1 Purpose

Image segmentation is the first stage of image processing in many practical computer vision systems. The development of static image segmentation algorithms has attracted considerable research interest and is enriched by a wide range of methodologies. However, work that has been published in the video analyses domain is still quite narrow and biased towards the sole use of motion characteristics. The recent proliferation of digital video archives and the advent in video analyses techniques has augmented the interest in the identification and tracking of physical objects within videos. High level semantic annotation of physical objects is the key to applications ranging from security and surveillance to information retrieval to sports and entertainment.

Goldman et al [1] present several applications that can be realised by using tracked 2D object motion. One application is the video-to-still composition where a video stream can be used to compose a single still image. Shots of different subjects, in different frames, appearing in different points in time can be combined into a single still image using a drag and drop approach. Another application of object tracking which drew considerable interest in the sports domain is the analyses of player and ball movement in soccer games [2]. The motion trajectories of the ball and players across the field is essential for the analysis of matches and tactics.

The need for metadata describing high level components in the video, such as objects and motion trajectories, is common to a wide range of applications, so the methodology adopted in this project will be applicable across different domains. Having said that, the required format and type of metadata might vary across applications. Thus in this project we will focus on fulfilling the needs of a rich media application interface which will be capable of incorporating visual tagging for authoring rich media such as hyperlinked videos.

2 Hypothesis

Our hypothesis is that object detection and tracking in existing video segmentation algorithms can be improved by combining techniques used in point tracking algorithms with features used in static image segmentation. This approach will serve for better identification of moving objects and for possible identification of stationary ones.

In this report we will propose a methodology and a project plan for evaluating our hypothesis. In the next section we will give a general overview of different approaches to object tracking, which will be a key element in our methodology. In section 4 we will present a methodology for video segmentation and object identification that is based on features from point tracking algorithms and features used in static image segmentation. In section 5, we will proceed by discussing an evaluation plan for assessing the performance of our methodology. Section 6 presents a high level project plan for developing the methodology.

3 Background

A key component in many video analyses applications is object representation and tracking. Before a subject can be tracked, it must be represented by some model. Object modelling is usually directed by the shape and natural properties of the objects being tracked. Object tracking can be defined as the annotation of frames with trajectories of moving objects around the scene. The methods used for object tracking vary according to the targeted domain and the required output. Common methodologies for representing objects include point sets, primitive shapes and regional representations.

3.1 Point representations

Point trackers model the objects as a set of points and track the movement of the points from one frame to another. An object can either be represented by a single centroid point (Figure 1[a]) or by a set of points (Figure 1[b]). Point representations are usually applied for tracking multiple objects moving in different directions around a fixed scene. In [3] point trackers were used to track very small objects (distant birds). Goldman et al in [1] use point tracking to tag and annotate subjects with speech balloons, video graffiti, path arrows, video hyperlinks and schematic storyboards. The procedure for this representation can be summed up in a three step process. Firstly, points of interest dispersed throughout an initial frame are selected. Secondly, the motion of the individual points is tracked in subsequent frames. Thirdly the points are grouped together by similarity in motion vectors. The groups of points will identify distinct objects. We will further discuss the methodology for point selection and tracking in section 4.

3.2 Primitive shapes

An alternative object representation technique uses primitive shapes, such as a rectangular frame, that surrounds the boundary of an object (Figure 1[c]). Primitive shape representations are appropriate for objects whose shape is generally rigid and can be approximated by standard shapes such as rectangles or ellipses (Figure 1[d]). Such modelling allows the use of more complex transformations such as affine, translation or projection when defining motion tracks. Using multiple primitive shapes for object modelling allows the use of kinetic relations between the objects (Figure 1[e,f]) and can track the change in the shape of the object apart from only movement.

The identification of objects in this model is generally done using background subtraction. Background subtraction identifies the effected regions of movement in a frame by differencing the pixels in adjacent frames. Kernel tracking is used to capture the motion of objects represented as primitive shapes. This motion can be either parametric motion, such as translation, conformal and affine transformations or dense flow fields.

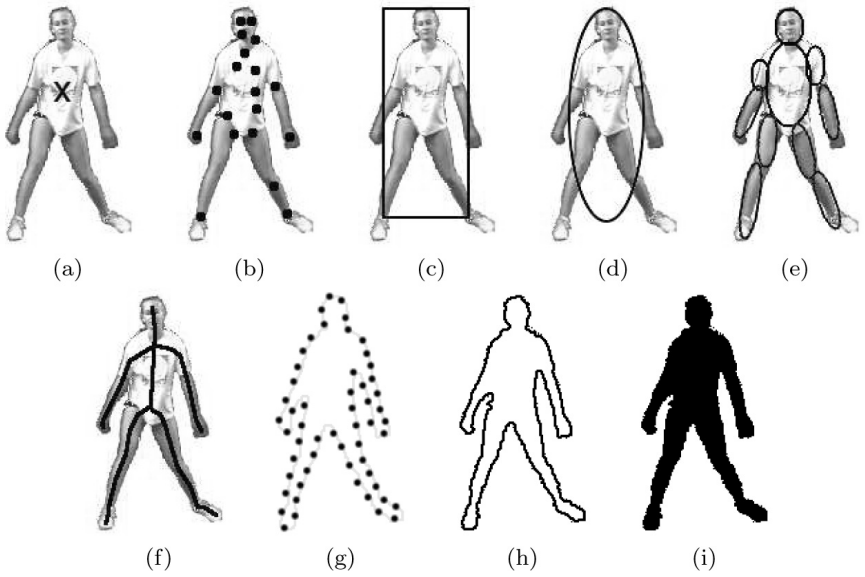

Figure 1: Different representations for the same (human) subject. (a) centroid point (b) multiple points (c) rectangular shape (d) elliptical shape (e) multiple shapes (f) skeleton (g) contour (h) points on contour (i) regional silhouette *Figure taken from Yilmaz el al [4]*

3.3 Regional representations

For representing objects having complex shapes, or shapes that can change considerably (such as humans) silhouette or contour representations are the most appropriate. In this model regions can be defined as either a set of points lying on the boundary (Figure 1[g]) of the objects, a contour surrounding the edge of the object (Figure 1[h]) or a silhouette defining the region occupied by the object (Figure 1[i]). Object identification in such representations is usually performed either by using an image segmentation algorithm or by using background subtraction (for moving objects). Such algorithms are able to track the exact object's shape and position in each frame.

Other object identification techniques such as segmentation or supervised learning can also be applied. Segmentation partitions the frames into similar regions while supervised learning is usually used to track a pre-specified class of objects (which have distinctive features) by training a classifier. The latter case requires a large collection of samples for each object class. Silhouette tracking is typically performed either by shape matching or by contour tracking, which enables accurate descriptions of the objects' shapes in individual frames in terms of pixel regions occupied by each object.

4 Methodology

The methodology for object extraction and tracking is pretty much dictated by the object representation as well the desired output. It is also very dependent on the application domain and the nature of the multimedia. Methodologies may vary on the number, type and nature of objects

being tracked. The nature of the multimedia includes the distinction of shots taken by a fixed camera versus a moving camera (which implies the existence of a static background or otherwise), change in illumination across the shots and the use of a single or multiple camera shots.

In this project we will aim at crafting a methodology for object segmentation within compressed videos which uses a combination of features from point tracking algorithms and static image segmentation algorithms. We will start by a baseline model for object segmentation based only on point tracking features. We will then explore possible enhancements to the methodology by adding features and techniques which are used in other object segmentation algorithms and domains.

4.1 Point Tracking

Point tracking algorithms have already proved their usefulness in various domains [1] and adopting a point tracking and particle grouping approach will grant more flexibility to the application. Intuitively, moving points are not effected by the size or shape of objects. They are also invariant to the number of objects in the scene, their angle to the camera and type of movement. As the system will be annotating compressed video streams, features such as motion vectors within the compressed stream can be exploited to increase the speed of the application.

The aim of point tracking algorithms is to record the movement of a set of selected points across adjacent frames. The approach can be abstracted in two main processes - point selection and point tracking. As explained in [4] the key element in point selection is to find a set points in the scene that are invariant to changes in illumination and camera viewpoint. Such points will enhance tracking as their presence will tally with the presence and movement of the underlying objects. Point selection is usually the result of common interest point detectors such as Moravec's interest operator, Harris interest point detector, KLT detector or SIFT detector.

Point tracking is achieved by recording the translation of each point with its corresponding point in adjacent frames. The points' motion can be expressed as a set of motion vectors for each frame which describe the transposition of each point from a frame to the subsequent frame. The main challenge in point tracking is point correspondence, especially when dealing with entrance and exits of objects in the scene. This issue can be overcome by defining movement constraints on the motion vectors. As we will discuss in the next section, constraints such as proximity, maximum velocity, small velocity change, common motion, rigidity and proximal uniformity can help to not only overcome the correspondence problem but also detect physical objects [4].

The motion vectors produced by point tracking algorithms can be used to identify physical objects in a given scene. Specific patterns in the motion vectors over a series of frames can be used to cluster points supposedly pertaining to distinct moving objects. Such patterns can be expressed as assumptions on properties such as constant velocity [5], constant acceleration, common motion [6] and rigidity [7] over a set of adjacent points. Approaches to extract such features can vary according to the type and nature of the multimedia. In the most general form, motion feature extraction is performed on raw, decoded video frames. However, in an attempt to enhance the application performance, we envisage to exploit features already present in compressed video technology, such as MPEG's motion vectors.

4.2 Static image segmentation

Whilst point tracking algorithms perform well in detecting rigid, roaming objects, their precision decreases when detecting nonrigid objects, objects which are part of the background or overlapping objects which are moving at the same speed. To compensate for such limitations, features that are commonly used in static image segmentation, such as color distribution, texture distribution, light intensity and location will be utilised. Although static image segmentation methodologies vary drastically, common approaches to feature selection can be observed across some techniques.

In clustering based image segmentation pixel features are used to define distance between pixels. Carson et al [8] use eight features in their model. Three features are used to describe the color in the L*a*b* color space. Another three features are used to describe the regional texture properties, namely anisotropy, polarity and contrast. The final two features are the x and y coordinates defining the location. Image segmentation techniques based on edge detection rely on regional pixel gradient estimates in the x and y directions. A threshold is applied to the strength of the gradient to decide the presence of edges. In graph-based segmentation a similarity measure is used to weigh the graph edges connecting nodes (pixels). In [9], Shi et al present an example of node clustering using only image brightness.

4.3 A combined model

Our proposed methodology will follow a standard clustering approach on a combination of features extracted from motion vectors and visual appearance properties. Identifiable objects will be tagged on a frame by frame basis, thus for the scope of a baseline algorithm we will assume that we will be performing point clustering on each individual frame in the video. As we have already discussed, the first step of the methodology is feature extraction. In the feature extraction pass, the individual frames in the video will be tagged by a set of distinguishable features comprised of visual appearance properties, texture properties and motion characteristics.

Visual appearance features can be extracted from a single, disjoint, uncompressed video frame, pretty much like a static image. The three normalised components of the L*a*b* color space can be used to describe the color properties. As the L*a*b* is a perceptually uniform color space, the euclidean distance between the components can be measured. Common techniques for describing regional texture properties include multiorientation filter banks and the second moment matrix. Carson et al in [8] adopt a simplified version of second moment matrix for describing texture in terms of anisotropy, polarity and contrast.

In this work, the second moment matrix of the gradient vectors describing the change of luminosity over a neighbourhood of pixels is used. Polarity is computed from the ratio of vectors residing on the positive side and the negative side of the principal component. This feature measures the extent to which gradient vectors in a region of pixels all point in the same direction along the principal eigenvector. Anisotropy is a property showing the degree at which gradient vectors are pointing along the dominant orientation and is computed from the difference between the first and the second eigenvalue.

Unlike visual appearance features, motion properties are time-based and must be viewed over a sequence of frames. Having said that, we know that our method will be clustering features over single frames. Thus, given a frame, we will extract motion features from the motion vectors of points appearing in the given frame and a sequence of neighbouring frames. Recall that motion vectors outputted by a point tracking algorithm are a set of transposition vectors for each frame at time t where each point will be described by a single motion vector. Thus, the motion of a point p appearing in a given frame at time t can be described a set of motion vectors of p from time $t - \lambda_1$ to $t + \lambda_2$.

Rather than working on the motion vectors directly, we will work on the second moment matrix of the vectors within this time window. Consider M_p to be the covariance matrix of motion vectors for point p, we can extract motion features from the eigenstructure of M_p. The principal eigenvector of M_p will give us the general orientation of the point's motion. The ratio between the two eigenvalues will show us the dominance of the orientation of the vectors towards the principal eigenvector. The *direction strength* can be defined as the ratio of positive-sided vectors against negative-sided vectors when motion vectors are projected onto the principal eigenvector. A high *direction strength* will show dominance in vectors pointing in the same direction. Points having a low ratio between the two eigenvalues or a low *direction strength* will be discarded as these will show intermittent motion behaviour of the point.

Features that can be computed from the eigenstructure of M_p include the general motion direction, the general degree of motion (distance) and the velocity behaviour. The general direction of the motion vectors can be expressed as the angle of rotation of principal eigenvector from the principal axis. The degree of motion can be computed as the mean magnitude of the projection of the motion vectors on the principal eigenvector. The general velocity behaviour is shown by the variance of data along the principal eigenvector, that is, the largest eigenvalue. A small variance will be indicative of a constant velocity.

4.4 Clustering

Following the feature extraction pass, the second step of our methodology constitute of point clustering. In this step, spatial points having similar features will be grouped together. A standard clustering approach is typically adopted in this step. Carson et al [8] use the EM algorithm to approximate K Gaussians in a mixture of Gaussians model and cluster the pixels into in K groups. In this work, clustering is followed by running the connected-components algorithm to group the pixels into regions. However due to the overwhelming amount of data that the video domain entails we might adapt to a simpler clustering algorithm such as K-Means.

More so, even the simplest clustering algorithm might be too expensive to compute for each frame. As algorithms such as K-Means must be seeded with the number of expected clusters (K), and the optimal K for each frame is not known beforehand, clustering must be iterated over different K values for each frame. We aim to reduce the complexity by exploiting the non-rigid output of the algorithm (further described in section 4.5). We aim to perform full clustering only at certain time intervals and approximate the objects' positions in frames between the time intervals.

As discussed in section 4.1, point trackers often limit tracking on a set of selected interest points, rather than on the whole pixel set. We also aim to explore the effect of reducing the frame size by interpolating small regions of pixels.

4.5 Inputs and Outputs

Our proposed algorithm will work on compressed MPEG video streams and produce metadata describing the appearance of physical objects at any point throughout the stream. An object will be identified by a rectangular region of pixels in which it resides and a unique identifier. Annotating objects' boundaries using a rectangular frame will both simplify the evaluation of the project and will also be forgiving on imprecise boundary detection. A singular physical object will also be tracked across adjacent frames. Tracking will be accomplished by assigning the same identifier to an object across adjacent frames. The object's motion track can then be computed as a series of transpositions of the object's centroid.

A frame can contain multiple objects identifiable by different object identifiers. However, a frame cannot contain two or more objects having the same identifier. A physical object will be identified if and only if it spans over a number of adjacent frames such that it is visible for long enough to be recognised by a human. Such constraint is essential for the (manual) evaluation of the system. Although real-time object identification is desirable, it is not required by our application and thus we assume that metadata annotation will be processed off-line. A sample annotated frame is shown in Figure 2 where the algorithm is identifying five physical objects using a yellow rectangular frame.

5 Evaluation

The performance of the methodology will be evaluated in terms of the average *recall* and *precision*. We will informally define *recall* and *precision* as follows:

Recall is the percentage of objects that the algorithm managed to identify. It will be defined as the ratio of objects that were correctly detected by the algorithm against the total number of objects present in the video.

Precision is the percentage of objects that the algorithm managed to identify correctly. This will be defined as the ratio of objects that were correctly marked by the algorithm against the total number of objects.

The algorithm's output will be evaluated on a set of preselected video frames. We will refer to such frames as *test frames*. Each *test frame* will be manually annotated such that all the prominent objects visible in the frame are tagged by a bounding rectangular frame. The manually tagged object regions will be used as the *ground truth*. The *ground truth* and the *hypothesised regions* of objects marked by the algorithm will then be matched to mark the objects that were correctly

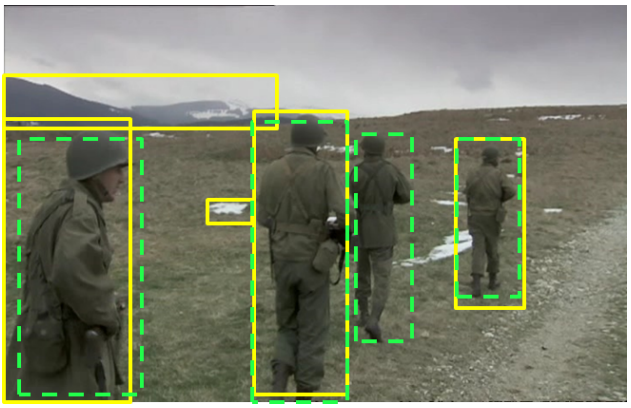

Figure 2: A sample frame from a scene showing objects annotated by the algorithm (yellow solid rectangle) and manually annotated objects (green dashed rectangles).

detected by the algorithm to be used in the recall and precision measures. Thus we can define *recall* and *precision* more formally as:

$$Recall = \frac{|\{\text{ground truth objects}\} \cap \{\text{hypothesised objects}\}|}{|\{\text{ground truth objects}\}|}$$

$$Precision = \frac{|\{\text{ground truth objects}\} \cap \{\text{hypothesised objects}\}|}{|\{\text{hypothesised objects}\}|}$$

We will identify a match between a ground truth object and a hypothesised object by the area of overlap between the two rectangular regions. The evaluation function will be realistically forgiving on slight misalignment between the two bounding frames. In particular, the ratio of overlapping area between the two regions and the total non-overlapping area covered by any single region must be below a prespecified threshold, say 70%.

Figure 2 shows a sample test frame containing both manually annotated objects (green, dashed rectangular frames) and hypothesised identified objects (yellow, solid rectangular frames). In this sample we can see that there are three correctly identified objects, one non identified object and two additionally identified objects. Thus we can compute the *recall* as 75% (3/4) and the *precision* as 60% (3/5) in this particular frame. One must note that in the sample frame shown in Figure 2, it is arguable whether the two additional identified objects can be considered as incorrectly identified. The fuzzy notion of what can be considered as an object might result in a mismatch between the hypothesised object set and the ground truth object set resulting in a lower precision. Due to this, we will compare the performance of our methodology with the results of a baseline point tracking algorithm.

In order to ensure that we have a representative set from the whole video archive that is capable of testing different aspects of the algorithm the sample set will be manually chosen. For instance, frames selected at random time intervals will be used to measure the general performance of object

Task	Start	End	Duration	2011		
				June	July	August
Further research	1/6/2011	21/6/2011	15			
Refine methodology	13/6/2011	29/6/2011	13			
Familiarisation with tools & libraries	27/6/2011	7/7/2011	9			
Develop basic prototype	7/7/2011	25/7/2011	13			
Tag testing videos	26/7/2011	1/8/2011	5			
Enrich methodology	1/8/2011	12/8/2011	10			
Write report	1/8/2011	18/8/2011	14			

Figure 3: Project Plan

detection while a frame set annotated at small regular intervals will be used to evaluate the object tracking capability. We expect different performance from videos varying in characteristics, so the performance will also be compared across different video sets. Such video sets can exhibit different characteristics such as a varied number of objects, single versus multiple scene shots, single versus multiple camera shots and static versus moving camera shots.

6 Work plan

The work on the project will span over an approximate eleven week period. The work will start off by some deeper research related to point tracking algorithms and feature extraction which will help in refining the proposed methodology. This will be complemented by familiarisation with specific software libraries that will be used thought the project in view of the media content. Following this, a basic prototype of the baseline methodology comprised of only a small feature set will be developed. This will serve as a proof of concept. To check the performance of the algorithm one needs the testing sample space as described in section 5. Thus some time will be dedicated to the manual annotation of the testing set.

The methodology will then be enriched by adding more features and the performance at each step will be analysed. The analyses of the algorithm performance in relation with different feature sets and different parameters will form the basis of this project. Figure 3 depicts the main tasks of the project across the eleven week period.

References

[1] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 3–12, New York, NY, USA, 2008. ACM.

[2] Y. Ohno, J. Miura, and Y. Shirai. Tracking players and estimation of the 3d position of a ball in soccer games. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 145 –148 vol.1, 2000.

[3] K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 110 –115 vol.1, 2003.

[4] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38, December 2006.

[5] S.S. Intille, J.W. Davis, and A.F. Bobick. Real-time closed-world tracking. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 697 –703, June 1997.

[6] C.J. Veenman, M.J.T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):54 –72, January 2001.

[7] Ishwar K. Sethi and Ramesh Jain. Finding trajectories of feature points in a monocular image sequence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(1):56 –73, 1987.

[8] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:1026–1038, August 2002.

[9] Jianbo Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Computer Vision, 1998. Sixth International Conference on*, pages 1154 –1160, January 1998.