Markov Random Fields for Sketch based Video Retrieval

Rui Hu Stuart James Tinghuai Wang John Collomosse
Centre for Vision Speech and Signal Processing
University of Surrey
Guildford, UK
{r.hu, s.james, tinghuai.wang, j.collomosse}@surrey.ac.uk

ABSTRACT

We describe a new system for searching video databases using free-hand sketched queries. Our query sketches depict both object appearance and motion, and are annotated with keywords that indicate the semantic category of each object. We parse space-time volumes from video to form graph representation, which we match to sketches under a Markov Random Field (MRF) optimization. The MRF energy function is used to rank videos for relevance and contains unary, pairwise and higher-order potentials that reflect the colour, shape, motion and type of sketched objects. We evaluate performance over a dataset of 500 sports footage clips.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing Methods; H.3.3 [**Information Search and Retrieval**]: Retrieval models; Search process

Keywords

Sketch based Video Retrieval (SBVR), Markov Random Field (MRF), Storyboard Sketch, Semantic Labelling.

1. INTRODUCTION

Video repositories are typically searched by matching text queries to keywords that have been manually assigned to each clip. Although keywords are efficient *semantic* descriptors of content (e.g. "horse", "car") they are inefficient at describing the *appearance* or *motion* of those objects. Furthermore the level of annotation — at the level of the *clip*, rather than frames or even objects within frames — limits the spatial and temporal resolution at which video may be searched. Querying by *Visual* Example (QVE) offers a solution, yet many QVE systems require photorealistic queries (e.g. images [1], or video [2]) that may not be available to the user at query-time.

In this work we describe a novel system for searching video clips using *annotated free-hand sketches*. Our query sketches depict the appearance and motion of objects, which are each annotated to indicate their semantic category. Rather than relying on keyword annotation at the level of the clip to match the latter, we harness a semantic segmentation algorithm to label video regions using a set of pre-determined object categories (e.g. grass, person, horse). In this respect, our system extends the "storyboard sketches" proposed by Collomosse *et al.* [3] for Sketch Based Video Retrieval (SBVR). Storyboard sketches are free-hand sketches drawn by the users depicting both the video content and dynamics (using arrows). Our system not only incorporates the annotation of objects in storyboard sketches with semantic tags, but also improves upon [3] through faster matching and the handling of non-linear object motion.

Our core contribution is a Markov Random Field (MRF) based framework capable of evaluating the support for a *query sketch* within a given *video*, and hence the likelihood of the two matching. We over-segment the video into a set of space-time sub-volumes, each of which forms a node in a graph with connectivity determined using space-time adjacency of sub-volumes. A graph-cut operation [4] identifies the sketched object under an MRF defined across the nodes in this graph, as well as determining a likelihood score for the purpose of inter-video comparison. The potentials on the MRF incorporate both semantic similarity, and appearance similarity as a function of colour, motion, and shape. In considering shape and motion, information from spatially 'higher order' segmentations of the video are also considered. We also correct for global camera motion in the scene present, caused by the camera tracking moving objects.

2. RELATED WORK

Sketch based QVE largely focuses on the image retrieval problem. Early sketch based image retrieval (SBIR) systems accepted queries comprising blobs of coloured texture, matched through region adjacency and topology [5, 6], shape [7], or spectral descriptors such as wavelets [8]. More recently, SBIR has been applied to large scale (>1 million record) retrieval by matching line-art query sketches to edge information within photographs [9, 10, 11, 12]. These systems have demonstrated the value of sketches as an effective tool for shape and appearance based SBIR.

Sketch has also been applied to motion retrieval within video using sketched object trajectories [13, 14]. However, the combined use of appearance and motion cues in SBVR has been sparsely researched. VideoQ [15] is one of the first SBVR systems that consider spatio-temporal attributes. However, VideoQ requires users draw exact motion curve and

ICMR'13, April 16–20, 2013, Dallas, Texas, USA.

specify the object's speed (in pixels/second). More recent SBVR systems accept greater flexibility in the specification of both appearance and motion, making them more amenable to depictions of events recalled under the ambiguities of human episodic memory [3].

Our work is aligned with the latter approach of Collomosse *et al.*, which also seeks a maximum likelihood labelling of super-pixels to the objects depicted within a storyboard sketch. Given the ambiguity inherent in users' sketches, we also find it attractive to treat the sketch as a probabilistic model to be fitted to video under a constrained optimization. Nevertheless, there are a number of key differences between our contribution and this prior work [3].

First, we incorporate semantics within our optimization framework rather than relying on appearance and motion alone. This improves scalability over larger video datasets, as the ambiguity inherent in user sketches can result in numerous coincidental false positives (e.g. based on shape or colour) as the dataset grows. The presence of both semantic and appearance information in the sketch overcomes the in-principle limitation of matching based on appearance alone. Second, we formulate our optimization as an MRF which may be solved orders of magnitude faster than Collomosse *et al.*'s Linear Dynamic Systems (LDS) and to give a globally rather than locally optimal video labelling. These efficiencies are largely due to our video representation which employs space-time volumes as atomic units for labelling, rather than spatial (per-frame) super-pixels. Third, our representation of object motion admits non-linear trajectories which are unavailable to [3]. We further enhance the features matched by our system by incorporating a state of the art descriptor (GF-HOG) for sketch based shape matching [16].

Another trend of SBVR systems match query sketches with spatio-temporal sub-volumes segmented from video. Hu *et. al.* [17] track SIFT keypoints to form short trajectories which are clustered to form a set of space-time tokens. A Viterbi-like process matches the space-time graph of tokens to the colour and motion description of the query sketch. The approach is extended in [18] using a more robust motion clustering algorithm, where semantic information is also considered. However, as with early SBVR [15], retrieval performance is strongly dependent on the accuracy of the video segmentation.

In contrast to these approaches that pre-process video into segments offline, our contribution is to segment video at query-time using a Markov Random Field (MRF) optimization that simultaneously ranks clips for relevance and localises the sketched object. MRFs have been successfully used to find a globally optimal segmentation of images [4, 19] and videos [20, 21]. A restriction on their use is frequently cited to be high computational complexity, since individual pixels are used as the nodes in the graph (lattice). In this work we propose to represent each video by an irregular spatio-temporal graph, containing a few hundred nodes each of which is a spatio-temporal fragment within the video (analogous to a super-pixel within an image).

In MRFs commonly used for segmentation, an image or video is encoded as an undirected graph — representing anything from a regular lattice of pixels [4, 19, 21], to an irregular network of regions [20]. A Gibbs energy function is defined, often containing a unary data term and a pairwise term, the minimum of which is sought to divide (cut) the graph, and so yield a segmentation. Recent research has extend this function to include higher order constraints [22, 21] enforcing labelling consistency within a local neighbourhood. Our contribution is to apply this latter development to the SBVR problem; specifically to label spatio-temporal fragments to sketched objects. To the best of our knowledge, a graph cut solver has not been used in this way for SBVR — nor have the modalities of semantics, shape, motion and colour been previously combined within an SBVR system.

3. SYSTEM OVERVIEW

Our system accepts keyword annotated *storyboard sketches* [3] as queries to retrieve similar videos in a dataset. We formulate the video retrieval problem as a pixel labelling and matching problem which is solved by a graph cut optimization to simultaneously estimate the likelihood of the match, whilst also localising the sketch object in the video. Fig.2 shows two example sketch queries, their top returned results and the estimated foreground area. Given a video dataset, we pre-process each video in to a set of space-time subvolumes — analogous to the spatial concept of superpixels [23] — via a process described in section 4. The sub-volumes form nodes in an undirected graph with edges linking a pair of nodes when the respective sub-voumes are adjacent in space-time (section 5.1).

Upon accepting a query sketch, our retrieval system segments each video (i.e. label sub-volumes) into background and foreground regions; the latter being the sketched object of interest. The unary term in the graph cut measures the agreement between foreground nodes with a model built from the sketched query. Agreement is measured using a weighted combination of similarity scores expressing motion, colour, semantic and shape similarity. Similarly, a background model is also learnt offline from the video. Pairwise terms are computed between nodes using a similar set of appearance attributes. A graph cut solver generates an optimal labelling, with an associated normalised energy which is returned to the system as a (dis-)similarity measure to rank video clips with respect to the query.

3.1 Sketch Parsing and Description

The query sketch is a combination of strokes that coarsely depict object shape and colour, with the addition of 'motion strokes' that depict the movement of the object. Additionally, keywords are assigned to objects from a pre-determined set of categories, to specify its semantic class. Optionally, user may also depict the background color and semantics (e.g. green grass).

These attributes are extracted from the query using the following features:

Color distribution. A color histogram is computed from pixels within the sketch, indicating the frequency of occurrence for each of the 15 colours in the user's palette. The histogram encodes an area-weighted, non-spatial colour distribution of the desired object.

Shape. The GF-HOG framework of [9] is applied to compute a shape descriptor from the sketch. GF-HOG applies Laplacian constraints to smoothly extrapolate a dense field of edge orientations from sketched strokes. Histogram of Gradient (HOG) features are sampled along sketched strokes at varying spatial scales. A standard hard-assignment Bag of Visual Words (BoVW) pipeline converts these descriptors into a frequency histogram of codewords. We use a code-

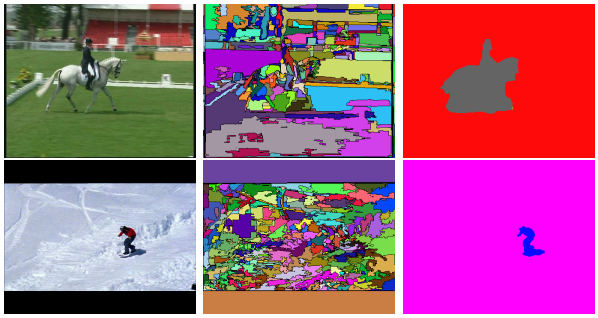

Figure 1: Left: video frame. Middle: over segmented spatio-temporal supervoxel. Right: motion segmentation resulted subvolume.

book size of 1000 in our experiments. To enable comparison of this shape descriptor with videos at query-time, the common codebook generated during video ingest (section 4.3) is used to generate the BoVW histogram.

Motion direction. Strokes indicating the motion direction are drawn in a different ink which is used to depict the object. Simple arrow pictograms may be recognised in the sketch using [24] and the shaft of the arrow isolated. The shaft sampled at regular intervals to yield a sequence of vectors of constant length, which represent the motion of the according segment.

Semantics. Our interface provides keywords describing eleven semantic classes from which users can pick to annotate their sketched object. This information is encoded by a probability vector across those classess; each keyword on the object is weighted equally (e.g. 2 keywords produces a 50:50 distribution over two bins).

4. VIDEO PRE-PROCESSING

For each video within our dataset, we conduct the following pre-processing steps. Later, the representation parsed from the video in this pre-process is matched with the representation parsed from the sketch (section 3.1) at query-time, via the method outlined in section 5.

We begin by applying shot detection to temporally segment video into clips, each of which forms a candidate for retrieval within our video dataset. SIFT keypoints are detected on every frame of a clip, and correspondence robustly established between adjacent frames to compute inter-frame homographies which we take as approximating camera egomotion over time. We also compute the pixel-wise foreground probability for each frame. A background mosaic is constructed by warping and averaging temporally neighbouring frames under their homographies. The difference between the current frame and its temporally local background mosaic is used as the foreground probability map for that frame.

4.1 Spatio-temporal video over-segmentation

Many vision applications have benefit from representing an image as a collection of superpixels [22, 25, 20, 26]. Superpixels are spatially coherent groups of pixels that are similar in color and texture, so in turn tend to constitute a semantic object, or part thereof. This assumption leads to advantages of superpixel primitives over pixels, both in terms of computational efficiency and improved local consistency in segmentation problems. In order to increase the chance that superpixels do not cross object boundaries, an oversegmentation is often preferred.

We oversegment our videos into a set of spatio-temporal sub-volumes, which we refer to as "supervoxels" by analogy with super-pixels. We adopt the video segmentation algorithm proposed in [23]. This algorithm can segment a video into a hierarchy scale of spatio-temporal supervoxels; here we use the volumes from the finest scale level. Fig.1 (middle) gives an example of an oversegmented video. Video clips in our dataset are typically segmented into around 2k supervoxels, ensuring compact graphs for the subsequent optimization process at query-time.

4.2 Motion segmentation

In addition to performing a supervoxel segmentation cued colour and texture, we run a coarser grain segmentation of the video sequence cued on motion. It is difficult to meaningfully describe motion at the fine scale of a supervoxel. Similarly, appearance attributes such as shape are better described over larger spatial areas.

We apply the motion segmentation algorithm proposed in [27], and implement their recommendation to post-process the resulting sparse point labelling to obtain a dense motion segmentation result. Example motion segmentation results are shown in Fig.1 (right).

The resulting coarse-scale supervoxels identified by the motion segmentation process are used later in the higher order term of our energy function (section 5.2) as a soft constraint to improve spatio-temporal labelling consistency. Shape features for each fine-scale supervoxel (obtained via 4.1) are later computed within the scope of the coarse scale supervoxel in which they predominantly reside. As we are not matching these coarse scale sub-volumes to the query sketch directly, we do not require each coarse supervoxel to exactly represent a single object.

4.3 Supervoxel feature extraction

Our retrieval process aggregates the fine-scale supervoxels obtained in section 4.1 to form objects represented by the query sketch. We therefore extract a set of features from each of these supervoxels, to encode similar cues to that parsed from the query sketch in section 3.1.

Foreground probability. Each supervoxel is assigned a probability of being in the foreground. This is obtained using the pixel-wise foreground score obtained using the mosaic background subtraction performed earlier. This score is averaged across the entire supervoxel.

Color distribution. As in section 3.1, a colour histogram is built to represent the color distribution of all the pixels the supervoxel contains. As pixels may deviate from the 15 colour user palette, the histogram bins are contributed to in proportion to the RGB distance between pixel colour and palette colour.

Motion direction. The footprint of the supervoxel is computed within each frame it spans, yielding a sequence of region masks from which we obtain a sequence of centroids. We average the vectors between these centroids to produce an indicative direction for each supervoxel. These vectors are later aggregated across supervoxels and matched to the

sequence of motion vectors parsed from the sketched trajectory.

Shape. We apply the algorithm of Hu *et al.* [9] to compute a set of sparse GF-HOG descriptors across each video frame. Descriptors are quantised into visual words using a pre-computed codebook, obtained using k-means to cluster GF-HOG descriptors within video frames sampled at random across the dataset. In our experiments we constructed this codebook by sampling 10k random frames, and compute a codebook of size 1000. We aggregate descriptors within the local neighbourhood of the supervoxel into a frequency histogram, which is subsequently normalised. The neighbourhood is defined by the coarse supervoxel (obtained via the motion segmentation process of section 4.2) that predominantly contains the fine-scale supervoxel being processed.

Semantics. Pixelwise semantic labelling (also referred to as semantic segmentation) of images has started to gain attention in recent years. We apply the Semantic Texton Forests (STF)[28] classifier to label the pixels in each video frame as being in one of a pre-trained set of categories. In our experiments we train STF over eleven categories — corresponding to object classes within our video dataset, e.g. horse, grass, person, snow, et.al. In a one-off manual process, we hand-label around 250 frames from exemplar video clips in the dataset to serve as training data for STF. The pixel-wise labelling probabilities are accumulated within the supervoxel, and normalised to yield a probability distribution over the eleven semantic classes for each supervoxel.

5. GRAPH CUT BASED VIDEO RETRIEVAL

We propose a spatio-temporal graph representation of videos and formulate the video retrieval problem as a supervoxel-labelling and matching problem. Each supervoxel is assigned as either the user depicted foreground object (by query sketch) or the background. This is solved by graph cut as a global optimization problem. The normalised cost of the energy function (section 5.2) is used as the dissimilarity value of this video to the query sketch.

In the following, we explain in detail how we construct the graph model, formulate and optimize the energy function as well as how the retrieval system is built.

5.1 Spatio-temporal graph construction

For each video, we construct an undirected spatio-temporal graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{A} \rangle$. The node set $\mathcal{V}$ contains the over-segmented spatio-temporal supervoxels (as introduced in section 4.1). Two terminal nodes indicating the foreground model (depicted by the user sketched query) and the background model learned from individual video or depicted by query sketch. The arc set $\mathcal{A}$ consists both the 'neighborhood links' (*n-links*) and the 'terminal links' (*t-links*). Each node (supervoxel) in the graph has two *t-links* connect this node to the two terminal nodes; and *n-links* indicate the connection to its adjacent supervoxels. Two supervoxels are considered as connected is they share boundary either spatially (intra-frame) or temporally (inter-frame). Note that the segmented supervoxel could be of any shape and size both spatially and temporally. This makes our generated graph irregular unlike many pixel based models.

The arc between two nodes (*n-link*) indicates the similarity between these two adjacent supervoxels, which is pre-computed offline, based on their color, motion, semantic and foreground probability similarity. Note that the shape feature is not considered when measuring the neighborhood similarity. The arc that connect each node to the terminal nodes (*t-link*) are defined as the cost of labelling the according supervoxel to foreground and background. In our case this is computed as their dissimilarity in the feature space to each model.

Note a similar concept of spatio-temporal graph is also used in [20] for video segmentation. The nodes in their graph are superpixels segmented from each frame by 2D image segmentation algorithm and the arcs are defined by inter and intra frame colour similarity within a neighborhood area. While in our work, the nodes in our graph are spatio-temporal coherent supervoxels. Our unary and pairwise term are built on appearance, motion and semantic features which brings more rich information to the graph optimization.

5.2 Definition of Energy Potentials

Given the graph $\mathcal{G}$, a finite set $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$ of labels. $\mathcal{L}^V$ represents all the possible labelling stragegies for the node set. $X \in \mathcal{L}^V$ is a map that assigns to each vertex v a label x_v in $\mathcal{L}$. An energy function E maps any labelling strategy X to a real number $E(X)$ denoted as its energy. Energy functions are defined as the cost of the according labelling strategy. Therefore, finding the optimum labelling strategy is equivalent to find the minimum cost of the energy function.

Similarly to [21], our energy function consists of unary, pairwise and higher order terms as:

$$E(X) = \alpha \sum_{i \in \mathcal{V}} \psi_u(x_i) + \beta \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_p(x_i, x_j) + \gamma \sum_{i \in \mathcal{V}} \sum_{c \in \mathcal{S}} \psi_h(x_i) \quad (1)$$

where $\alpha, \beta, \gamma, \in [0,1], (\alpha + \beta + \gamma = 1)$ are weights for the unary term ψ_u, pairwise term ψ_p, and the higher-order term ψ_h respectively. $\mathcal{V}$ corresponds to the set of all super-voxels in the video, $\mathcal{S}$ represents the set of sub-volumes segmented by motion segmentation, $\mathcal{N}_i$ indicates the neighbouring supervoxel set of the current super-voxel i. This energy function encourages the neighbouring consistency both spatially and temporally. Moreover, the higher-order potential term increase the label consistency inside the sub-volumes generated by motion segmentation. The detail of how each of the potentials are defined is described in the following.

5.2.1 Appearance and motion model

The unary term ψ_u exploits the fact that different appearance and motion homogeneous voxels tend to follow different labelling models. In our case, its a binary labelling problem. This term encourages each super-voxel been assigned to its most similar model. We use the cost of a label being assigned to super-voxel i as the unary potential, which is computed as a weighted sum of the dissimilarity of colour, shape, motion, semantics to the model as well as the the probability to be foreground or background. The unary term is computed as:

$$\psi_u(x_i) = \theta_{cl}\psi_{cl}(x_i) + \theta_{sp}\psi_{sp}(x_i) + \theta_{mt}\psi_{mt}(x_i) + \theta_{sm}\psi_{sm}(x_i) + \theta_{fg}\psi_{fg}(x_i) \quad (2)$$

where $\theta_{cl}, \theta_{sp}, \theta_{mt}, \theta_{sm}, \theta_{fg}, \in [0,1], (\theta_{cl} + \theta_{sp} + \theta_{mt} + \theta_{sm} + \theta_{fg}) = 1$, are weights of colour, $\psi_{cl}(x_i)$, shape $\psi_{sp}(x_i)$, mo-

tion $\psi_{mt}(x_i)$, semantics $\psi_{sm}(x_i)$, and foreground potentials $\psi_{fg}(x_i)$ respectively.

We now first explain how we build models for the foreground and background labels. In most graph cut based systems, labelling models are often pre-defined manually [21], or online learned [29]. In this paper we build the foreground model from the query sketch. The appearance, motion and semantic features extracted from the query sketch is used as the foreground object feature model. When the background is also defined in the query sketch, the model of which could be learnt similarly as we learn the foreground model. Otherwise, we assume the 1% supervoxels that has the lowest foreground probability as definite background. The background model is built as the area weighted average feature vectors of supervoxels from the definite background area.

For the colour shape and semantic features the distance between a node and the foreground/background node is computed using the cityblock measure. In the case of the motion the query motion stroke is quantised into same number of equal segments as the number of frames of the matching video. Nodes from the video are mapped onto the corresponding segments from the quantised query motion stroke. The angular distance is computed over each segment and averaged as the node motion dissimilarity term. The foreground probability of each supervoxel $f_i, (i \in \mathcal{V})$ is directly used as the unary potential to be labelled as the background model.

5.2.2 Spatio-temporal Coherence model

The pairwise term is often used to encourages spatial coherence in region labelling and discontinuities to occur at high contrast locations. Given the graph defined in our paper, each node is a spatio-temporal supervoxel. Appearance feature alone is not enough to define the voxel coherence. Therefore, we define the neighbouring coherence as a weighted fusion of the similarity values based on colour, motion and semantic features. Our pairwise term is defined as:

$$\psi_p(x_i, x_j) = \begin{cases} 0, & \text{if } x_i = x_j \\ e^{-d_{i,j}}, & \text{if } x_i \neq x_j \end{cases} \quad (3)$$

where $d_{i,j}$ is the weighted sum of distance between nodes i and j using different features. Given the feature representations of each supervoxel, the distance between which is computed similarly as we compute the distance to the foreground/background models (unary term). Note, that shape feature is not considered to measure the supervoxel similarity in the pairwise term.

5.2.3 Motion segments Consistency Term

In recent works, a higher-order term is often defined in the energy function to encourage the pixels belonging to a super-pixel to be assigned with the same label.

Similarly to [22, 21], we define a soft constraint to reflect the label consistency. However, different with their pixel-wise graph model where over-segmented superpixel is used in the higher order term to improve the labelling consistency, our model takes the supervoxels as nodes and the motion segmentation resulted sub volumes are used in the higher-order term to encourage spatio-temporal labelling consistency. We define this term as a weighted sum of unary potentials of all the supervoxels within the current subvolume to be labelled the same as the current node i:

$$\psi_h(x_i) = \begin{cases} 0, & \text{if } i \notin m \\ \frac{1}{\sum_{j \in m} a_j} \sum_{j \in m} a_j \psi_j(x_i), & \text{if } i \in m, \end{cases} \quad (4)$$

where $m \in \mathcal{M}$ is one sub-volume of the motion segmentation sub-volume set $\mathcal{M}$, $j \in m$ represents each of the supervoxels that belongs to sub-volume m, $\sum_{j \in m} a_j \psi_j(x_i)$ is the weighted cost if all supervoxels constituting m are labelled as x_i (the current labelling for node i), weight a_j is the total number of pixels within supervoxel j, $\psi_j(x_i)$ is thus defined as the unary potential of supervoxels in m against label x_i. This function indicates that an optimal label assignment to node i should also fit all supervoxels within the same motion segmented subvolume. Since supervoxels within one motion segmented subvolume are represented by the same subvolume, it is not necessary to compute shape feature again in this higher-oder term.

5.3 Optimization

Similarly to [21], our 'higher order' term can also be effectively merged to unary term. So that the energy function Eq. 1 can be simplified to:

$$E(X) = \sum_{i \in \mathcal{V}} (\alpha \psi_u(x_i) + \gamma \sum_{c \in \mathcal{S}} \psi_h(x_i)) + \beta \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_p(x_i, x_j) \quad (5)$$

The simplified energy function in Eq. 5 is of the form of Potts model. It can be minimised using the α-expansion and $\alpha\beta$-swap algorithm [4]. Each α -expansion iteration can be solved by performing a single graph-cut using the min-cut/max-flow [30]. We also use the same technique in [31] to improve the optimization process.

5.4 Video retrieval

From Eq. 5, the energy of each optimization iteration defines the cost of the according labelling strategy. The optimized minimum energy indicates the cost of the best strategy to spatio-temporally cut the video clip into the query sketch depicted foreground object and the background. The more similar the sketch query depicted foreground object (and background if applicable) to the video, the smaller cost could spend to match them, i.e. given one video clip and several sketches each depicting a different object (and scene), the smaller the final energy cost to match the video clip to one of the sketches, the more similar the pair are. Therefore, the final optimized energy could be used as a dissimilarity between a video clip and a query sketch.

However, since the energy function defined in Eq. 5 is also related with the number of nodes (the unary and higher order term) and links (the pairwise term) within the graph, it can not be directly used to rank the similarity of a collection of videos to the query sketch. In our work we apply the normalised energy function in our video retrieval system. We normalise the energy as following:

$$E'(X) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{V}} (\alpha \psi_u(x_i) + \gamma \sum_{c \in \mathcal{S}} \psi_h(x_i)) + \beta \sum_{i \in \mathcal{V}} (\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \psi_p(x_i, x_j))$$

where $|\mathcal{N}|$ is the total number of supervoxels in a video clip, $|\mathcal{N}_i|$ is the number of neighbours. and this normalised energy E' will be used as the dissimilarity score to rank videos in our retrieval system.

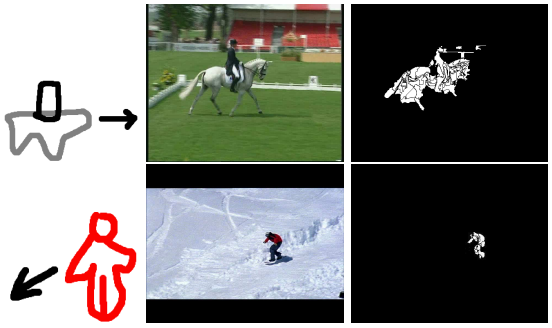

Figure 2: Left: The query sketch. Middle: top returned video frame. Right: The estimated foreground area that matches with the query sketch.

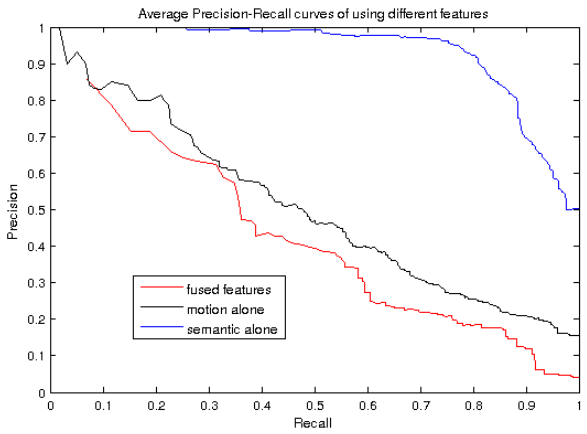

Figure 3: Average Precision-Recall curves of using motion stroke alone, semantic keywords alone to retrieve video clips and our annotated *storyboard sketches* that depicting the color, motion, shape and semantics of objects (and background if sketched).

In our proposed video retrieval system, the off-line processing steps include: video pre-processing (section 4), computing the pair-wise potential and the unary potential to the background model built from the video itself. Upon accepting a query sketch, we first build the foreground model using the sketch and then match with the pre-computed features of supervoxels (the unary potential to the foreground), the higher order potential, and then the graph is optimized.

6. EXPERIMENTS

We evaluate our system over a sport footage composed of 500 video clips, among which there are 304 horse riding and 196 snow boarding/ski video clips. Objects/scene within these video clips contains: person, horse, grass, snow, stands, tree, sky, water. Camera motion happens in most of the clips. This dataset is comparable to the 'TSF dataset' used in [3] (which contains 298 clips); and the 200 similar clips used to evaluate VideoQ [15].

6.1 Parameter settings

In video preprocessing step, we use the default parameters for both the supervoxel and subvolume segmentation. We use the same parameters as in [9] to extract the GF-HOG feature, for each point along sketch/edge we compute histogram distribution of eight orientations on a 3×3 grid with three window size (5, 10, 15).

In the graph cut model, there are two sets of weightings. One is the weights for the three energy terms, and the other is the weights of different features. Both weights can be freely adjusted by the users according to their preference. In our experiments, we use the same weighting parameters through all videos in our dataset. Weighting parameters for each term in the energy function are set by experience, we use $\alpha = 0.9$, $\beta = 0.05$, $\gamma = 0.05$. In the unary term we set $\theta_{cl} = 0.25$, $\theta_{sp} = 0.05$, $\theta_{mt} = 0.3$, $\theta_{sm} = 0.3$, $\theta_{fg} = 0.1$. Since shape feature is not considered in the pairwise and higher-order term, we set the feature weights for these two terms as: $\theta_{cl} = 0.3$, $\theta_{mt} = 0.3$, $\theta_{sm} = 0.3$, $\theta_{fg} = 0.1$.

6.2 Performance Evaluation

Our system takes into consideration spatio-temporal features to match video clips to sketch queries. In order to understand how each of these components works we first visualize some retrieval results of using color, motion, semantic keywords alone in Fig.4. Each of these features alone is typically used in appearance, motion or keywords based video retrieval systems. This example shows that although each component alone is able to find the related clips by considering the particular feature, they are not sufficient to define the spatio-temporal aspect of a video clip. In comparision, *storyboard sketches* is a convenient yet powerful query mechanism for video retrieval to describe the spatio-temporal feature of the object/scene. In Fig.5 we show two typical query sketches used in our system, and the top 5 returned results. From this graph we can see that our system is able to return video clips that share spatio-temporal and semantic similarity.

The average Precision-Recall (P-R) curves are shown in Fig.3, this demonstrates 3 different PR curves the 'motion alone','semantic alone' and the 'fused features' results. For the motion PR curve five different motion strokes depicting unique directions of motions for the foreground objects are evaluated. The semantic performance is evaluated using 2 different semantic queries: 'horse' depicting the foreground object; and 'man' depicting the foreground together with 'snow' describe the background (the object class 'man' appears in all video clips, the background class 'snow' is used to discriminant from horse riding clips). We manually create groundtruth for each query by considering the related clips based on motion or semantics alone separately. We do not draw the precision recall curve of using shape and color feature alone, since without the support of motion, semantics or foreground probability; these features could easily match to background areas (Second row of Fig.4 – the red query retrieved the background).

The P-R curve 'fused features' in Fig.3 considers all 4 features combined as explained in section 5. In order to design reasonable set of queries to evaluate the proposed system, we use the groundtruth to select suitable candidates that have a combined direction, semantic class and colour of greater than 10 examples within the dataset. In total 7 free-hand sketch queries are used to evaluate our system. The 7 queries cover 4 motion directions, 7 colours and 2 semantic classes to demonstrate the range of the dataset. On average each

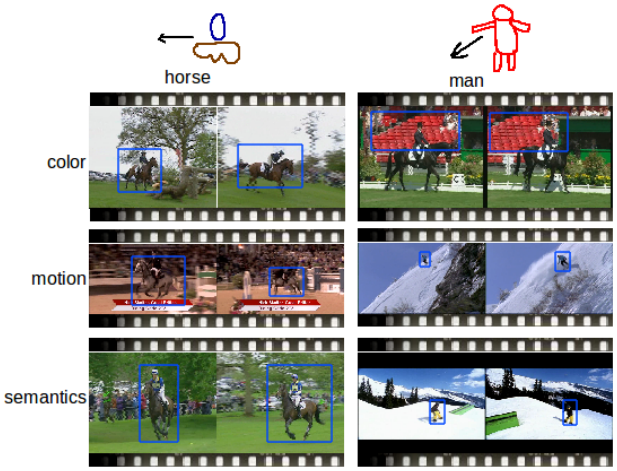

Figure 4: Example query sketches and their example results of using color, motion, semantics features alone individually. The blue bounding box indicate the area of interest that matched with the query sketch described by the according feature.

query has 20 related video clips. A clip is considered relevant with the query sketch when it shares approximate shape, color, motion and semantics to the sketched foreground object (and background if sketched).The P-R curve for 'fused features' shows that our system achieves a comparable performance with that of the VideoQ [15] and [3]. Note that the performance of the curves in Fig. 3 are not comparable to each other, since the groundtruth of each are created by considering different aspects of the video.

Overall the dataset we obtain MAP=0.48 by considering all the features, the top 10 results have ∼=57% relevance. This performance is comparable to the performance achieved in VideoQ [15] and [3]. We also evaluate our system without using the shape component. The system achieves a MAP=0.45, and the top 10 results have ∼=53% relevance. This shows that although using shape feature alone is difficult to achieve satisfied retrieval result, our proposed method to incorporate the shape feature is efficient yet effective to improve the retrieval performance.

At run time, once the user submit a query sketch to the system, it takes on average around 53.42 seconds to rank the 500 video clips in the dataset. This improves the run time computational complexity in [3], which takes 2 minutes to ranks videos in a similar but smaller dataset.

7. CONCLUSION

We have presented a sketch based video retrieval (SBVR) system driven by free-hand sketches depicting object appearance and motion, and annotated with keywords to indicate semantics. To the best of our knowledge we are the first to combine shape, motion, colour and semantics within a single SBVR framework. Furthermore we have introduced the use of Markov Random Fields (MRFs), more commonly used for video segmentation, as a novel form of SBVR solution capable of both ranking clips for relevance, and localising sketched objects within retrieved clips. We have demonstrated good accuracy over a challenging dataset of 500 sports footage clips.

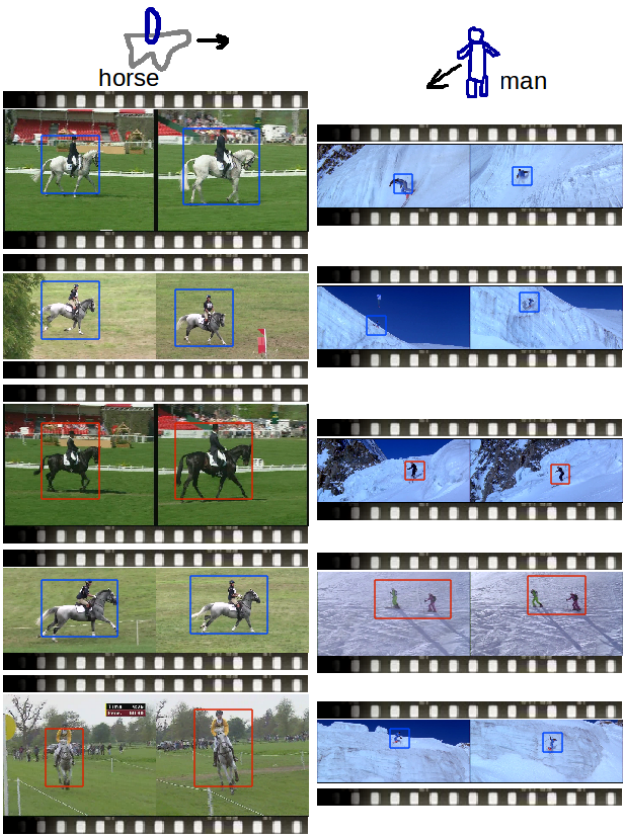

Figure 5: Example query sketches and the top 5 returned results, ranked from top to bottom. The red bounding box indicate the negative results, and the blue bounding box indicate the positive results.

In adopting an MRF optimization over all videos, we follow recent SBVR approaches [3] that tackle ambiguity in sketch by phrasing the retrieval task as a model fitting problem. Here we extract a multi-modal representation of the object to be retrieved, and solve for the resulting MRF to compute the most likely supervoxel labelling given that model. Whilst this solution offers unique advantages in seeking a globally optimal labelling for a given sketch, it is open to two potential criticisms.

First, the run-time expense of performing an MRF solve for each video. We have addressed this by adopting a supervoxel representation that contains a relatively low (around 2k) number of nodes, and we show that it can be solved fairly efficiently. This is a significant reduction in complexity over [3] where set of per-frames superpixel (not per-clip supervoxels) are labelled at query time.

Second, the system might at first consideration be deemed unsatisfactory due to perceived sensitivity to the weights within our unary term that combine our various multi-modal features. In fact we regard this as a strength; users will frequently wish to express a preference between modalities. Consider a user sketching a red car in the absence of such a clip in the dataset. Would they prefer a set of results containing red objects, or car shaped objects, or red objects moving in the direction sketched? The balance between these modalities is user task specific, and in future an ideal

candidate interactive specification through a relevance feedback interface. We believe this is the most promising future direction for our system.

8. REFERENCES

[1] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV. Volume 2. (2003) 1470–1477

[2] Bertini, M., Del Bimbo, A., Nunziati, W.: Video clip matching using mpeg-7 descriptors and edit distance. In: CIVR. (2006) 133–142

[3] Collomosse, J., McNeill, G., Qian, Y.: Storyboard sketches for content based video retrieval. In: ICCV. (2009) 245–252

[4] Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV. (2001) 105–112

[5] Ashley, J., Flickner, M., Hafner, J.L., Lee, D., Niblack, W., Petkovic, D.: The query by image content (qbic) system. In: SIGMOD. (1995) 475

[6] Smith, J.R., Chang, S.F.: Visualseek: A fully automated content-based image query system. In: ACM Multimedia. (1996) 87–98

[7] Sciascio, E.D., Mingolla, G., Mongiello, M.: Content-based image retrieval over the web using query by sketch and relevance feedback. In: Proceedings of the Third International Conference on Visual Information and Information Systems. VISUAL (1999) 123–130

[8] Jacobs, C.E., Finkelstein, A., Salesin, D.: Fast multiresolution image querying. In: SIGGRAPH. (1995) 277–286

[9] Hu, R., Barnard, M., Collomosse, J.P.: Gradient field descriptor for sketch based retrieval and localization. In: ICIP. (2010) 1025–1028

[10] Hu, R., Wang, T., Collomosse, J.P.: A bag-of-regions approach to sketch-based image retrieval. In: ICIP. (2011) 3661–3664

[11] Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: Sketch-based image retrieval: Benchmark and bag-of-features descriptors. IEEE Transactions on Visualization and Computer Graphics **17** (2011) 1624–1636

[12] Cao, Y., Wang, C., Zhang, L., Zhang, L.: Edgel index for large-scale sketch-based image search. In: CVPR. (2011) 761–768

[13] Shim, C.B., Chang, J.W.: Efficient similar trajectory-based retrieval for moving objects in video databases. In: CIVR. (2003) 163–173

[14] Su, C.W., Liao, H.Y.M., Tyan, H.R., Lin, C.W., Chen, D.Y., Fan, K.C.: Motion flow-based video retrieval. IEEE Transactions on Multimedia **9** (2007) 1193–1201

[15] fu Chang, S., Chen, W., Meng, H.J., Sundaram, H., Zhong, D.: Videoq: An automated content based video search system using visual cues. In: Proceedings of ACM Multimedia. (1997) 313–324

[16] Hu, R., Barnard, M., Collomosse, J.: Gradient field descriptor for sketch based retrieval and localization. In: ICIP. (2010) 1025–1028

[17] Hu, R., Collomosse, J.P.: Motion-sketch based video retrieval using a trellis levenshtein distance. In: ICPR. (2010) 121–124

[18] Hu, R., James, S., Collomosse, J.P.: Annotated free-hand sketches for video retrieval using object semantics and motion. In: MMM. (2012) 473–484

[19] Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH. (2004) 309–314

[20] Li, Y., Sun, J., Shum, H.Y.: Video object cut and paste. ACM Transactions on Graphics **24** (2005) 595–600

[21] Wang, T., Collomosse, J.P.: Probabilistic motion diffusion of labeling priors for coherent video segmentation. IEEE Transactions on Multimedia **14** (2012) 389–400

[22] Kohli, P., Ladicky, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision **82** (2009) 302–324

[23] Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR. (2010) 2141–2148

[24] Collomosse, J.P., McNeill, G., Watts, L.A.: Free-hand sketch grouping for video retrieval. In: ICPR. (2008) 1–4

[25] Csurka, G., Perronnin, F.: An efficient approach to semantic segmentation. International Journal of Computer Vision **95** (2011) 198–212

[26] Hu, R., Larlus, D., Csurka, G.: On the use of regions for semantic image segmentation. In: Indian Conference on Vision Graphics and Image Processing. (2012)

[27] Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: CVPR. (2012) 614–621

[28] Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. (2008) 1–8

[29] Yang, B., Nevatia, R.: An online learned crf model for multi-target tracking. In: CVPR. (2012) 2034–2041

[30] Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transaction on Pattern Analysis and Machine Intelligence **26** (2004) 1124–1137

[31] Alahari, K., Kohli, P., Torr, P.H.S.: Reduce, reuse & recycle: Efficiently solving multi-label mrfs. In: CVPR. (2008)