Annotated Free-hand Sketches for Video Retrieval using Object Semantics and Motion

Rui Hu, Stuart James and John Collomosse

Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey, Guildford, Surrey, U.K.
{R.Hu | S.James | J.Collomosse}@surrey.ac.uk

Abstract. We present a novel video retrieval system that accepts annotated free-hand sketches as queries. Existing sketch based video retrieval (SBVR) systems enable the appearance and movements of objects to be searched naturally through pictorial representations. Whilst visually expressive, such systems present an imprecise vehicle for conveying the semantics (e.g. object types) within a scene. Our contribution is to fuse the semantic richness of text with the expressivity of sketch, to create a hybrid 'semantic sketch' based video retrieval system. Trajectory extraction and clustering are applied to pre-process each clip into a video object representation that we augment with object classification and colour information. The result is a system capable of searching videos based on the desired colour, motion path, and semantic labels of the objects present. We evaluate the performance of our system over the TSF dataset of broadcast sports footage.

1 Introduction

Text keywords are the dominant query mechanism for multimedia search, due to their expressivity and compactness in specifying the semantic content (e.g. car, horse) desired within a scene. However, keywords lack the descriptive power to concisely and accurately convey the visual appearance, position and motion of objects. Querying by Visual Example (QVE) offers a solution, yet most video QVE techniques require a photo-real query (e.g. image [33], or video [5]) and so are unsuitable in cases where exemplar footage is absent. Free-hand sketch is a complementary query mechanism for specifying the appearance and motion of multimedia assets, and has recently been applied to video retrieval [8, 15]. However the throw-away act of sketch, combined with limited artistic skill of non-expert users, can make unambiguous depiction of objects challenging. Such ambiguity limits the size and diversity of the dataset that can be queried purely by pictorial means. The contribution of this paper is to fuse the orthogonal query methods of *sketch* and *text* — for the first time presenting a QVE system for searching video collections using *textually annotated sketch* queries.

Our system accepts a colour free-hand sketched query annotated with text labels indicating object classification (semantics), and motion cues (arrows) that indicate the approximate trajectory of the desired object. We focus upon these cues to assess relevance, following recent studies [9, 8] that observe users to draw upon their *episodic memory* during sketch recall — resulting in sketches exhibiting low spatial and temporal fidelity [34]. Users typically recall the names of a few salient objects in a scene, and their approximate trajectories, rather than their detailed appearance (e.g. shape). Object appearance tends to be depicted coarsely, using a limited yet approximately correct colour palette. Therefore, although users naturally depict an object's shape within a sketch, we *do not currently use shape information* to influence the type of object to retrieve. Rather, our contribution is to combine spatio-temporal position information in the sketch with colour, and the semantic tags associated with the object to create a more scalable solution than that offered by shape alone [8, 15].

We represent video as a set of video objects, identified during video ingestion by motion segmentation based on an unsupervised clustering of sparse SIFT feature tracks. A super-pixel representation of video frames is used to aggregate colour information local to each video object. An object class distribution is also computed local to each video object, based on a per-pixel labelling of frames via a random-forest classifier. Thus each spatio-temporal video object is accompanied by colour, semantic and motion trajectory data. At query-time sketched trajectories are matched to the trajectories of video objects using an adapted Levenshtein (edit) distance measure, alongside a measurement of similarity between the colour and semantic distributions of the query and candidate objects.

We describe the extraction and matching of the video object representation in Sec. 3 and 4 respectively, evaluating over a subset of the public TSF dataset in Sec.5.

1.1 Related work

Sketch based retrieval (SBR) of visual media dates back to the nineties, and the development of image retrieval systems where queries comprised sketched blobs of colour and texture [13, 18, 30]. Image retrieval using sketched line-art depictions has been addressed by exploring the relationship between image edges and sketched lines. Matusiak *et al.* [27] proposed curvature scale-space [28] as a depiction invariant descriptor. Affine invariant contour representations for SBR were also explored by [17]. The relationship between edge detail and sketches was made explicit by Del Bimbo and Pala [10] where an deformable model derived from the sketch was fitted over image edges via non-linear optimization. More scalable solutions to image SBR have been proposed via the Structure Tensor[11], and the combination of Gradient-Field HoG descriptor and the Bag of Visual Words (BoVW) framework [16] initially proposed for QVE using photographic queries.

Although such sketch based image retrieval (SBIR) may be extended to video through key-frame extraction, motion also plays an important role within video content. A number of approaches [14, 2, 23, 3, 1] have explored the description of object trajectory through sketch, but neglect the appearance and semantic properties of the video content. Collomosse *et al.* combined sketched shape, colour and motion cue through free-hand storyboard sketches [8] — solving an inference problem to assign super-pixels in video to sketched objects at query-time. The expense of the inference step motivated Hu *et al.* to consider alternative approaches to matching storyboard sketches [15] using a trellis-based edit distance.

Our system directly builds upon [15], also adopting a edit-distance measure to match tokenized motion trajectories. However our system is unique in considering not only motion and colour, but also the semantic labelling of content within the video.

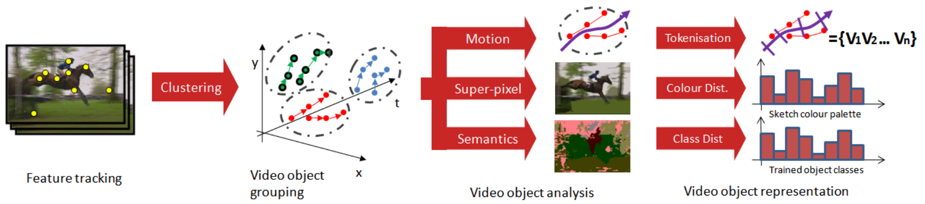

Fig. 1. Video pre-processing pipeline. Space-time video objects are first segmented from sparse tracked feature correspondences. Medial axis of each object's point cloud is extracted and augmented with colour and semantic label distributions obtained from local super-pixels.

This overcomes the scalability limitations inherent in the basic appearance features (e.g. colour, shape) considered in recent work [8, 15], and inherent in the medium of sketch, by relying instead on a user-annotated semantic labelling of sketched objects.

The consideration of semantics in SBR is currently sparsely researched. Semantic SBIR systems proposed by Liu *et al.* [24] and Wang *et al.* [35, 6] demonstrate how annotated examplar images can be exploited to retrieve *images*. Both systems used example images from a database of images (Mindfinder – found by interactive keyword search) to construct the query either using boxes or freehand shapes to depict the shape of the object. These approaches showed how semantic based retrieval is useful in adding spatial information to the user query, but are not suitable to be extended to video. To the best of our knowledge, the extension of semantic sketch to video has not been previously explored in literature.

2 System Overview

Our system parses videos upon ingestion to extract a set of video objects, identified as tracked clouds of sparse feature points (SIFT keypoints) undergoing self-consistent spatio-temporal motion. This motion segmentation step is performed via Affinity Propagation, as outlined in Sec. 3.1. The resulting video objects are analysed further to extract motion, colour and semantic labelling information. Motion trajectory is identified by a space-time curve representing the medial axis of motion, and sampling regular intervals along this curve to encode a series of 'tokens' that are later matched to the sketched curve using a modified Levenshtein (edit) distance. Mean-shift segmentation is applied to each video frame to yield a super-pixel representation, under which we can compute a colour distribution as later described in Sec. 3.2. A per-pixel semantic label is assigned to each video frame using a random-forest based labelling algorithm [31]. Thus the image pixels local to each tracked feature point within a video object grouping contributes to a colour and semantic distribution for that object. These three components (motion, colour, semantics) comprise the video object representation that we match against sketches at query-time (Sec. 4). Fig. 1 outlines the sequence of these pre-processing steps.

3 Video feature extraction

Upon addition of a new video to the dataset, we segment each video into clips using shot-detection [37]. Each clip is then processed to identify objects in the video that exhibit coherent motion relative to the background. Color, motion and semantic information is then extracted for each object.

3.1 Motion segmentation and trajectory extraction

Extracting and clustering motion trajectories is crucial for video processing and has been used for event analysis [29, 19], pedestrian counting [2] and video retrieval [15]. In this paper, we adopt an unsupervised motion clustering method to group the trajectories, generated by SIFT feature tracking, into different categories. The dominant trajectory of each motion category is represented with a piece-wise cubic β-spline.

Trajectory extraction: SIFT feature tracking has been used for video stabilization [4], object recognition and tracking [26, 36], as well as video retrieval [15]. In this paper we use SIFT keypoints matching to compensate the camera motion and generate the individual trajectories.

SIFT keypoints are detected on each frame and matched between each two adjacent frames. We iteratively correspond descriptors using the L^1 norm, the correspondences where the distance ratio of the best two matches falls below tolerance are disregarded as in [25]. Keypoints within the TV logo areas are not considered, and due to the constant location of such logos in our dataset (TSF [8]), they are trivially masked out. The inter-frame homography is estimated via MAPSAC using the keypoint correspondences. The locations of SIFT keypoints are transformed using the inverse homography to effect compensate for camera motion during the clip. Keypoints moving below a threshold velocity are discarded as unwanted background detail.

The correspondences of keypoints after camera motion compensation generate a set of individual trajectories. In order to filter and remove erroneous correspondences, we delete and interpolate the position keypoints where the inter-frame displacement deviates from the local average. Trajectories are fragmented into separate individual trajectories from the point of sudden changes of velocity [15].

Trajectory clustering: Tracklet representations, such as our SIFT trajectories, are frequently adopted as a basis for motion clustering in structure-from-motion applications [2, 23, 3, 1], though often at the expense of imposing a simplifying motion model (e.g. near-linear motion [15]). In [15] we construct a $5D$ feature space from the mean space-time location (x, y, t) and velocity $(\Delta x, \Delta y)$ of each trajectory. However, despite the simplicity of individual trajectories, a grouping of trajectories can encode non-linear motion. In this paper, we perform this grouping via Affinity Propagation clustering as follows.

Given the trajectory set, we compute the *affinity* of each trajectory pair and represent each as a node in an *affinity graph*. Each edge of the graph is weighted in proportional the affinity between the two nodes. Only trajectory pairs that share at least one common frame are considered to compute the affinity; the similarity between trajectories that do not share a common frame is set to be 0.

Let A and B be two trajectories sharing at least one common frame. The dissimilarity between A and B is defined as the distance of these two trajectories at a time instance where they are the most dissimilar:

$$d^2(A,B) = max_t d_t^2(A,B). \tag{1}$$

$d_t^2(A,B)$ is the distance of A and B at the particular time instant t:

$$d_t^2(A,B) = d_{sp}(A,B)\frac{(u_t^A - u_t^B)^2 + (v_t^A - v_t^B)^2}{3\sigma_t^2}. \tag{2}$$

where $d_{sp}(A,B)$ is the average spatial distance of A and B in the common time window; $u_t := x_{t+3} - x_t$ and $v_t := y_{t+3} - y_t$ measures the motion aggregation of the two trajectories over 3 frames; $\sigma_t = min_{a \in \{A,B\}} \Sigma_{t'=1}^{3} \sigma(x_{t+t'}^a, y_{t+t'}^a, t+t')$.

The similarity of trajectory A and B is then computed as:

$$sim(A,B) = exp(-kd^2(A,B)). \tag{3}$$

where in our experiments, constant $k = 0.1$. Having computed the affinity matrix, we apply the Affinity Propagation (AP) algorithm [12] to group the trajectories into different motion categories. In contrast to k-means clustering, AP requires only the similarity between trajectories as input, and does not require prior knowledge of the number of the clusters. Rather, AP considers all data points as potential exemplars and iteratively exchanges messages between data points until the corresponding clusters gradually emerges.

Motion representation: We extract a representative *medial axis* from each clustered component by approximating its global trajectory with a piece-wise cubic β-spline. The solution is unavailable in closed-form due to the typical presence of outliers and piece-wise modelling of complex paths. We therefore fit the spline using RANSAC to select a set of control points for the β-spline from the set of keypoints in the corresponding cluster. One keypoint is selected at random from each time instant spanned by cluster, to form the set of control points. The fitness criterion for a putative β-spline is derived from a snake [20] energy term, which we seek to minimize:

$$E = \alpha * E_{int} + \beta * E_{ext} \tag{4}$$

$$E_{int} = \int_{s=0}^{1} \left| d^2B(s)/ds^2 \right|^2 \tag{5}$$

$$E_{ext} = \Sigma_{t=0}^{T} \left[\frac{1}{|\mathcal{P}_t|} \Sigma_{p \in \mathcal{P}} |p - B(t/T)|^2 \right] \tag{6}$$

where $B(s)$ is the arc-length parameterised β-spline, and $\mathcal{P}_t, t = \{0..T\}$ is the subset of keypoints within the cluster at time t. We set $\alpha = 0.8$, $\beta = 0.2$ to promote smooth fitting of the motion path.

3.2 Color feature extraction

After motion clustering, each group of individual trajectories represents motion from one moving object which we term a *video object*. However, the sparsely detected SIFT keypoints within the video object typically exhibit insufficient pixel coverage to sample the colour appearance information of the video object.

We therefore segment each video frame into super-pixels of homogeneous color, using mean-shift [7] algorithm. The color of each keypoint along the trajectory is deemed as the mean color of the underlying region, and a weighted contribution is made to the histogram proportional based on the area of the region and the number of times that region been touched by trajectories from the according group.

The color distribution histogram is computed on all the keypoints along the trajectories of that category.

3.3 Semantic labelling

Pixelwise Semantic Segmentation has started to gain attention in recent years, approaches such as TextonBoost[31] and ALE[21] provide a accurate way of segmenting images. These approaches suffer from the computation of complex filter banks and assignment at test time, and the addition of K-Means at train time. An alternative to these approaches Semantic Texton Forests (STF)[32] used Extremely Randomised Decision Forests to classify pixels, these ensembles of decision trees are fast to train and test their inherent random approach allows them to be flexible to a variety of applications. In evaluation the STF computational performance makes it an attractive approach for semantically segmenting videos allowing for database scalability, the alternative texton based approaches are generally too slow to handle large datasets.

The STF approach is composed of two components, training of an ensemble of random decision trees. These trees are trained based on CIELab colour value differences within a window around the training point. The comparisons of values are based on a random comparison function, these can be addition, subtraction, absolute difference for example. The second component of this approach is a global image classification, this trains a OneVsOthers SVM other each of the classes, the approach uses a unique kernel based on Pyramid Matching Kernel(PMK). The PMK is adapted from the random decision forest based on node counts of the ensemble classified image, this adds some spatial consistency of class adjacent class context within images.

We apply the STF classifier to label the pixels in each the video frame as being in one of a pre-trained set of categories. In our experiments we train STF over twelve categories — corresponding to object classes with the TSF dataset, e.g. horse, grass, person, car. We count the frequency of label occurrence over all keypoints present within the spatio-temporal extend of the video object. The resulting frequency histogram is normalized via division by the number of keypoints, yielding a probability distribution for the video object's semantic label over the potential object categories.

4 Matching the Annotated Sketch

The basic unit of retrieval in our system is the video object, parsed via the process of Sec. 3. Video retrieval proceeds by independently estimating the similarity of each video object $v_i \in \mathcal{V}$ to the annotated sketch Q supplied by the user. This provides both a video and temporal window containing relevant content, which can be presented in a ranked list to the user.

The probability of object v_i corresponding to a given sketch query is proportional to a product of three orthogonal cues:

$$p(V|Q) \propto \operatorname*{argmax}_i \left[sim_C(v_i) \times sim_M(v_i) \times sim_S(v_i) \right]. \tag{7}$$

where sim_C, sim_M and sim_S denote the color, motion and semantic similarity of the i^{th} video volume to the query sketch Q respectively — as defined below.

4.1 Motion similarity (sim_M)

We follow the observations of [9], who observe that users depict object motion against a static background (the drawing canvas) regardless of any global camera motion present in the scene. This leads to a mapping between the sketch canvas and the camera-motion compensated frame derived from the inter-frame homographies computed within the video. Introducing a further assumption, we consider the sketched trajectory to depict the entirety of a video object's motion. We are then able to construct a space-time (x,y,t) trajectory from the sketched motion path — with time (t) spanning the temporal extent of the video object being matched, and (x,y) spanning the total camera panorama covered during that time.

The problem of matching the sketched motion path is thus reduced the problem of assessing the similarity of two space-time trajectories; that derived from the sketch, and that derived from the medial axis ($\beta-$spline) fitted to the video object's keypoints in subsec. 3.1.

Tokenization: We match the sketched motion trajectory to that of the video object by considering the path as a sequence of discrete movements, which we achieve by sampling the trajectory at regular arc-length intervals. In our experiments we sample ten intervals. A codebook of space-time moves is generated, and each trajectory segment assigned to a token in the codebook. The two strings are compared efficiently using the Levenshtein (edit) distance [22]; the minimal cost of operations required to transform one token sequence to the other. In our system we use the classical Levenshtein distance comprising insertion, deletion and substitution operators. The cost of insertion and deletion are defined as unity (i.e. high), with the substitution cost defined as the probability of two motion token being similar (derived from their space-time position and angle). The use of an edit distance measure enables two similar trajectories that exhibit temporally misaligned segments (due to inaccuracies in the sketch) being matched with low cost.

4.2 Colour similarity (sim_C)

Colour similarity is measured by comparing the non-parametric colour distribution of the sketched object with that of the video object being compared. The colour distribution is determined by computing a normalised frequency histogram from the colours

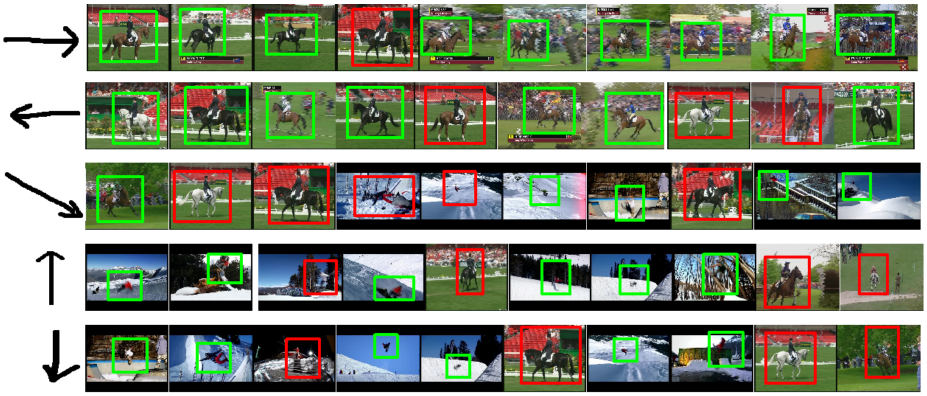

Fig. 2. Motion stroke queries and their top 10 returned results.

of pixels comprising the sketch. The set of colours comprising the histogram bins are derived from the discrete 16 colour palette available to the user during sketching; a similar palette is used when extracting the colour distribution from video objects with pixel colours conformed to this palette via nearest-neighbor assignment in CIELab space. The L^2 norm distance is used to compute the distance between two color histograms.

4.3 Semantic similarity (sim_S)

Our system enables the user to tag objects with a single object class, creating a class distribution for the sketched object with all contribution assigned to a single bin (object category). This histogram is directly compared with that of the video object, using the L^2 norm distance.

5 Experiments and Discussion

We evaluate our system over a subset of the TSF dataset, composed of 71 horse racing and 71 snow skating clips. For semantic labeling of the video frames, we define eight different semantic categories: person, horse, grass, snow, stands, plants, sky and void – although the void class is ignored for training. We manually label 143 frames from 12 video clips as training set, to classify the rest video frames.

Our system is tested in four different ways: use motion information alone as query; motion with color; motion with semantics, motion together with color and semantic information as queries. The example queries and their top 10 returned results are shown in Fig. 2 - Fig. 5 respectively. The positive results are highlighted in green and negative results are highlighted in red.

In Fig. 2 we demonstrate the effectiveness of our motion extraction and matching approach. The results over the selection of queries available for this dataset produce a Mean Average Precision(MAP) of 38.6%. Within the combination of motion and colour as queries as shown in Fig. 3, there is no shape information encoded therefore

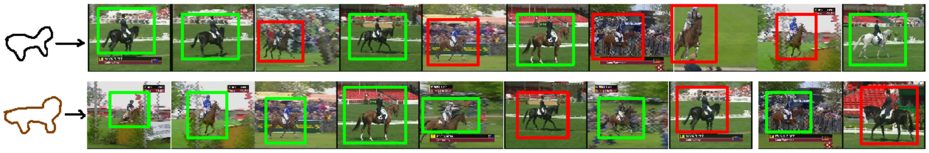

Fig. 3. Motion with color queries and their top 10 returned results.

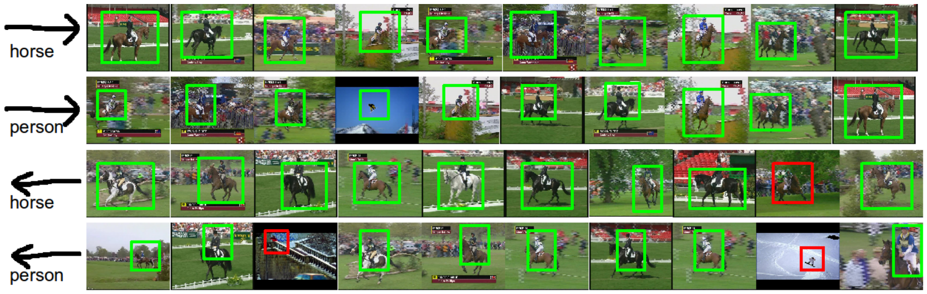

Fig. 4. Semantic with motion strokes as queries, and their top 10 returned results.

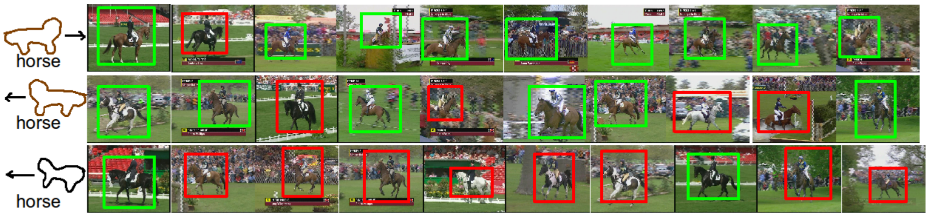

Fig. 5. Semantic query sketches and their top 10 returned results.

objects despite there depiction are referred to abstractly as a colour blob. These results demonstrate MAP of 42.7%.

The fusion of annotated class and motion as shown in Fig. 4, achieves a MAP of 75.85%. This improvement in contrast to motion alone demonstrates the advantages of annotated class as a facet of information.

When using the mix of all the different information sources as shown in Fig. 5, We achieve a MAP of 51.22%. The reduction in MAP is due to two main reasons, the difficulty in describing a feature points colour accurarly – generally in most scenarios the horse has a variety of colours on them even with the mean-shift filtering there are still regions such as the leg of the rider that are difficult for both the semantic segmentation and the colour description to deal with. Also with the amalgamation of all the different facets of information reduces the possible accurate results in the dataset down making it very difficult to get an accurate result.

Average Precision Recall curves of the four evaluated systems are ploted in the left of Fig. 6. From the curves we can see that by adding semantic information into the color, and motion query can significantly improve the performance of the retrieval

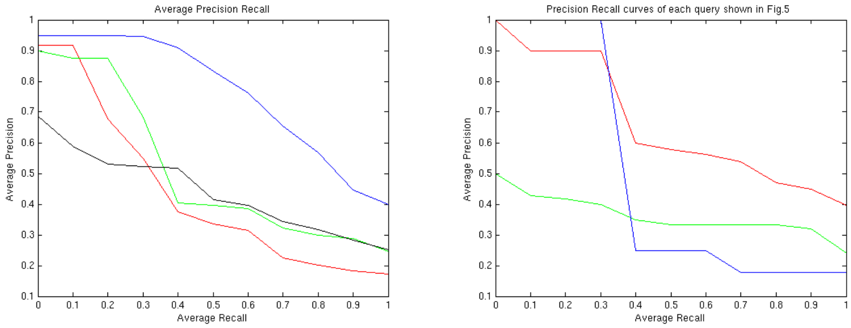

Fig. 6. (left)Average Precision Recall curves of using motion (black curve), motion with color (red curve), motion with semantics (blue curve), motion with color and semantics together (green curve) based retrieval. (right) Precision Recall curves of the three queries shown in Fig. 4. The curve for the query on the top is shown in red; the middle is shown in green; and the bottom one is shown in blue.

system. The figure on the right side of Fig. 6 show the precision recall curve of each of the three queries in Fig. 5.

6 Conclusion

We have presented a video retrieval system driven by annotated sketched queries. Salient objects are identified within video through unsupervised clustering of SIFT keypoint trajectories in a camera-motion compensated frame. Each object is analysed to develop an augmented object description comprising data on space-time locus (spatial position and motion path), colour and object category. The motion is derived from a $\beta-$spline robustly fitted in space-time to keypoints comprising the object. Although semantic sketch based retrieval has been recently applied to images [35, 6], our system is the first to explore the use of semantic (annotated) sketches for video retrieval. We have demonstrated improved retrieval performance through the integration of semantics, over previous sketch based video retrieval techniques using colour and motion alone [15].

Having incorporated multiple orthogonal cues into a video retrieval system, a natural direction for future work is explore the relative weightings of those cues. Such weightings seemingly cannot be prescribed in advance; a user sketching a red blob labelled “car” travelling right, may assign greater worth to red cars travelling left — or to yellow cars travelling right. Interactive relevance feedback, enabling re-weighting of the terms of eq. 7 seems a promising approach to resolving this ambiguity behind a user’s intention.

References

1. N. Anjum and A. Cavallaro. Multifeature object trajectory clustering for video analysis. *IEEE Trans. on Circuits and Systems for Video*, 18(11):1555–1564, 2008.

2. G. Antonini and J. P. Thiran. Counting pedestrians in video sequences using trajectory clustering. *IEEE Tran. on Circuits and Systems for Video*, 16(8):1008–1020, 2006.
3. F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Trans. Multimedia*, 9(1):58–65, 2007.
4. Sebastiano Battiato, Giovanni Gallo, Giovanni Puglisi, and Salvatore Scellato. Sift features tracking for video stabilization. In *International Conference on Image Analysis and Processing*, pages 825–830, 2007.
5. Marco Bertini, Alberto Del Bimbo, and Walter Nunziati. Video clip matching using mpeg-7 descriptors and edit distance. In *CIVR*, pages 133–142, 2006.
6. Yang Cao, Hai Wang, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. Mindfinder: interactive sketch-based image search on millions of images. In *ACM Multimedia*, pages 1605–1608, 2010.
7. Christopher M. Christoudias, Bogdan Georgescu, and Peter Meer. Synergism in low level vision. *ICPR*, 4:40150, 2002.
8. J. Collomosse, G. Mcneill, and Y. Qian. Storyboard sketches for content based video retrieval. In *ICCV*, 2009.
9. J. Collomosse, G. Mcneill, and L. Watts. Free-hand sketch grouping for video retrieval. In *ICPR*, 2008.
10. A. del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. 19(2):121–132, 1997.
11. M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. In *IEEE TVCG*, volume 99, 2010.
12. Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.
13. J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Effcient color histogram indexing for quadratic distance. *IEEE PAMI*, 17(7):729–736, 1995.
14. J. Hsieh, S. Yu, and Y. Chen. Motion-based video retrieval by trajectory matching. *IEEE Tran. on Circuits and Systems for Video*, 16(3):396–409, 2006.
15. R Hu and J Collomosse. Motion-sketch based video retrieval using a trellis levenshtein distance. In *Intl. Conf. on Pattern Recognition (ICPR)*, 2010.
16. Rui Hu, Mark Barnard, and John P. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *ICIP*, pages 1025–1028, 2010.
17. Horace H. S. Ip, Angus K. Y. Cheng, William Y. F. Wong, and Jun Feng. Affine-invariant sketch-based retrieval of images. In *International Conference on Computer Graphics*, pages 55–61, 2001.
18. C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multi-resolution image querying. In *Proc. ACM SIGGRAPH*, pages 277–286, 1995.
19. Cláudio Rosito Jung, L. Hennemann, and Soraia Raupp Musse. Event detection using trajectory clustering and 4-d histograms. *IEEE Trans. Circuits Syst. Video Techn.*, 18(11):1565–1575, 2008.
20. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Intl. Journal of Computer Vision*, 4(1):321–331, 1987.
21. Pushmeet Kohli, Lubor Ladický, and Philip H. S. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. In *International Journal of Computer Vision*, volume 82, pages 302–324, 2009.
22. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, Soviet Physics Doklady, 1966.
23. Xi Li, Weiming Hu, and Wei Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *ICPR*, pages 591–594, 2006.
24. C. Liu, D. Wang, X. Liu, C. Wang, L. Zhang, and B. Zhang. Robust semantic sketch based specific image retrieval. In *Proc. Intl. Conf. and Multimedia Expo*, 2010.

25. D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.
26. Fernando Lpez-Garca. Sift features for object recognition and tracking within the ivsee system. In *ICPR*, pages 1–4. IEEE, 2008.
27. Stanislaw Matusiak, Mohamed Daoudi, Thierry Blu, and Olivier Avaro. Sketch-based images database retrieval. In *International Workshop on Advances in Multimedia Information Systems*, pages 185–191, 1998.
28. Farzin Mokhtarian and Alan K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:789–805, 1992.
29. C. Piciarelli and G. L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recogn. Lett.*, 27:1835–1842, 2006.
30. E. Di Sciascio, G. Mingolla, and M. Mongiello. CBIR over the web using query by sketch and relevance feedback. In *Proc. Intl.Conf. VISUAL*, pages 123–130, 1999.
31. J Shotton, J Winn, C Rother, and A Criminisi. TextonBoost : Joint Appearance , Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *ECCV*, pages 1–14, 2006.
32. Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, pages 1–8, 2008.
33. J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV*, pages 2:1470–1477, 2003.
34. E. Tulving. *Elements of episodic memory*. 1983.
35. Changhu Wang, Zhiwei Li, and Lei Zhang. Mindfinder: image search by interactive sketching and tagging. In *WWW*, pages 1309–1312, 2010.
36. Jie xu, Getian ye, and jian zhang. Long-term trajectory extraction for moving vehicles. In *IEEE International Workshop on Multimedia Signal Processing*, pages 223–226, 2007.
37. HongJiang Zhang, Atreyi Kankanhalli, and Stephen W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.