markers show the ground truth generated using autonomous means. (a) Invalid GT points that reside outside of the hand's contour are highlighted. (b) Inaccurate GT points at are inside of the hand's contour. Yellow circles represent the result from the CDO approach, showing better then GT performance.

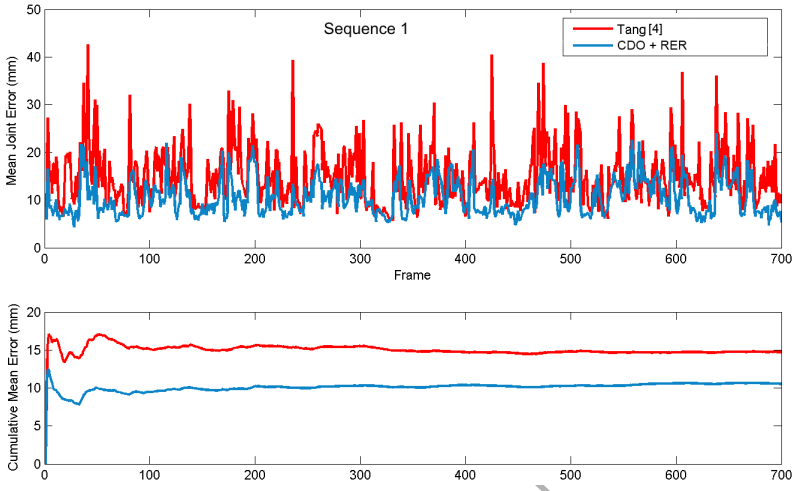


Figure 16: Evaluation over sequence 1 from ICVL comparing the per frame mean joint error, and it's cumulative moving average.

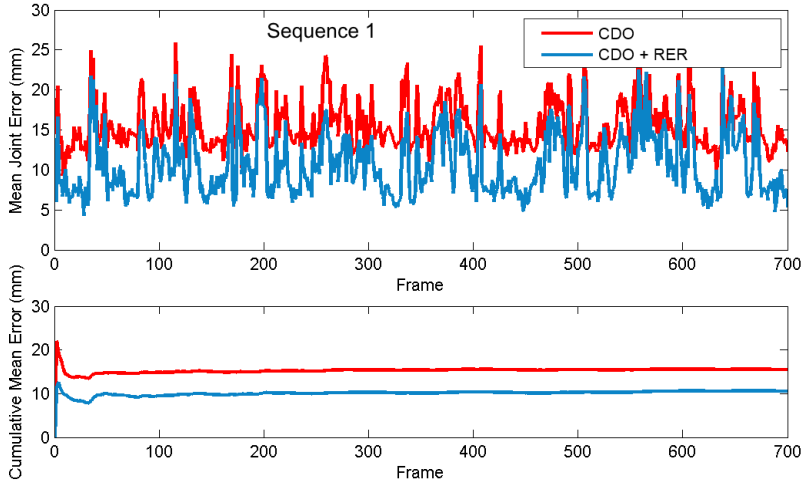


Figure 17: Error of hand pose estimation with and without the use of cascaded linear regression.

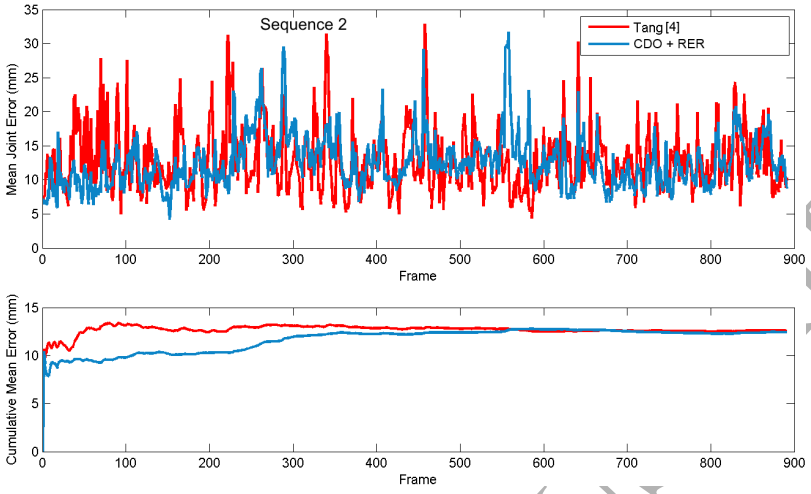


Figure 18: Evaluation over sequence 2 from ICVL comparing the per frame mean joint error, and it's cumulative moving average.

model becomes trapped in invalid poses, resulting in unrecoverable tracking failure. One such example can be seen in figure 19b while below (fig19d) shows our model is less susceptible to local minima. For Melax's approach, recovery of such errors is performed using finger detection, requiring the user to form an initialisation pose of splayed fingers. The CDO approach offers less intrusive reinitialisation as the RDF provides detection continually, allowing seamless recovery from tracking failure. The integration of prior data in the approach also allows application specific gestures to be trained.

A number of poses that result from the CDO method are illustrated in figure 20 and 21. The images show our robust localisation against the sequences provided in the ICVL dataset as well a live version of the system, which includes poses of increased difficulty. Many failures can be seen for the approach of Melax, while the CDO provides good estimation.

The combination of the discussed approaches allows rapid optimisation of the hand model. Table 2 compares the run time performance using a single CPU thread against existing approaches. The CDO performs with real time performance with accuracy greater than the approaches proposed by Melax [5] and Tang [4]. Sharp's [8] approach requires a GPU to perform in real time and as such limits its range of applications.

Figure 22 shows the steps taken during error regression in isolation, highlighting its ability to converge on the true joint location following CDO. The refined

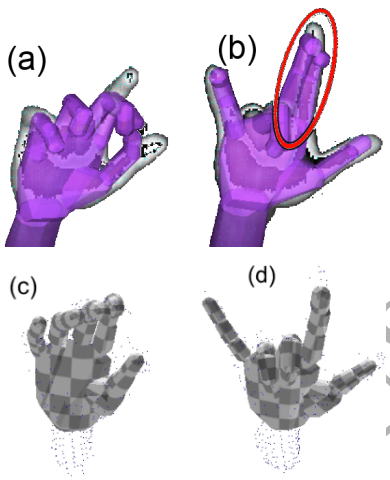


Figure 19: Example showing failure instances of Melax's approach. Comparing Melax's method[5](a,b) and CDO(c,d), the model in (a) fails to fully converge with the depth. The middle finger is incorrectly attached to the index finger forming a local minima (b). The proposed model optimisation over a similar sequence (c,d) does not suffer from local minima.

Method	**Sharp** [8]	**Keskin** [3]	**Xu** [26]	**Melax** [5]	**Tang** [4]	**CDO**
Device(# threads)	GPU	CPU(1)	CPU(1)	**CPU(1)**	**CPU(1)**	**CPU(1)**
Frame Rate(fps)	30	8	12	**60**	**63**	**40**

Table 2: Table demonstrating the real time performance of contending approaches. The device used for computation and the frame rate of each method is quoted. Those highlighted demonstrate real time >30 fps performance using a single threaded implementation.

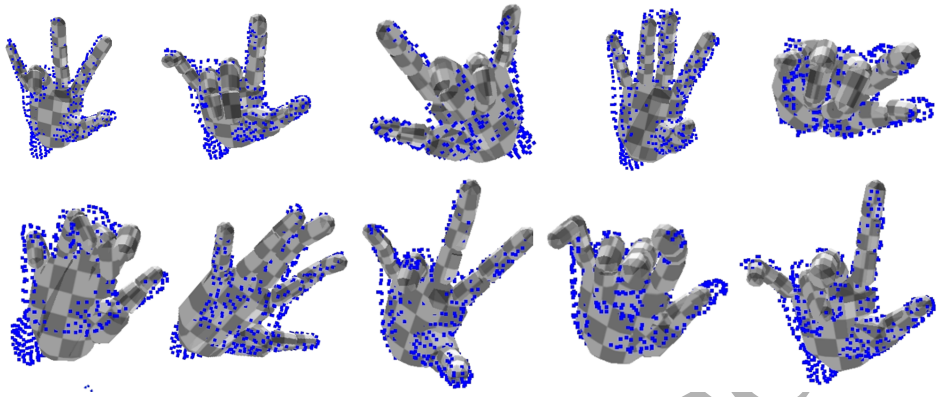


Figure 20: Qualitative Evaluation: examples of the combined descent method fitting to point cloud data. The kinematic model prevents self-intersection between fingers and provides realistic results.

locations provide improved accuracy against model discrepancies.

9. Conclusions

This work presented an approach for hand pose estimation that utilised a combination of techniques, seeking to reduce their mutual failure cases. Through training of a RDF, a region based correspondence can be learned from previously observed hand data. This segmentation provides the assignment for point to surface constraints, allowing a realistic hand model to be fitted to the observation data. Operating as a ICP based method, the minimisation is fast to converge with the observed point cloud. The model provides structural information of the hand, enforcing kinematic limitations and hierarchical constraints, ensuring only natural poses are evaluated. Self-intersection is prevented through the application of collision based constraints, which serve to drive intersecting bodies apart. The model fitting is conducted using a Rigid Body Simulation in a Newtonian formalisation, realistically modelling changing poses. Minimisation is initialised using the previous frame's estimate and motion state to incorporate temporal information. In many approaches, such initialisation would lead to model fitting becoming trapped in local minima. To recover from such failures, approaches depended on manual or fingertip reinitialisation. However, the use of segmentation provides a continuous detection at each frame, allowing the tracking to successfully recover the hand from a range of challenging poses. This allows graceful recovery of tracking, which is important for natural interaction.

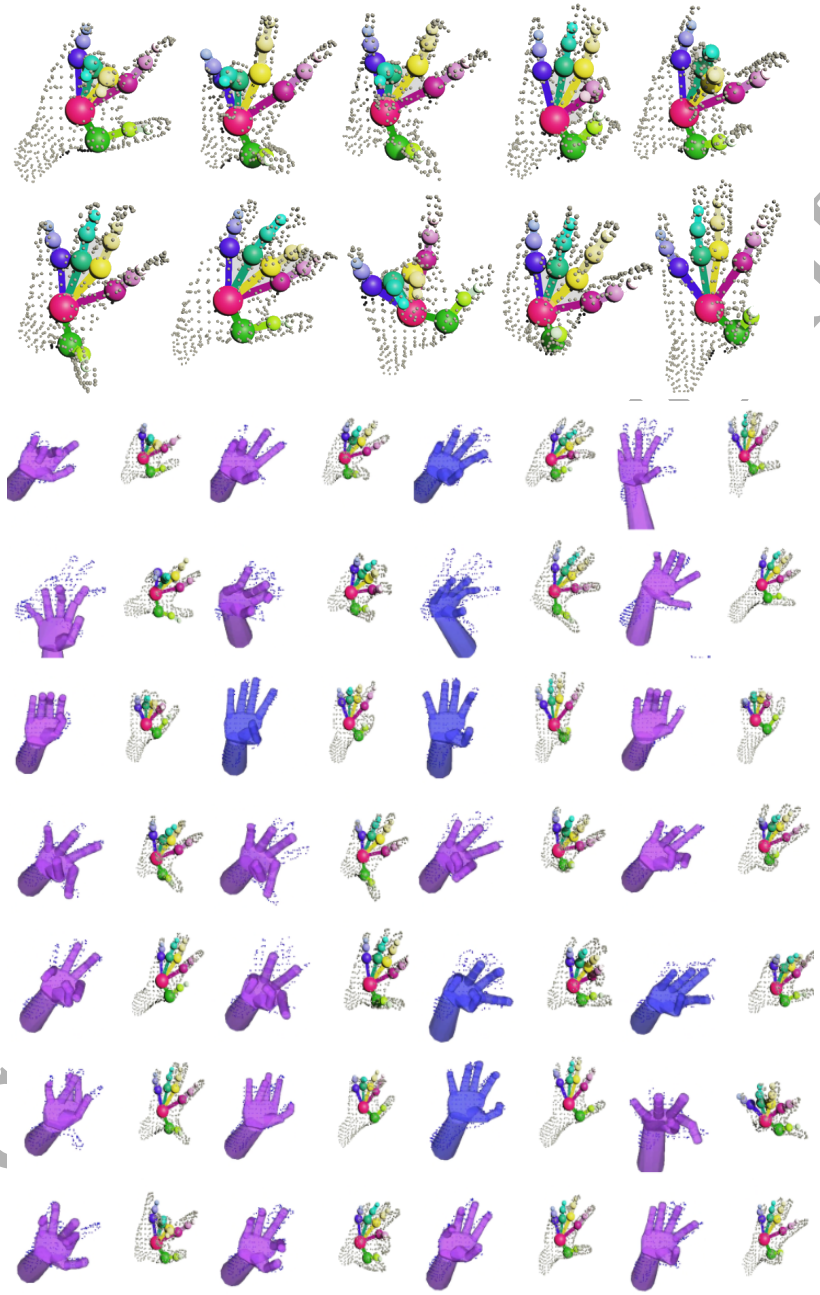


Figure 21: The result of model fitting and joint refinement for several pose examples, with comparison between Melax (Blue/Purple model) and the refined model over Sequence 1 from ICVL dataset.

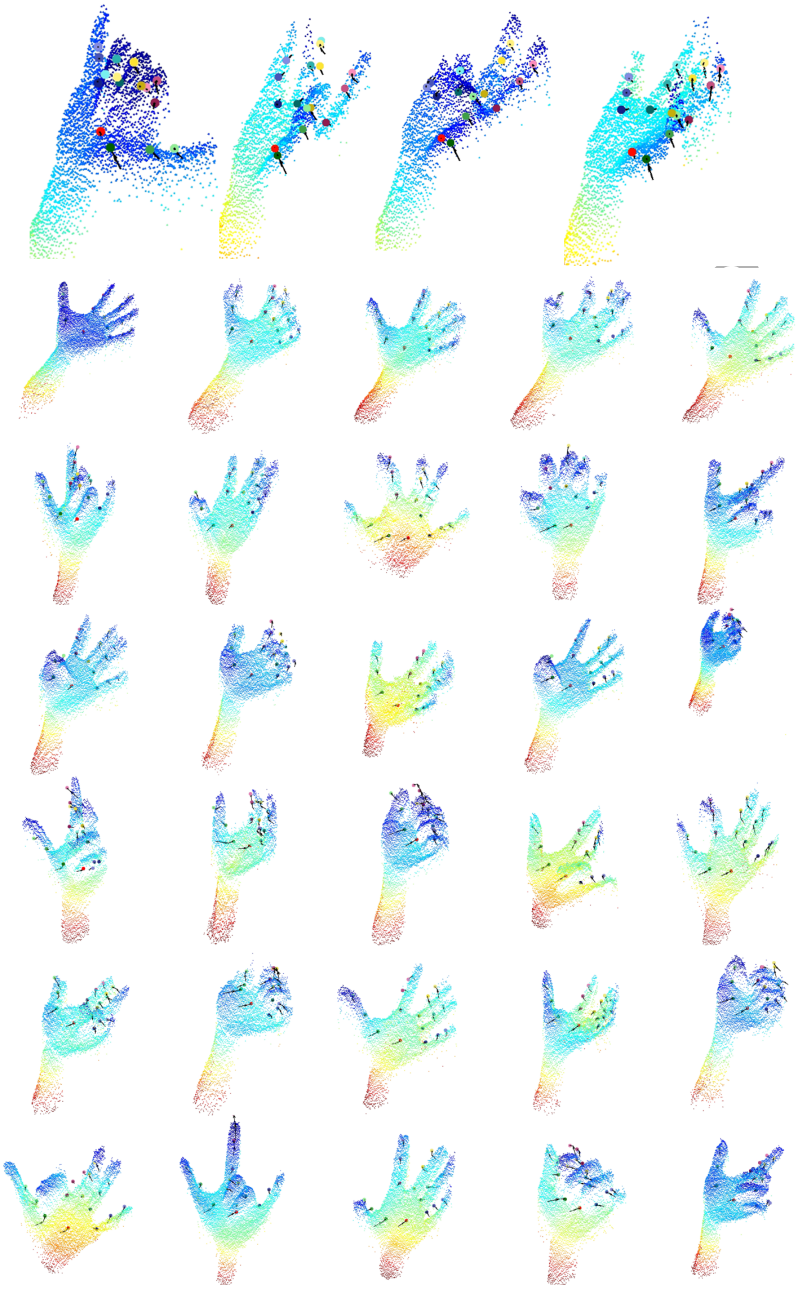


Figure 22: Linear regression provides an update vector represented by the arrows at each joint. The final proposed joint are labelled as coloured points.

In fitting a generic hand model, accuracy is typically limited, due to variation in the shape and proportions of the hand. For this reason, we proposed the use of linear regression that samples from the residual error between the proposed model configuration and the hand's appearance. A correction vector, projected from a high dimensional feature space is then applied iteratively, refining the optimisation's resulting pose, with a mean error of approximately 10mm across the hand's joints.

The presented system provides state of the art performance over three challenging sequences, demonstrating the ability to generalise to new users. Providing real-time tracking with limited computing resources demonstrates potential use in embedded applications and general computing.

For future work, we seek to improve the hand's initial segmentation, allowing robust use for a wider range of global poses. Machine learning will also be used to partition the hand and forearm, without the need for the user to extend their hand forward. We also wish to explore other learning techniques, that could provide improved partitioning of the hand. One such method would be Vitruvian manifold learning, which regresses to a surface location, allowing a dense surface correspondence to be formulated.

10. Acknowledgments

This work was supported by a EPSRC studentship and the EPSRC project, Learning to Recognise Dynamic Visual Content from Broadcast Footage (EP/I011811/1)

References

[1] P. Krejov, A. Gilbert, R. Bowden, A Multitouchless Interface, Computer Graphics and Applications, IEEE.

[2] J. Tompson, M. Stein, Y. Lecun, K. Perlin, O. Database, Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks, in: SIGGRAPH, 2014. doi:10.1145/2629500.

[3] C. Keskin, F. Kraç, Y. Kara, L. Akarun, Hand pose estimation and hand shape classification using multi-layered randomized decision forests, Computer VisionECCV 2012 (2012) 852–863.

[4] D. Tang, H. Chang, A. Tejani, T. Kim, Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture, in: CVPR 2014, 2014.

[5] S. Melax, L. Keselman, S. Orsten, Dynamics based 3D skeletal hand tracking, in: Graphics Interface (ACM), ACM Press, New York, New York, USA, 2013. doi:10.1145/2448196.2448232.

[6] I. Oikonomidis, N. Kyriazis, A. Argyros, Efficient Model-based 3D Tracking of Hand Articulations using Kinect, British Machine Vision Conference (BMVC)doi:10.5244/C.25.101.

[7] A. Tagliasacchi, M. Schröder, A. Tkach, Robust Articulated-ICP for Real-Time Hand Tracking, Computer Graphics Forum 34 (5).

[8] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freeman, P. Kohli, E. Krupka, A. Fitzgibbon, S. Izadi, Accurate, robust, and flexible real-time hand tracking, in: Human Factors in Computing Systems (CHI), Seoul Korea, 2015.

[9] P. Krejov, A. Gilbert, R. Bowden, Combining Discriminative and Model Based Approaches for Hand Pose Estimation, in: Automatic Face and Gesture Recognition (FG), 2015 11th IEEE, 2015, pp. 1–7.

[10] J. Rehg, T. Kanade, Visual tracking of high dof articulated structures: an application to human hand tracking, European Conference on Computer Vision (ECCV) (May) (1994) 35–46.

[11] B. Stenger, P. Mendonca, R. Cipolla, Model-based 3D tracking of an articulated hand, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Vol. 2, IEEE Comput. Soc, 2001, pp. II–310–II–315. doi:10.1109/CVPR.2001.990976.

[12] B. Stenger, A. Thayananthan, P. H. S. Torr, R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter., IEEE transactions on pattern analysis and machine intelligence 28 (9) (2006) 1372–84. doi:10.1109/TPAMI.2006.189.

[13] M. de La Gorce, D. J. Fleet, N. Paragios, Model-Based 3D Hand Pose Estimation from Monocular Video., IEEE transactions on pattern analysis and machine intelligence (2011) 1–15doi:10.1109/TPAMI.2011.33.

[14] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, X. Twombly, Vision-based hand pose estimation: A review, Computer Vision and Image Understanding 108 (1-2) (2007) 52–73. doi:10.1016/j.cviu.2006.10.012.

[15] A. Aristidou, J. Lasenby, Motion Capture with Constrained Inverse Kinematics for Real-Time Hand Tracking, Communications, Control and Signal Processing (March) (2010) 3–5.

[16] R. Y. Wang, J. Popović, Real-time hand-tracking with a color glove, in: Transactions on Graphics (ACM), 2009. doi:10.1145/1531326.1531369.

[17] L. Dipietro, A. Sabatini, P. Dario, A survey of glove-based systems and their applications, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviewsdoi:10.1109/TSMCC.2008.923862.

[18] H. Hamer, K. Schindler, E. Koller-Meier, L. Van Gool, Tracking a hand manipulating an object, Computer Vision.

[19] I. Oikonomidis, M. Lourakis, A. Argyros, Evolutionary Quasi-random Search for Hand Articulations Tracking, in: Computer Vision and Pattern Recognition (CVPR), 2014. doi:10.1109/CVPR.2014.437.

[20] I. Oikonomidis, N. Kyriazis, A. a. Argyros, Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints, International Conference on Computer Vision (ICCV)doi:10.1109/ICCV.2011.6126483.

[21] S. Sridhar, A. Oulasvirta, C. Theobalt, Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data, 2013 IEEE International Conference on Computer Vision (2013) 2456–2463doi:10.1109/ICCV.2013.305.

[22] C. Qian, X. Sun, Y. Wei, X. Tang, J. Sun, Realtime and Robust Hand Tracking from Depth, 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 1106–1113doi:10.1109/CVPR.2014.145.

[23] T. Schmidt, R. Newcombe, D. Fox, DART: Dense Articulated Real-Time Tracking, Robotics: Science and Systems (1).

[24] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: CVPR 2011, IEEE, 2011, pp. 1297–1304. doi:10.1109/CVPR.2011.5995316.

[25] C. Keskin, F. Kirac, Y. E. Kara, L. Akarun, Real time hand pose estimation using depth sensors, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011, pp. 1228–1234. doi:10.1109/ICCVW.2011.6130391.

[26] C. Xu, L. Cheng, Efficient Hand Pose Estimation from a Single Depth Image, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3456–3462.

[27] D. Tang, T. Yu, T. Kim, Real-time Articulated Hand Pose Estimation using Semi-supervised Transductive Regression Forests, in: ICCV, 2013.

[28] J. Shotton, T. Sharp, A. Kipman, Real-time human pose recognition in parts from single depth images, Communications of the ....

[29] N. Neverova, C. Wolf, G. Taylor, F. Nebout, Hand segmentation with structured convolutional learning, Asian Conference on Computer Vision (ACCV).

[30] X. Sun, Y. Wei, S. Liang, X. Tang, J. Sun, Cascaded Hand Pose Regression.

[31] P. Doliotis, V. Athitsos, D. Kosmopoulos, S. Perantonis, Hand shape and 3D pose estimation using depth data from a single cluttered frame, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7431 LNCS (PART 1) (2012) 148–158. doi:10.1007/978-3-642-33179-4{_}15.

[32] P. Krejov, R. Bowden, Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima, in: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), IEEE, 2013, pp. 1–7. doi:10.1109/FG.2013.6553778.

[33] D. Thalmann, H. Liang, J. Yuan, 3D fingertip and palm tracking in depth image sequences, Proceedings of the 20th ACM international conference on Multimedia - MM '12 (2012) 785doi:10.1145/2393347.2396312.

[34] H. Liang, J. Yuan, D. Thalmann, Z. Zhang, Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization, The Visual Computer 29 (6-8) (2013) 837–848. doi:10.1007/s00371-013-0822-4.

[35] Bulletphysics.org, Real-Time Physics Simulation.

[36] E. Gilbert, D. Johnson, S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space, IEEE Journal on Robotics and Automation 4 (2) (1988) 193–203. doi:10.1109/56.2083.

[37] I. Albrecht, J. Haber, H.-p. Seidel, Construction and Animation of Anatomically Based Human Hand Models, SIGGRAPH.

[38] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, A. Fitzgibbon, User-Specific Hand Modeling from Monocular Depth Sequences, 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 644–651doi:10.1109/CVPR.2014.88.

[39] A. Buryanov, V. Kotiuk, Proportions of Hand Segments, International Journal of Morphology 28 (3) (2010) 755–758.

[40] J. Lin, Y. Wu, T. S. Huang, Modeling the Constraints of Human Hand Motion, Constraints (2000) 121–126 167doi:10.1109/HUMO.2000.897381.

[41] X. Xiong, F. De la Torre, Supervised Descent Method and Its Applications to Face Alignment, 2013 IEEE Conference on Computer Vision and Pattern Recognition (2013) 532–539doi:10.1109/CVPR.2013.75.