Storyboard sketches for content based video retrieval

J. P. Collomosse, G. McNeill and Y. Qian
Centre for Vision, Speech and Signal Processing,
University of Surrey
Guildford, UK
{J.Collomosse, Y.Qian} @ surrey.ac.uk

Abstract

We present a novel Content Based Video Retrieval (CBVR) system, driven by free-hand sketch queries depicting both objects and their movement (via dynamic cues; streak-lines and arrows). Our main contribution is a probabilistic model of video clips (based on Linear Dynamical Systems), leading to an algorithm for matching descriptions of sketched objects to video. We demonstrate our model fitting to clips under static and moving camera conditions, exhibiting linear and oscillatory motion. We evaluate retrieval on two real video data sets, and on a video data set exhibiting controlled variation in shape, color, motion and clutter.

1. Introduction

This paper presents a Content Based Video Retrieval (CBVR) system, capable of retrieving video clips using free-hand sketched queries. Our query sketches depict both the *scene content* and *dynamics* present in the clip, using motion cues (arrows and streak-lines) commonly used in production story-boarding [6, 10]. We refer to this input medium as a *storyboard sketch* (after [5], e.g. Fig. 1).

When people recall events, such as those in video, they draw upon their *episodic memory* [22]. Crucially, such recollections are not appearance based (photo-accurate); rather they are a meaningful, ordered account of an episode documenting the relative positions and actions of a few key actors/objects. We argue that *storyboard sketches* are well suited for specifying descriptions of episodes to a CBVR system. The semantic gap created by ambiguities both in episodic recall, and in sketch depiction, poses a challenging matching problem for Computer Vision (Sec. 2).

Our contribution is a novel algorithm for matching sketched object descriptions to video, combining spatial and weak dynamic constraints to identify and rank relevant clips. We propose a probabilistic, autoregressive model of a video clip, based on parallel Linear Dynamical Systems (LDSs), that each encode the shape, color and motion parameters of a sketched object. We explain clips through this model, which simultaneously labels video regions to sketched objects and evaluates support for the sketch in a given clip (Sec. 4). We have evaluated our algorithm on both real and synthetic video data sets in Sec. 5.

1.1. Related Work

Keyword tagging can be labor intensive and, when performed collaboratively, leads to descriptive inconsistencies [8]. Querying by *visual example* (QVE) provides an attractive alternative. Recent successes with 'bag of words' approaches have lead to systems able to quickly locate objects within movie-length data sets [20], or large image collections [4] from a *photographic* query.

Much of the *sketch based* retrieval (SBR) literature focuses on image retrieval [8]. Queries typically comprise blobs of color or predefined texture [11, 15]. Color and texture information is augmented with shape descriptors in [18], and spectral descriptors such as wavelets have also been explored [12]. These techniques have been trivially extended to video through key-frame extraction, however as with [20, 4], such systems are *appearance based*. There is no temporal component to the query, and a high level of realism is expected in the sketch. The nature of episodic recall suggests this expectation to be unrealistic in sketched queries [5].

Although several motion-based video retrieval systems have been proposed (e.g. for activity recognition [13, 14]), only a handful of systems explicitly query on sketched motion. Most match sketched trajectories with optical flow fields in the video [19, 21]. However, flow-based approaches model neither camera motion, nor the spatial structure in a scene. By contrast, Chang et al.'s VideoQ [2] adopts an approach closely related to our own; segmenting the video frames into regions and matching on both spatial properties (color and shape) and motion at the region level. However our system differs from VideoQ, and other SBR

systems, in a number of ways.

Existing SBR systems require sketches to indicate the precise trajectory of objects. VideoQ also requires users to specify the object's speed (in pixels/second). However, a recent user study [5] found that sketches drawn for CBVR recall are often imprecise with respect to both appearance and motion depiction (Sec. 2.1). There is usually no indication of speed, and only a few salient objects are sketched. Furthermore, objects co-present in a sketch often appear at different instants in a clip. In order to bridge the semantic gap between free-hand sketch query and a given retrieval target, we propose a weaker model of space and motion (Sec. 4.1) that can both accommodate these ambiguities in sketch, and match broader motion classes than previous systems (both monotonic trajectories and oscillatory motion). In addition, our model enables multiple regions to be aggregated and labelled to a single sketched object (Sec. 3). We do not assume that the video is perfectly segmented into semantically meaningful regions as in VideoQ. Indeed, we aim for an over-segmentation of the video and later aggregate regions under our probabilistic model of objects.

2. Overview of Sketch Parsing

Our system accepts query sketches as temporally ordered lists of strokes (trajectories, with associated attributes e.g. color). Whilst sketching, we ask users to indicate whether strokes form the foreground, background or a motion cue in their drawings (e.g. Fig 1).

Given a query, we apply our existing motion-sketch parsing algorithm [5] to group strokes into sketched objects. We then perform feature extraction to obtain a description of each object depicted in the sketch. These *object descriptions* form the input to our CBVR algorithm (Sec. 4).

2.1. Descriptions of Sketched Objects

Our motion sketch parsing algorithm was motivated by an earlier user study exploring motion-sketches under episodic recall. We observed that users sketch using a combination of coarse shape approximations and a consistent, shared alphabet of pictograms for depicting common objects (e.g. stick-men) or motion (e.g. trailing streak-lines or leading arrows). Objects in motion are depicted as moving relative to a static background; regardless of the motion of the camera/viewpoint. Crucially, only direction is indicated in a sketch; there is no indication of speed or other motion parameters. Sketches are also under-specified spatially; only a few salient objects are depicted per query.

Our two-step sketch parsing process [5] comprises: HMM based recognition of common pictogram objects (stick-men, arrows, streak-lines) followed by grouping of remaining non-pictogram strokes into objects using graph-cut. From each non-motion cue object grouped by [5], we

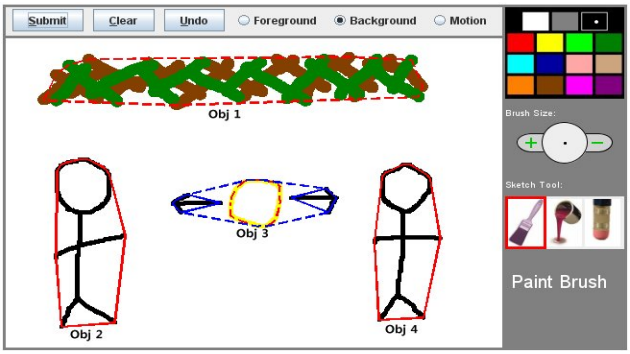

Figure 1. A query sketch grouped by [5] and processed into descriptions of moving objects. Non-motion pictograms identified in solid red, motion pictograms in solid blue, remaining strokes grouped in dashed red. Motion cues are assigned to objects based on proximity (dashed blue).

extract a number of features; the "*object description*":

1. **Color** distribution (Gaussian Mixture Model (GMM) of colors in the sketched object, in CIELab space).
2. Global **shape** descriptors (Eccentricity, Orientation, Convexity, Area) derived from the object boundary; plus a measure of Connectedness from object mask.
3. **Foreground** probability (ratio of foreground to background strokes in object).
4. Probability of being a **person** (derived from the pictogram recognition algorithm [5]).
5. **Direction** of the object's motion

Feature (5) is derived from motion cues present in the sketch. Motion cues are associated with the object according to their spatio-temporal proximity. Multiple motion cues can be associated to an object, to indicate oscillation.

In this way, each query sketch yields one or more *object descriptions*, that we subsequently match to video (Sec. 4).

3. Video Pre-processing

When videos are added to our database, pre-processing steps prepare the video for query matching (Sec. 4). First, videos are divided into clips using cut-detection [23] to identify scene transitions. Within each clip, we compute the homography between adjacent frames, obtaining an estimate of global motion. This information is used in Sec. 4.1 to compensate for camera motion (Fig. 2e). Users tend to subtract camera motion in their sketches (Sec. 2.1), and we must do the same to enable comparison.

We segment each video frame independently into a series of regions using mean-shift [3], under an assumption of color homogeneity within regions (Fig. 2b). Attributes such

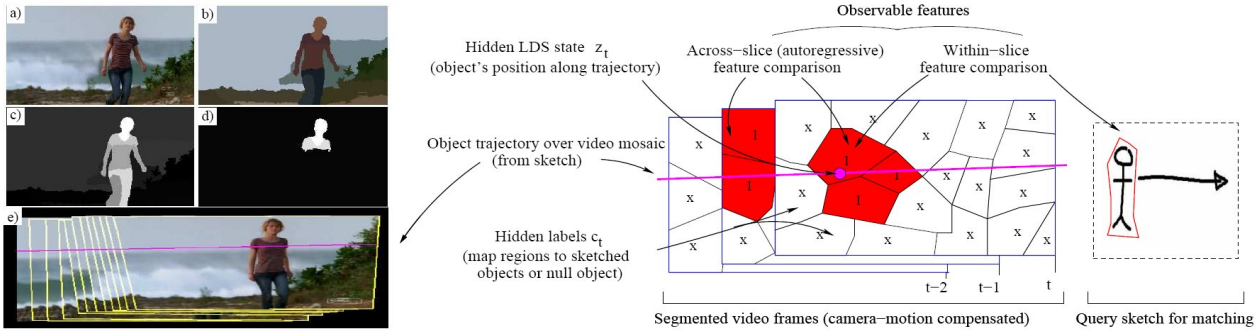

Figure 2. *Left: Pre-processing clips*. Frames (a) are segmented into colored regions (b). Region properties are computed: area, color, and probabilities of being foreground (c) or a person (d). Camera motion is estimated by inter-frame homography (e). *Right: Matching a sketch to clip*. The sketched direction of the object is extended over a clip's camera-motion compensated frames, forming a trajectory (magenta). We model an object's progress along its trajectory as transitions along states in a LDS[1]. Fitting our model (Sec. 4.1) yields a labelling of clip regions to sketched objects (hidden labels $\mathbf{c_t}$), and object positions (hidden states $\mathbf{z_t^u}$), for all frames $t = [1, T]$ and objects $u = [1, U]$.

as color, area, and centroid are computed for each region. The VGG upper-body detector [7] is also applied yielding a likelihood that each region belongs to a person (Fig. 2d). Moving from a pixel-based representation to a region-based (i.e. super-pixel [16]) representation in this way greatly improves the efficiency of our matching algorithm – typically we move from 100,000 pixels to $\sim$ 50 regions. However, note that we are not attempting to carry out true object segmentation at this stage (as in [2]). Our region segmentation is finer than the coarse object level segmentation corresponding to a query sketch and hence, multiple regions are typically later aggregated to form a single object.

Finally, we compute the probability of each region being in the 'foreground' of the scene. For each frame we apply the inter-frame homographies computed earlier to construct a mosaic from adjacent frames. The current frame is differenced with the mosaic to generate a 'foreground' map; which is averaged and normalized for each region (Fig. 2c).

4. Matching and Retrieving a Clip

The sketch parsing process (Sec. 2.1) outputs a compact description of the desired clip based on objects and their motion. Our retrieval strategy is to *explain* each clip using that description, ranking clips according to the *evidence* present in the clip for a given sketch. We refer to the process of explaining a clip in this way as 'matching'.

We match a sketch to a clip by first extending the motion direction of each object to form a trajectory over the mosaiced (camera motion compensated) video frames within the clip (Fig. 2, right). We then look for evidence supporting the object in video regions along that trajectory. We 'match' each region of each frame to one object such that objects and trajectories arising from this assignment most closely resemble those in the sketch. A 'null object' is included for regions that belong to undrawn objects; this allows to select only the regions which combine to give the best approximation to the sketched objects – i.e. the sketch need *not* describe all the regions of a frame.

Sketched objects and trajectories are only rough approximations to their counterparts in a clip and many of the objects contained within the clip may not be present in the sketch (Sec 2). Given the approximate and incomplete nature of sketches, and that the desired region assignment is unobserved, it is natural to formulate our 'matching algorithm' in a probabilistic setting. We now formalize this approach using a generative model for video clips; sketched object features and trajectories are the model parameters.

4.1. The Probabilistic Model

We first introduce the notation used in our model. Time is indexed by (subscript) t and we assume a clip to comprise T frames. Objects are indexed by (superscript) u, where U is taken to be the total number of objects identified in the sketch. Each object u is associated with a Linear Dynamical System (LDS) whose state corresponds to the point the object is at on its trajectory; one can think of an object 'moving along' its trajectory as the LDS transitions through its states (Fig. 2). A stationary object is represented by a single state LDS. The unobserved states of all the LDSs at time t are stored in the vector $\mathbf{z_t} = (z_t^1, ..., z_t^U)$ – so if object $u = 1$ is 1.74 units along its trajectory at time $t = 9$, then $z_9^1 = 1.74$.

Frame t is denoted by $\mathbf{X}_t$ and for each of these T images, we partitioned pixels into N_t regions during video pre-processing (Sec. 3). For each frame, we introduce an unobserved label vector $\mathbf{c_t} = (c_{t,1}, ..., c_{t,N_t})$, where $c_{t,n} \in \{1, ..., U, U+1 = \text{'null object'}\}$ indicates which object region n of frame t belongs to – e.g. if region $n = 6$ at time $t = 4$ belongs to object $u = 2$, then $c_{4,6} = 2$.

For brevity, we sometimes use the notation $\mathbf{X}, \mathbf{Z}, \mathbf{C}$ to denote all the observed frames, hidden states and hidden labels respectively.

The graphical model for our approach is shown in Fig. 3; from this, we see that the joint distribution is given by:

$$p(\mathbf{X},\mathbf{Z},\mathbf{C}) = \prod_{t=1}^{N} p(\mathbf{c}_t) \times \prod_{u'=1}^{U} p(z_q^{u'}) \prod_{t=2}^{T} p(z_t^{u'}|z_{t-1}^{u'}) \times p(\mathbf{X}_1|\mathbf{z}_1,\mathbf{c}_1)p(\mathbf{X}_2|\mathbf{c}_1,\mathbf{X}_1,\mathbf{z}_2,\mathbf{c}_2) \times \prod_{t=3}^{T} p(\mathbf{X}_t|\mathbf{c}_{t-2},\mathbf{X}_{t-2},\mathbf{c}_{t-1},\mathbf{X}_{t-1},\mathbf{z}_t,\mathbf{c}_t). \quad (1)$$

Let us consider each of the distributions in turn. Though it is not indicated in Fig. 3, the priors on the region labels within a single slice are assumed to be independent: $p(\mathbf{c_t}) = \prod_{n=1}^{N_t} p(c_{t,n})$. We have used a uniform distribution $p(c_{t,n} = u) = 1/(U+1)$ to reflect our lack of prior knowledge about these label values. Recall that the video represents the data and that the same model (i.e sketch) will be used to analyze many clips. Thus, it would not be valid to derive values for the $p(c_{t,n})$ from a video.

4.1.1 Emission Distribution (Appearance)

The *emission distribution* for the observed data – the bottom two lines of eq.(1) – is defined so that the objects associated with the current labeling should resemble both the corresponding objects in the sketch *and* the corresponding objects at the previous time step – i.e. there is an *autoregressive* component. An object's appearance may change during a clip and hence, it may not resemble even an accurately drawn model object for the full duration of the clip. The autoregressive structure of the model encourages temporal coherence in the object labeling and hence, provides a degree of robustness to this problem. Formally, we combine the two components of the distribution using an 'across slice' (or frame-to-frame) term q_A and a 'within slice' (or sketch-to frame) term q_W (refer to Fig. 2, right):

$$p(\mathbf{X}_t|\mathbf{c}_{t-2},\mathbf{X}_{t-2},\mathbf{c}_{t-1},\mathbf{X}_{t-1},\mathbf{z}_t,\mathbf{c}_t) \propto q_w(\mathbf{X}_t|\mathbf{z}_t,\mathbf{c}_t) q_A(\mathbf{X}_t|\mathbf{c}_{t-2},\mathbf{X}_{t-2},\mathbf{c}_{t-1},\mathbf{X}_{t-1},\mathbf{c}_t). \quad (2)$$

The expressions for q_W and q_A incorporate the individual object comparisons[1]. Recall from Sec. 2 that we represent sketched objects by a set of features (shape descriptors, color distribution, foreground and 'person' probability). We also have a centroid for the object, derived from its LDS state. To compare a sketched object to a video object, we extract the same features from the putative video object associated with the labelling $\mathbf{c}_t$ i.e. we group together all the regions that are currently assigned to the object and compute the features for the aggregate region. These features are computed as follows: the area, centroid and global shape descriptors are computed from the binary mask formed by the putative object-to-region labelling, the object foreground score is the average score over the regions belonging to the object (see Sec. 3), the person score is defined as the maximum person (i.e. upper-body) score of any region in the object and the color GMM has a component for each region with a mean set to the color of the region and mixing coefficient given by the relative area of the region to the full object. We refer to the feature vector for object u in frame t given the current value of the label $\mathbf{c}_t$ as $\mathbf{f}_t^u$. Similarly, we write $\tilde{\mathbf{f}_t^u}$ for the corresponding feature vector of the sketched/model object given the state z_t^u. Note that only the centroid feature of $\tilde{\mathbf{f}_t^u}$ changes with time; the others are fixed after sketch parsing. The centroid is updated at each time step (i.e. with z_t^u) to reflect expected position of the sketched object along its trajectory. We correct for camera motion by transforming the region centroids by the inter-frame homography computed during pre-processing (Sec. 3). We rely on the auto-regressive component of our model (eq. 4) for robustness to changing object features over time, which we assume to be negligible between frames. Although our model has no explicit spatial coherency term, we find our shape features (e.g. connectedness) and optimization strategy (Sec. 4.2) to adequately discourage non-contiguous region to object labellings in $\mathbf{c}_t$.

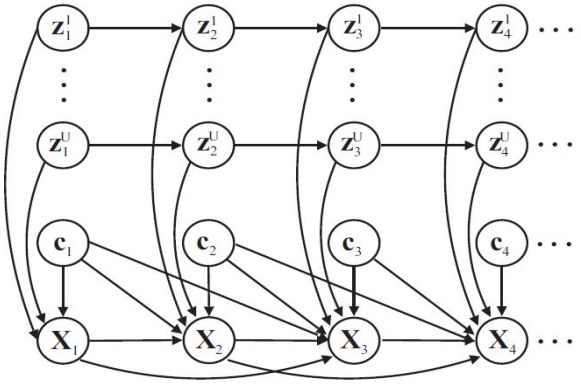

Figure 3. Illustrating the dependency structure of our probabilistic model for a video clip — Sec. 4.1.

Having placed sketched and video objects in a common feature space we define q_W and q_A from eq.(1) as:

$$q_W(\mathbf{X}_t|\mathbf{z}_t,\mathbf{c}_t) \equiv \prod_u \mathcal{N}(\mathbf{f}_t^u;\tilde{\mathbf{f}_t^u},\Sigma)^{A_u} \quad (3)$$

$$q_A(\mathbf{X}_t|\mathbf{c}_{t-2},\mathbf{X}_{t-2},\mathbf{c}_{t-1},\mathbf{X}_{t-1},\mathbf{c}_t) \equiv \prod_u \mathcal{N}(\mathbf{f}_t^u;\mathbf{f}_{t-1}^u,\rho\Sigma)^{A_u} \quad (4)$$

[1] At $t = 1$ we use only q_W and at $t = 2$, q_A is necessarily only dependent on the previous time-step.

where A^u is the area of object u, $\mathcal{N}$ denotes the Gaussian distribution and Σ is a diagonal covariance matrix for the object features. In other words, we compare objects (both sketch-to-frame and frame-to-frame) using a Gaussian in feature space under the assumption that the features are uncorrelated. The scalar ρ in eq.(4) weights the importance of the frame-to-frame contribution relative to the sketch-to-frame contribution. In other words, ρ indicates whether the model object or previously observed object will better predict the appearance and position of the current object. The above expressions for q_A and q_W are not precise in the following ways: Firstly, the mean centroid in q_A is equal to the object centroid at $t-1$ (i.e. the relevant feature of $\mathbf{f}_{t-1}$ as expected) *plus* the displacement vector from the centroid at $t-2$ to $t-1$; this is what makes the bottom chain in Fig. 3 second order. Secondly, since each object's color distribution is described by a GMM, we cannot use the Euclidean distance to compare these features when evaluating the Gaussian. We use a measure similar to the approximation of Kullback-Leibler based on the unscented transform [9] that is fast to compute and suitable here as the GMMs are effectively describing color frequency counts at the mean of each component – the covariances of the GMMs reflect how accurately we expect colors to match rather than summarizing color spatial distribution. Given two GMMs $\mathcal{G}_1, \mathcal{G}_2$ with density functions $g_1(x)$, $g_2(x)$, means $\mu_{1,1..G_1}$, $\mu_{2,1..G_2}$ and mixing coefficients $\pi_{1,1..G_1}, \pi_{2,1..G_2}$, we define the symmetric distance between them as:

$$D(\mathcal{G}_1,\mathcal{G}_2)=\left(\prod_{j=1}^{G_2} g_1(\mu_{2,j})^{\pi_{2,j}}\right)^{\frac{1}{2}}\left(\prod_{k=1}^{G_1} g_2(\mu_{1,k})^{\pi_{1,k}}\right)^{\frac{1}{2}}. \quad (5)$$

i.e. we evaluate the means of one model under the density of the other using mixing coefficients to weight the terms.

4.1.2 Transition Distribution (Motion)

For object u, the *initial distribution* $p(\mathbf{z}_1^u)$ and *transition distribution* $p(\mathbf{z}_t^u|\mathbf{z}_{t-1}^u)$ encapsulate our belief about how the object moves along its trajectory. Since motion cues typically only indicate the direction of motion we are forced to make assumptions about the speed and extent of an object's motion. We stretch the motion trajectory across the full extent of the mosaic (Sec.3) and assume that the object travels across this trajectory at uniform speed. When coupled with a large transition variance (σ^2 in eq.(6)), this seemingly strict assumption enables the model to lock on to objects that move monotonically along some part of the trajectory (Fig. 6, 8). The transition distribution is given by:

$$\mathbf{z}_t^u|\mathbf{z}_{t-1}^u \sim \mathcal{N}(\mathbf{z}_{t-1}+\lambda\hat{\mathbf{v}}_u, \sigma^2). \quad (6)$$

where $\hat{\mathbf{v}}_u$ is the unit vector in the direction of motion and $\lambda \equiv T/(\text{trajectorylength})$. Although this definition makes the model dependent on the clip, it enables us to account for different clip lengths and mosaic sizes and once eq.(1) is normalized by T, the probability of different clips for a given query sketch are comparable (for clip ranking).

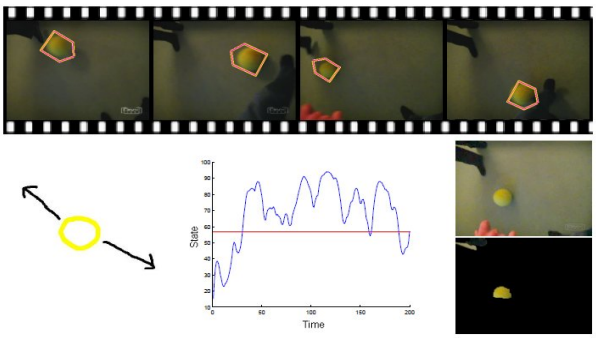

Figure 4. Oscillating objects (drawn with two motion cues) are handled by our model when $\lambda = 0$ (eq. 6). LDS states $\mathbf{z}_t$ plotted before (red) and after (blue) inference; and region-object labelling before (upper) and after (lower) inference (null-object in black).

Note that the transition distribution only relates to the movement of the object *along* the model trajectory; given any $\mathbf{z}_t$, the actual position of the object is generated from the emission distribution for the object centroid (i.e. regions must lie close to, but not necessarily on the trajectory). Setting $\lambda = 0$ predicts that the object remains motionless, but allows for smooth motion in either direction – Fig. 4.

The number of objects in the model, the objects' trajectories, and the objects' features are computed during sketch parsing (Sec. 2). The sketched trajectory of each object u determines direction $\hat{\mathbf{v}}_\mathbf{u}$; for oscillating motion we fix λ to 0 to model smooth variation along the trajectory.

Ideally, the remaining model parameters – Σ, ρ and the starting and transition distributions – should be estimated from a training set of sketch-clip pairs; currently values are set empirically. Note that we do not wish to learn different parameters for each clip since the model parameters encapsulate our sketched description of the desired clip, and our prior belief about sketch accuracy. For example, the variance parameter for the centroid (an element of Σ) specifies our belief about what distance constitutes sketch inaccuracy in position vs. genuine difference in object position.

4.2. Inferring Values of the Hidden Variables

Having defined the model, our objective is to compute the $p(\text{clip}|\text{sketch}) = p(\mathbf{X}|\text{modelparameters})$. We are also interested in the values of the hidden states $\mathbf{z}_t$ and labels $\mathbf{c}_t$ as these enable us to visually assess how a clip has been 'interpreted' with respect to a given sketch (see last two columns of Fig. 8). These hidden variables are estimated using iterated conditional modes (ICM) [1] as follows.

Given putative values of $\mathbf{c}_t$, our model decomposes into $U+1$ independent autoregressive LDSs. Since q_A eqs.(2,4))

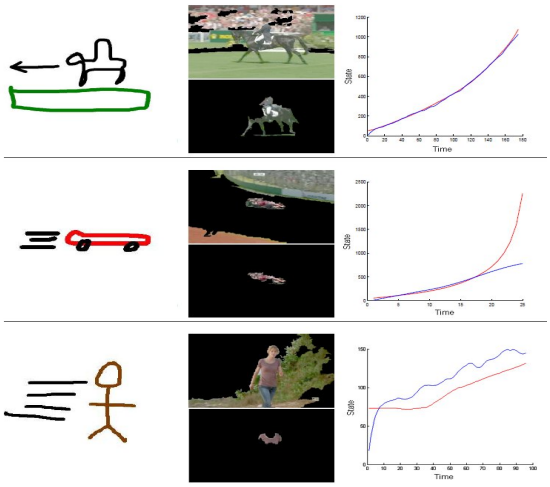

Figure 5. Left: Three queries over *TSF*. Middle: Region to object labelling for best clip, before (upper) and after (lower) inference. Right: LDS state $\mathbf{z}_t$, before (red) and after (blue) inference.

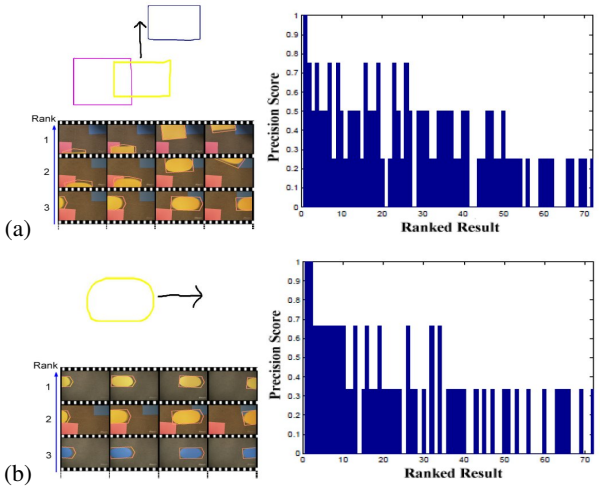

Figure 6. Synthetic (*SYN*) database; 72 clips. Left: Sketched queries with (a) and without (b) background, and the top 3 resulting clips for each. Right: Precision of returned clips — from most to least relevant; higher precision scores cluster in the top ranks.

does not depend on the $\mathbf{z}_t$, the autoregressive component can be ignored when selecting the $\mathbf{z}_t$ and each chain reduces to a simple first order LDS. Thus, our individual object models (i.e. given the region labels $\mathbf{C}$) are similar to Kalman trackers and just as in tracking, we intuitively think of the models 'locking on' to objects. However, here we have observations *for all* T *frames*, allowing us to use the forward and backward Kalman equations when updating the hidden states: $\mathbf{z}_t \leftarrow \text{argmax}_{\mathbf{z}_t} p(\mathbf{z}_t|\mathbf{X}, \mathbf{C})$.

At initialization, the hidden states are chosen so that the object traverses its entire trajectory at a speed proportional to camera motion. This is obtained from camera motion compensated centers of all frames, projected onto the trajectory. The hidden labels are then initialized to objects using the priors and centroid information only.

Given values for the hidden states $\mathbf{Z}$ and all the hidden labels except $\mathbf{c}_n$ (written $\mathbf{C}\backslash\mathbf{c}_t^n$), we update $\mathbf{c}_t^n$ using: $\mathbf{c}_t^n \leftarrow \text{argmax}_{\mathbf{c}_t^n} p(\mathbf{c}_t^n|\mathbf{X}, \mathbf{Z}, \mathbf{C}\backslash\mathbf{c}_t^n)$; this is equivalent to selecting the $\mathbf{c}_t^n$ that maximizes the joint distribution (eq. 1). The joint likelihood of all the variables can be estimated at any point by evaluating eq.(1) with current estimates of $\mathbf{Z}$, $\mathbf{C}$.

Practically, we must update *multiple* labels per iteration to prevent ICM falling into local minima. We exhaustively consider all possible relabellings of a subset of r regions within $\mathbf{C}$, choosing the relabelling that maximizes the joint distribution. These r regions are chosen to form a spatially connected aggregate region incorporating $\mathbf{c}_t^n$. Selection of the r regions is stochastic, biased towards neighboring regions with closest color to $\mathbf{c}_t^n$. Each ICM iteration requires $(u+1)^r$ evaluations; r thus drives a trade-off between accuracy and speed. We found $r = 8$ worked well for our trials. Our algorithm converges on a $\mathbf{Z}$ and $\mathbf{C}$ that maximizes the joint distribution eq.(1). Convergence typically takes $10-20$ iterations; the ICM process terminates when improvements in $p(\mathbf{X}, \mathbf{Z}, \mathbf{C})$ fall below a threshold. We evaluate eq.(1) and normalize with respect to the number of frames T; enabling comparison between clips. This value is then used to rank the relevance of clips for a given query.

Fig. 4 shows our model fitting to oscillating motion. The transition graph shows $\mathbf{z}_t$ initialized to a single position (due to a static camera), but converging to an oscillating pattern under the 'smooth' ($\lambda = 0$) motion model. The region-object labels post-inference identify the oscillating object. Fig. 5 shows our model fitting to linear motion. As before, LDS states are initialized to positions closest to the frame center. For the horse, the transition graph shows the initial (blue) and inferred (red) $\mathbf{z}_t$ are near identical; the camera pans to track the horse. For the car, the camera initially tracks the car before it drives out of frame. The inferred $\mathbf{z}_t$ reflect this motion, leading to correct region labelling.

5. Evaluation and Discussion

We evaluated our CBVR system using three data sets: (i) *synthetic* video footage containing controlled lab cases (*SYN*); (ii) a *real* video subset of the public *KTH* data set [17]; (iii) a *real* video data set compiled from TV drama and sports footage (*TSF*). We chose our *TSF* set to resemble the VideoQ [2] test data set, which was not published[2].

5.1. Synthetic (Lab-based) Video Evaluation

The *SYN* data set contains 72 clips of filmed 2D shapes moving in the plane (Fig. 6). Clips comprise all combina-

[2]The SYN and TSF datasets used in this Section are available at `http://www.ee.surrey.ac.uk/CVSSP/VMRG/doodle/`

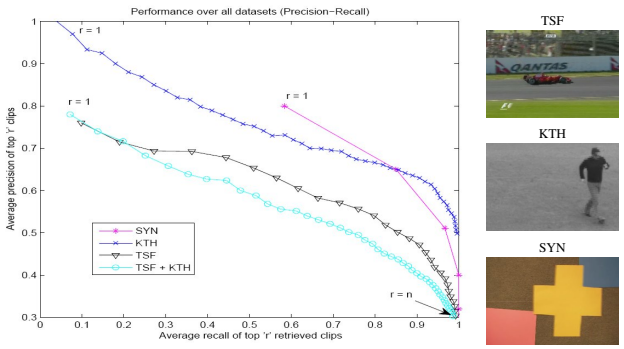

Figure 7. Precision-Recall for 4 data sets (*SYN*, *KTH*, *TSF*, *TSF+KTH*) averaged over q queries (*SYN* q=30, *KTH* q=64, *TSF* q=100, *TSF+KTH* q=100). Plotting r clips returned from set of n clips (*SYN* n=72, *KTH* n=200, *TSF* n=298, *TSF+KTH* n=498).

tions of 3 shapes, 4 colors, and 3 motion directions, with and without background clutter.

Fig. 6a contains a sample query sketch depicting a shape moving against a background, the top 3 clips retrieved, and the precision of each of the $r = [1, 72]$ ranked clips. A clip accumulates precision of 0.25 for each correctly matched variable; scores are thus distributed: $1 \times 1.0, 8 \times 0.75, 23 \times 0.5, 28 \times 0.25, 12 \times 0$. Average Precision (AP) is (cumulative precision of r clips/ maximum attainable cumulative precision at r), where $r = [1, 72]$. Averaging AP over range $r = [1, 72]$ yields a Mean Average Precision (MAP) of 0.91.

Our algorithm addresses sketch ambiguity by seeking evidence for sketched objects only; no LDS is created for unsketched objects (the 'null object') and so they are ignored. Thus the query of Fig. 6b equally supports clips of ovals moving left-right, with or without clutter. To account for this, we modified our precision score to consider color, shape and motion only; the score distribution is: $2 \times 1.0, 14 \times 0.67, 32 \times 0.33, 24 \times 0$. Again the most relevant clips (scoring 1.0) ranked highest; MAP was 0.85.

Fig. 7 plots Precision-Recall averaged over 30 queries (depicting 15 moving shapes with background, and the same 15 without). Here we use binary *precision* to make *SYN* comparable with *KTH* and *TSF*; an exact match across all relevant properties (motion, shape, etc.) is required for precision of 1. We obtained a high overall MAP (0.88) for *SYN*, representing an ideal to compare real video against.

5.2. Real Video Evaluation

We evaluated our system using a 200 clip subset of the *KTH* [17] activity data set. We selected 25 clips of running/walking people in various directions. The query set comprised sketches of stick-men with motion cues. Fig. 7 plots Precision-Recall averaged over all queries (MAP=0.74). Performance is comparable to *SYN* indicating good scalability, and correct aggregation of over-segmented

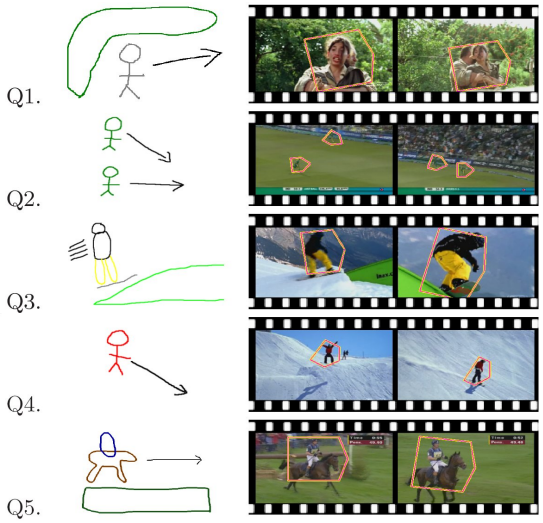

Figure 8. *TSF* data set; example sketch queries and best clip retrieved, discussed in Sec. 5.2. Average precision (AP) for queries: Q1=0.63; Q2=0.48; Q3=0.67; Q4=0.62; Q5=0.60.

regions to 'track' the person, and so correct discrimination of motion direction. However KTH contains only greyscale people and so does not fully exercise variability in all our features. We therefore evaluated a second data set *TSF* comprising 298 color TV drama/sports clips ($\sim$ 300 frames/clip). Objects were people, cars, or horses predominantly a single object with camera motion panning to follow. The data set is comparable to VideoQ who evaluate 200 similar clips over 4 queries. For each query we manually specify a ground-truth; a clip is relevant if, visually, the object/s shares approximate *shape* (aspect), *color*, and *direction* with the query (and *background* color if sketched). To test scalability we also ran this query set over the combined data sets *TSF+KTH*. Fig. 7 plots Precision-Recall curves averaged over the query set. Fig 8 illustrates sample queries each with corresponding best clip and AP score.

Overall for the *TSF* set we obtain MAP=0.65, and for the *TSF+KTH* set MAP=0.59. Interpreting Fig. 7 for *TSF* (298 clips), we expect the top 6 results to have on average $\geq\sim$ 70% relevance, which we regard as acceptable given our application of retrieving recalled episodes from video databases. For the combined set of 500 clips the top 6 results have $\geq\sim$ 65% relevance, suggesting good scalability to larger databases. Although *TSF* does not exactly match VideoQ [2], our results compare favorably with the AP of their 4 queries (0.40, 0.36, 0.55, 0.36 $\mapsto$ MAP=0.42)[3].

Fig 8 shows correct handling of linear motions over single (Q1,3-5) and multiple objects (Q2). In (Q2,3,5) the person detector fails due to scale; nevertheless the distinctive motion and color in these cases encourages correct recall. In all cases (Q1-5) camera motion is correctly compensated.

[3]Obtained from P-R curves of Fig.10 in Cheng *et al.* [2]

6. Conclusion

We have presented a probabilistic model for video clips based on Linear Dynamical Systems (LDS), and applied our model to match descriptions of sketched moving objects to video for CBVR. We have shown our model to correctly aggregate over-segmented video regions to form objects approximated by sketches. As such, our system does not assume temporally stable or semantically correct video pre-segmentation (as in *Chang et al.* [2]). We make further progress by matching novel motion types e.g. oscillation.

Although sketches are an expressive and intuitive query medium, they are also ambiguous. For example, motion cues have direction but do not reliably depict the magnitude of motion [5] (yet this information is required by [2]). This ambiguity forced us to introduce assumptions into our model; we assume the sketch canvas to approximate a mosaic spanning all frames, and that sketched motion extends across this. A further ambiguity is that objects may be sketched as co-present, but appear in the clip at different instants. Our model accounts for this, making no assumptions about objects' temporal relationships (Sec. 4). Furthermore, not all objects present in video must be sketched.

Sketches are unable to express the relative importance of features. In Fig. 6b three ovals are returned left-right. In rank 3, shape and motion seem more important than color, yet depending on our usage context this may not be appropriate. Future work will improve our implementation to interactive speeds and explore relevance feedback to interactively adjust the covariances on these features (eq.3). More complex features (e.g. for shape or person detection) might be substituted into our framework. We grounded our choices in an empirical study [5], that observed episodic sketches to contain only approximate color distributions and shape. Similarly, although our LDS accommodates any parametric path, complex motions are seldom sketched [5].

Rather than computing and then matching feature vectors from query and clip, we evaluate support within clips under a probabilistic model of content (our LDS framework). The main benefit is that components of the video are interpreted in the context of a sketch; ambiguities in the sketch are resolved in light of evidence within the video (much as one might realize that a child's drawing is of an elephant once told so). Given that unsupervised grouping of pixels into semantic objects eludes Vision, this seems a promising approach to bridging the semantic gap for SBR. Adaptations of our model might also be used for tracking.

Acknowledgements

This work is funded by EPSRC grant EP/D055032.

References

[1] J. Besag. On the statistical analysis of dirty pictures. *Jrnl. Royal Statistical Society B*, 48:259–302, 1986.

[2] S. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong. VideoQ: an automated content based video search system using visual cues. In *Proc. ACM Multimedia*, pp. 311–324, Nov. 1997.

[3] C. Christoudias, B. Georgescu, and P. Meer. Synergism in low-level vision. In *Proc. ICPR*, pp. 4:150–155, 2002.

[4] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with generative feature model for retrieval. In *Proc. ICCV*, pp. 1–8, 2007.

[5] J. Collomosse, G. McNeill, and L. Watts. Free-hand sketch grouping for video retrieval. In *Proc ICPR*, 2008.

[6] R. D. Dony, J. W. Mateer, J. A. Robinson, and M. G. Day. Iconic versus naturalistic motion cues in automated reverse storyboarding. In *Proc. CVMP*, pp. 17–25, 2005.

[7] V. Ferrari, M. Jimenez, A. Zisserman. Progressive search reduction for human pose estimation. *Proc. CVPR*, 2008.

[8] B. Furht and O. Marques. *Content-based Image and Video Retrieval*. Kluwer Acad., 2002. ISBN: 1-402-0700047.

[9] J. Goldberger and H. Aronowitz. A distance measure between GMMs based on the unscented transform and its application to speaker recognition. In *Proc. Eurospeech*, 2005.

[10] D. Goldman, B. Curless, D. Salesin, and S. Seitz. Schematic storyboards for video editing and visualization. In *Proc. ACM SIGGRAPH*, volume 25, pp. 862–871, 2006.

[11] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Effcient color histogram indexing for quadratic distance. *IEEE Trans. PAMI*, 17(7):729–736, 1995.

[12] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proc. ACM SIGGRAPH*, pp. 277–286, Aug. 1995.

[13] I. Laptev and P. Perez. Retrieving actions in movies. *Proc. ICCV*, pp. 432–439, 2007.

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[15] Z. Li, O. R. Zaane, and Z. Tauber. Illumination invariance and object model in image and video retrieval. *Jrnl. Vis. Comm. and Image Rep.*, 10(3):219–244, Sept. 1999.

[16] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. ICCV*, pp. 1:10–17, 2003.

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proc. CVPR*, 2004.

[18] E. D. Sciascio, G. Mingolla, and M. Mongiello. CBIR over the web using query by sketch and relevance feedback. In *Proc. Intl.Conf. VISUAL*, pp. 123–130, June 1999.

[19] C. Shim, J. Chang. Efficient similar trajectory retrieval for moving objects in video. *Proc. CIVR*, pp. 163–173, 2003.

[20] J. Sivic and A. Zisserman. A text retrieval approach to object matching in videos. In *Proc. ICCV*, pp. 1470–1477, 2003.

[21] C. Su, H. Liao, H. Tyan, C. Lin, D. Chen, and K. Fan. Motion flow-based video retrieval. *IEEE Trans. Multimedia*, 9(6):1193–1201, Oct. 2007.

[22] E. Tulving. *Elements of episodic memory*. Oxford Clarendon, 1983. ISBN: 0-198-521251.

[23] H. Zhang, Z. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. In *Proc. ACM Multimedia*, volume 1, pp. 10–28, June 1993.