Clustering Videos by Location

Florian Schroff[1]

C. Lawrence Zitnick[2]

Simon Baker[2]

[1] Visual Geometry Group
University of Oxford

[2] Microsoft Research
Microsoft Corp.

Abstract

We propose an algorithm to cluster video shots by the location in which they were captured. Each shot is represented as a set of keyframes and each keyframe is represented by a histogram of textons. Clustering is performed using an energy-based formulation. We propose an energy function for the clusters that matches the expected distribution of viewpoints in any one location and use the chi-squared distance to measure the similarity of two shots. We also add a temporal prior to model the fact that temporally neighboring shots are more likely to have been captured in the same location. We test our algorithm on both home videos and professionally edited footage (sitcoms). Quantitative results are presented to justify each choice made in the design of our algorithm, as well as comparisons with k-means, connected components, and spectral clustering.

1 Introduction

Location is a useful source of information for a variety of tasks. Just as users may want to tag and search their personal photo collections and videos for specific people, they may also want to specify a location to further narrow down the search. Users may also want to browse videos by location, annotate locations, or create location specific compilations.

In this paper, we propose an algorithm that uses visual information to cluster video shots by the location in which they were captured. We demonstrate our algorithm on both home videos and professionally edited footage such as sitcoms [1, 20]. In the context of home movies, location generally means a specific room in the house, or a frequently visited place outside, such as in the garden, or at the local park. In the context of sitcoms, location means a film "set" such as the coffee shop in the sitcom "Friends." We chose to develop an unsupervised clustering algorithm. Such algorithms can be combined with manual intervention to allow efficient tagging, as has been demonstrated by the use of face recognition in commercial photo organization software [2, 5]. In these systems, the primary use of face recognition is to cluster faces into groups that can all be tagged at the same time.

Our algorithm breaks the video into shots first [6]. This is based on a simple color histogram algorithm. It is important to fully represent the visual varieties in each shot. We empirically compare three approaches: (1) Using a single keyframe, the middleframe of the shot. (2) Using multiple keyframes sampled uniformly in time from the video [17]. And

BMVC 2009 doi:10.5244/C.23.48

(3) Stitching the frames into a mosaic [1]. Using multiple keyframes is a robust way to ensure that the full variety of the shot is captured. This is confirmed by our results, where this second approach performs slightly better than the other two.

Given the keyframes, we need to measure the similarity between each pair of shots. We considered two approaches: (1) Each keyframe is represented by a histogram of textons [4] and the chi-squared distance between the histograms is used to measure similarity. (2) Each keyframe is represented by a bag-of-words representation computed using vocabulary trees [14] and based on MSER features [11]. The similarity score between keyframes is then computed using Term Frequency Inverse Document Frequency (TF-IDF) scoring [18]. Empirically we found the first approach to perform substantially better.

The last step in the design of our algorithm is the core clustering algorithm. Again, we considered several choices: (1) k-means, (2) a "connected components" algorithm, (3) a spectral clustering algorithm [12], and (4) a model-based algorithm [9] using an energy function that is specifically designed to model the expected shape of clusters for the task at hand. In particular, intuition about the likely distribution of viewpoints and their overlap in a typical room provided motivation for the specific energy function used, which is shown to outperform the other methods in our experiments.

This completes the visual part of our algorithm. As subsequent shots in a video are likely to have been captured at the same location it is reasonable to incorporate this prior knowledge into the clustering process. We show how a temporal prior[1] can be added to the energy function in our model-based clustering algorithm, and was found to improve performance significantly.

Throughout the paper we provide quantitative empirical evaluations on both home videos and professionally edited content (4 episodes of the sitcom "Friends") to justify each choice made in the design of our algorithm. These evaluations are performed using manually-specified ground-truth location labels. We treat the problem as a binary classification task. Either two shots are at the same location or not. We are then able to plot ROC curves; *i.e.* the false positive classification rate against the true positive rate.

2 Algorithm

2.1 Computing and Representing Shots

The first step in our algorithm is to break the video into shots [6]. We use a simple algorithm based on color histograms and choose a parameter setting that tends to result in an oversegmentation of the video, as we want to ensure that each shot contains just one location.

Each shot typically contains between a few 10s and a few 100s of frames. There are three main approaches to representing the multiple frames in a shot. The simplest approach is to use a single keyframe, typically the middle frame. Another approach is to use multiple keyframes [17]. Finally the multiple frames (or a subsampling of them) can be stitched into a mosaic [1]. We illustrate these three choices in Figure 1(a).

To choose the best method, we empirically compared these three choices (keeping all other components of the algorithm identical.) The multiple keyframes are sampled uniformly

[1] In general videos, particularly home videos, there is rarely enough temporal consistency to perform a temporal segmentation into scenes, as has been the focus of a lot of work dedicated solely to processing professionally edited content such as TV shows and full length movies [1, 6, 10, 20]. We argue that for clustering arbitrary videos by location, a temporal prior is more appropriate than a temporal segmentation.

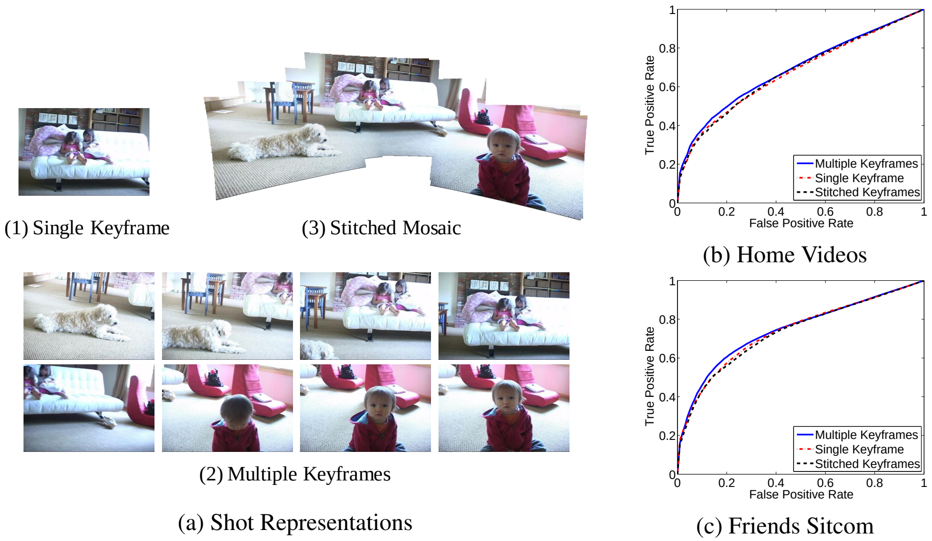


Figure 1: **Shot Representation**: (a) Three approaches to representing a video shot: (1) use a single keyframe, (2) use multiple keyframes, and (3) stitch the keyframes into a mosaic. (b) and (c) Empirical results show the multiple keyframe approach to perform slightly better than the other approaches. See Section 3 for the details of the data and how the evaluations are performed.

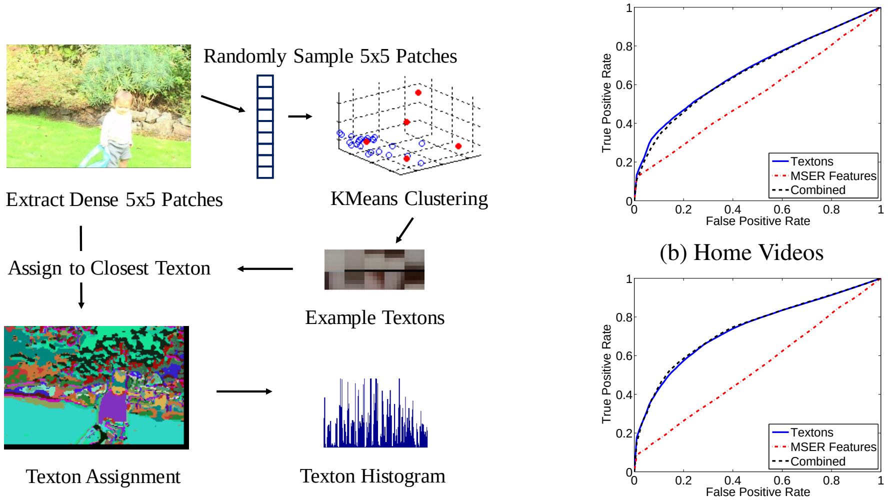
every 20 frames from the shot. The mosaic is constructed using the same frames. The details of how we performed the evaluation are contained in Section 3. We include ROC curves for two types of videos: (1) home videos and (2) professionally edited content, specifically episodes of the TV sitcom “Friends”. The details of the data are included in Table 1 in Section 3. The results in Figure 1(b) and (c) are each averages over 4 videos. The multiple keyframe approach performs slightly better than the other two approaches.

2.2 Inter-Keyframe Similarity Measure

Much of the prior work on clustering or recognizing location in video [7, 13, 17, 19] has focused on the matching cost between a pair of shots or keyframes, or the closely related question of the representation. In particular, Torralba *et al.* [19] used GIST features, Schaffalitzky and Zisserman considered feature point matching [17], Ni *et al.* [13] used an epitome based representation, and Heritier *et al.* [7] used Latent Dirichlet Allocation [3].

Here we explore two approaches. The first representation consists of a histogram of textons [4]. First, a texton vocabulary consisting of 128 textons is learnt offline using randomly sampled 5×5 patches and k-means clustering. For each keyframe in a shot we extract 5×5 patches in a dense grid. Each patch is then assigned to the closest texton. Finally, by aggregating over the entire keyframe we compute a histogram of textons for that keyframe. See Figure 2(a) for an illustration. We compared (results omitted) a number of different distance metrics (L2, L1, chi-squared) to compute the distance between a pair of texton histograms and found the chi-squared distance to perform the best. We also considered the use of Latent Dirichlet Allocation (LDA) [3] to learn a set of topic histograms, as in Heritier *et al.* [7]. Empirically we found LDA gave no significant improvement in performance.

Our second inter-keyframe distance function is based on point feature matching using the bag-of-words approach with vocabulary trees proposed by Nister and Stewenius [14]. First,



(a) Computation of Texton Histograms

(b) Home Videos

(c) Friends Sitcom

Figure 2: **Inter-Keyframe Similarity Measure**: (a) A set of textons are learnt offline using k-means clustering. A dense grid of 5×5 patches is extracted from each keyframe, matched to the closest texton, and a histogram computed. We use the chi-squared distance to measure the similarity of texton histograms. (b) and (c) Empirical results show that texton histograms significantly outperform the use of MSER features, and there is little benefit to combining the two approaches.

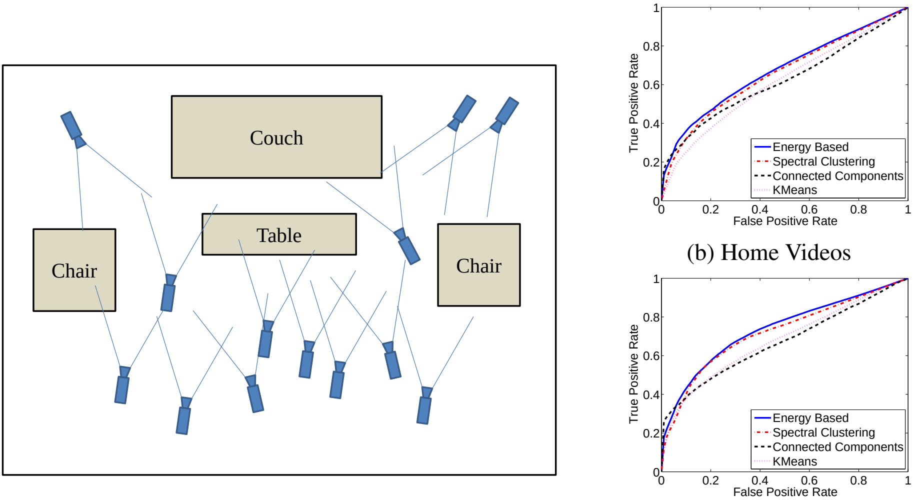
a set of affine invariant features are found using MSER [11]. A visual word is assigned to each image patch extracted by the features using a vocabulary tree. For our experiments, the vocabulary tree had one million leaf nodes. Finally, the similarity score between images is computed using Term Frequency Inverse Document Frequency (TF-IDF) scoring [18].

In Figures 2(b) and (c) we include results comparing texton histograms with the MSER feature matches. The results show that the approach of using texton histograms significantly outperforms the MSER feature based matching. We suspect that this difference in performance is because the MSER feature matching only provides information when there is a significant overlap in viewpoint. On the other had, the texton histograms provide information even when there is little or no overlap in viewpoint. We experimented with ways of combining the two approaches, but with no improvement in the results. It appears that whenever there is significant overlap in viewpoint and the MSER features provide a strong match, the texton histograms also provide a strong match. In Figures 2(b) and (c) we include the results of a simple weighed combination of the two measures to illustrate this point.

2.3 Cluster Model

We expect the set of shots captured in a single location to have a characteristic structure. Figure 3(a) illustrates a possible set of viewpoints in a room. For simplicity, we do not illustrate pans or zooms, although they may be present as well. Most viewpoints have substantial overlap with a few others. Such overlapping pairs of viewpoints can be expected to match well for many possible similarity measures, including the texton-based measure in Section 2.2. On the other hand, there are many pairs of viewpoints that have little, if any, overlap. The similarity of these pairs will vary significantly depending on the complexity of the scene.

For simplicity, assume that there is a single keyframe per shot. We explain how to extend the algorithm to deal with multiple keyframes in Section 2.5. We first compute the texton



(a) Example Distribution of Viewpoints in a Room

(b) Home Videos

(c) Friends Sitcom

Figure 3: **Cluster models**: (a) An illustration of possible viewpoints in a room. For simplicity we do not show pans and zooms. This figure illustrates the intuition behind our choice of energy function. In (a) each viewpoint is close to a number of others. This suggests locally well connected clusters, which will be represented by our choice of energy function. Empirical results, (b) and (c), show that our energy-based algorithm with a cluster model based on the intuition in (a) outperforms k-means clustering, a simple connected components algorithm, and a spectral clustering algorithm [12].

similarity measure in Section 2.2 between all pairs of keyframes. This sets up a fully connected graph where the nodes are the keyframes and the length of each edge is the similarity measure between the corresponding pair of keyframes.

We would like to develop a clustering algorithm that allows us to model the likely cluster shapes and connectivity based on the intuition in Figure 3(a). An approach that allows us to do this is the “model-based” approach of [9]. In this approach, almost any energy function can be used. This freedom to choose an energy function allows us to encode a preference for a certain shape of clusters. Our choice of energy function is based on the observation that each viewpoint is close to a number of others in Figure 3(a) by encouraging locally well connected clusters. Specifically, although clusters can be “elongated” in shape they need to be well connected, as would be the case in Figure 3(a) due to the overlapping viewpoints. This differs from standard k-means where usually ball shaped clusters are assumed, or standard agglomerative clustering where also either very compact or very loosely connected clusters are assumed.

Assume that the graph has already been split into a set of disjoint clusters $\{C_1, C_2, \ldots\}$. The cluster energy is then defined as a sum of energies, one for each cluster:

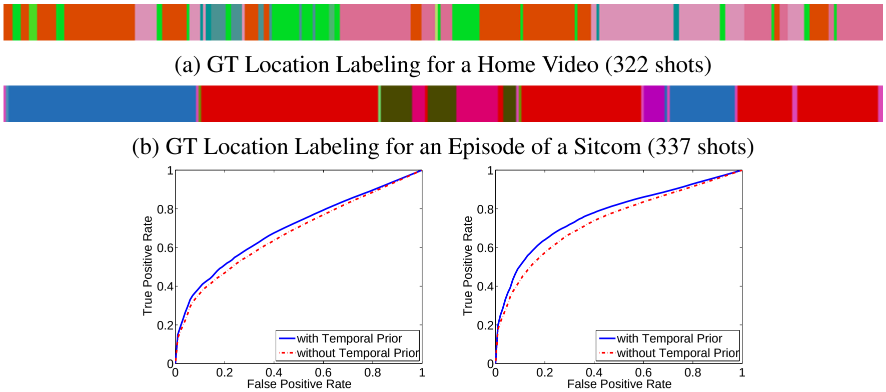
$$E_{\text{Cluster}} = \sum_i \text{MST}(C_i^N). \tag{1}$$

In this equation $\text{MST}(C_i^N)$ is the length of the minimum spanning tree (MST) of C_i^N and:

$$C_i^N = C_i^{N-1} - MST(C_i^{N-1}), \quad C_i^1 = C_i \tag{2}$$

is a recursive definition which says that C_i^N should be computed by removing[2] all of the edges

[2] To avoid the possibility of any cluster becoming disconnected, instead of removing the edges, we actually replace the edges with the longest edge in the cluster C_i.



(a) GT Location Labeling for a Home Video (322 shots)

(b) GT Location Labeling for an Episode of a Sitcom (337 shots)

(c) Home Videos

(d) Friends Sitcom

Figure 4: **Temporal Prior**: (a) and (b) Color-coded visualizations of ground-truthed location labellings of a home video and a sitcom. Shots with the same location have the same color. Both videos exhibit a large degree of temporal coherence, especially the sitcom. (c) and (d) Empirical results which show that the addition of a temporal prior improves performance.

in the minimum spanning tree (MST) from C_i^{N-1}; *i.e.* C_i^N is the graph obtained after removing $N-1$ MSTs in sequence from C_i. In summary, the energy cost of a cluster $\text{MST}(C_i^N)$ is the length of its MST, after having previously removed $N-1$ MSTs. In this definition, $N = \alpha(|C_i|-1)$ is a constant proportion of the size of the cluster $|C_i|-1$. We use $|C_i|-1$ rather than $|C_i|$ because $|C_i|-1$ is the number of other nodes that each node is connected to.

In Figures 3(b) and (c) we present results comparing our algorithm with k-means clustering, simple connected components (equivalent to single-link agglomerative clustering), and a spectral clustering algorithm [12]. We find that on both types of data, our cluster model performs the best. Following [9] our energy function is a generalization of the single-link energy and thus ensures locally connected clusters as outlined in the intuition above. The improved performance confirms the intuition of the cluster shapes as induced by Figure 3(a).

2.4 Temporal Prior

Two subsequent shots in a video are more likely to be captured in the same location than not. This is particularly true for professionally edited footage such as sitcoms and movies. It is also true for home videos, although to a lessor extent. In Figure 4(a) and (b) we include visualizations of the temporal smoothness for both a home video and one episode of a sitcom. We color-code the location (ground-truthed by a human) and display the sequence of frames as a horizontal bar. In previous work on clustering sitcoms [1, 20] a temporal segmentation was used as a first step to break the video into scenes. While possibly appropriate for sitcoms, the reduced temporal consistency in the home video makes such an algorithm inappropriate. Instead, we add a temporal prior[3] to yield a global energy function:

$$E_{\text{Global}} = E_{\text{Cluster}} + \lambda E_{\text{Temporal}}. \tag{3}$$

The temporal prior is:

$$E_{\text{Temporal}} = \sum_t \delta(\mathbf{s}_t, \mathbf{s}_{t+1}) \tag{4}$$

[3] Another approach would have been to weight the texton-based match scores with the temporal separation of the shots. Such an approach is less principled than the addition of a temporal prior.

where $\delta(\mathbf{s}_t, \mathbf{s}_{t+1})$ is an indicator function:

$$\delta(\mathbf{s}_t, \mathbf{s}_{t+1}) = \begin{cases} 1 & \mathbf{s}_t \in C_i, \mathbf{s}_{t+1} \in C_j, i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Equations (4) and (5) count the number of times that temporally neighboring shots belong to different clusters. In Markov Random Field terminology, these equations describe a Potts model [15] where a penalty is added between neighboring frames if they occur in a different location. The constant weighting factor λ was chosen empirically to be $\lambda = 100$ (in all experiments). Figures 4(c) and (d) include empirical results that show the temporal prior improves performance on both the home video and sitcom data.

2.5 Optimization

As in [9], we optimize the global energy E_{Global} in Equation (3) using a greedy algorithm that is equivalent to agglomerative clustering. We initialize the algorithm by assigning each shot to its own cluster. With a single keyframe or a stitched mosaic, each cluster starts with a single node. With multiple keyframes, each cluster starts with multiple nodes. Pairs of clusters are iteratively merged. In each iteration, we consider every possible pair of merges and compute the change to E_{Global}. The merge that results in the lowest new value of E_{Global} is then applied.

A natural question is when to terminate the algorithm. One possible approach would be to threshold the change in E_{Global}. Such simple approaches rarely work well. Determining the number of clusters is a very difficult problem, arguably far harder than estimating the best K clusters for a given K. In many cases, however, estimating the number of clusters accurately is not essential. In photo organization systems that use face clustering [2, 5], the user interface dictates that only small clusters, an "over-clustering", are presented to the user. For the quantitative evaluation in Section 3, our approach is to cluster for every possible K and plot parametric ROC curves across K.

Two possible criticisms of the above algorithm are that it is greedy, and that there is no guarantee of convergence to the global minimum. Note, however, that due to the combinatorial nature of the problem, most clustering algorithms contain a greedy component, including k-means, and spectral clustering algorithms such as [12]. Because our algorithm allows the use of more complex cluster energies such as those in Equations (1) and (3), we found it outperforms these other approaches.

3 Experiments

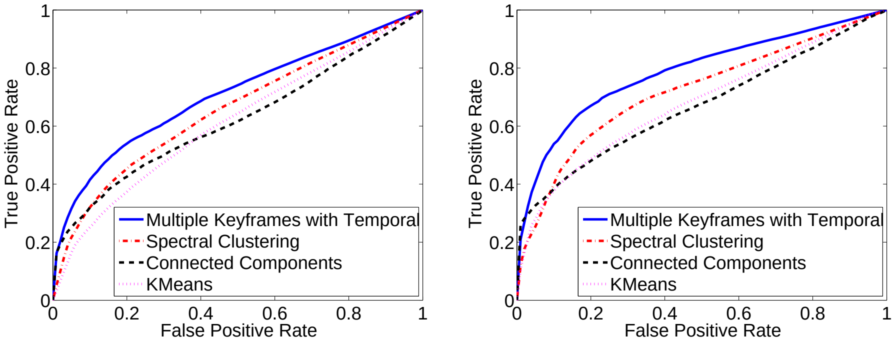
We experimented on 8 videos, 4 home videos (captured by 3 different people in 3 different houses), and 4 episodes of the sitcom "Friends". The home videos were generally captured over multiple days, although the segments were consecutive on the tapes. Table 1 includes some statistics, including the length in minutes, the number of shots, and the number of distinct locations in the manually labeled ground-truth.

Our quantitative evaluations (including those in Figures 1, 2, 3, and 4) are based on treating the problem as binary classification, where the algorithm must decide whether two shots are in the same location or not. Given a clustering $\{C_1, C_2, \ldots\}$, and the ground-truth

	Length	Shots	Locs
Home 1	27:00 Min	322	10
Home 2	24:00 Min	396	12
Home 3	23:00 Min	313	7
Home 4	20:00 Min	146	4

	Length	Shots	Locs
Friends 1	20:26 Min	337	10
Friends 2	20:00 Min	376	6
Friends 3	19:50 Min	390	11
Friends 4	18:57 Min	296	6

Table 1: **Data Statistics:** We experimented on 4 home videos and 4 epsiodes of the sitcom "Friends".



(a) Home Videos

(b) Friends Sitcom

Figure 5: **Quantitative Results**: A comparison of our energy-based algorithm (with multiple keyframes and a temporal prior) with k-means, a spectral clustering algorithm [12], and a connected components algorithm (all without multiple keyframes and a temporal prior.) The combination of our cluster-model, multiple keyframes, and a temporal prior yields a significant performance improvement.

clustering $\{C_1^{\mathrm{GT}}, C_2^{\mathrm{GT}}, \ldots\}$ we compute the false positive and true positive rates:

$$\begin{aligned}\mathtt{FPR} &= \left|\left\{(\mathbf{s}_i,\mathbf{s}_j) \in \mathrm{F}^{\mathrm{GT}} \,|\, i<j,\, \mathbf{s}_i \in C_k,\, \mathbf{s}_j \in C_k\right\}\right| / \left|\mathrm{F}^{\mathrm{GT}}\right| \\ \mathtt{TPR} &= \left|\left\{(\mathbf{s}_i,\mathbf{s}_j) \in \mathrm{P}^{\mathrm{GT}} \,|\, i<j,\, \mathbf{s}_i \in C_k,\, \mathbf{s}_j \in C_k\right\}\right| / \left|\mathrm{P}^{\mathrm{GT}}\right|\end{aligned}$$

where $\mathrm{P}^{\mathrm{GT}} = \left\{(\mathbf{s}_i,\mathbf{s}_j) \,|\, i<j,\, \mathbf{s}_i \in C_k^{\mathrm{GT}},\, \mathbf{s}_j \in C_k^{\mathrm{GT}}\right\}$ is the set of unordered positive match shot-pairs in the ground truth and $\mathrm{F}^{\mathrm{GT}} = \left\{(\mathbf{s}_i,\mathbf{s}_j) \,|\, i<j,\, \mathbf{s}_i \in C_k^{\mathrm{GT}},\, \mathbf{s}_j \in C_l^{\mathrm{GT}},\, l \neq k\right\}$ is the set of unordered negative match shot-pairs. False positives are unordered shot-pairs that are *not* in the same cluster in the ground-truth but in the same cluster in the evaluated clustering. True positives are unordered shot pairs that are in the same cluster in both clusterings. Rather than combining these two measures as in the Rand Index [8, 16], we keep them separate and plot a ROC curve (across the iteration of the algorithm, *i.e.* the number of clusters.) We average the ROC curves over a fixed size window of 100 shots that is slid through the video, and subsequently average over the 4 videos of each type.

In Figure 5 we compare our algorithm with k-means, a spectral clustering algorithm [12], and a connected components algorithm (equivalent to single-link agglomerative clustering). The main benefits of our algorithm are: (1) it allows the use of more complex cluster energies such as Equation (1), (2) it is straight-forward to add a temporal prior as in Equation (3), and (3) it is easy to use multiple keyframes. In Figure 5 we present results obtained by our algorithm with these enhancements. The results for the other algorithms do not, as it would be hard, if possible at all, to add them. All algorithms use the same texton-based similarity measure. To give a sense of the numerical results for the "Friends" results in Figure 5 at a False Positive Rate of 20%, the True Positive Rate is 66.8% for our algorithms and 56.2% for the spectral clustering algorithm.

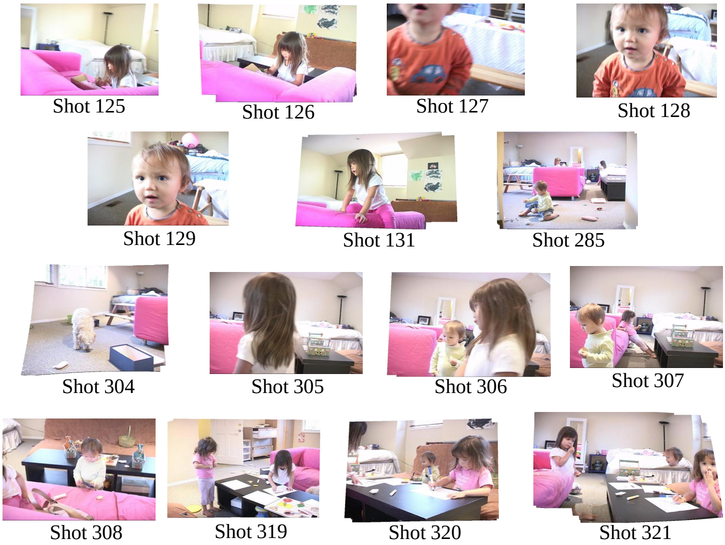

Shot 125
Shot 126
Shot 127
Shot 128
Shot 129
Shot 131
Shot 285
Shot 304
Shot 305
Shot 306
Shot 307
Shot 308
Shot 319
Shot 320
Shot 321

Figure 6: **An example cluster** illustrating the main points of our algorithm. See text for more details.

In Figures 6 and 7 we include qualitative results for one of the home videos, “Home 1”. The algorithm uses multiple keyframes, texton histograms, our model-based energy function, and the temporal prior. We include the stitched mosaics to help better visualize the shots, although the results in the figures use the multiple keyframes approach of Section 2.1. The first thing to note is the wide variety of viewpoints in each cluster. *E.g.* the viewpoints of Shots 319–321 are very different from those of 125–129, 285, and 304–306. The pink couch, the most visually dominant object in the room, is present in most shots, but not in shots 127–129, and 320. Similarly, the viewpoints of 88 and 95 are very different from those of 89–92, and 96. Secondly, note how the viewpoints qualitatively match the example distribution in Figure 3(a). Also note that even pairs of overlapping viewpoints (e.g. 320 and 321) are widely separated enough that feature-matching would be very difficult, if possible at all. Such shot-pairs explain the poor performance of feature matching in Figure 2.

Another point to note is the importance of the temporal prior. *E.g.* shots 127-129 are visually very different from the rest of the cluser, but are sandwiched in time between shots 125, 126, and 131. Similarly, shot 88 which is a view of the other half of the room, is visually very different from the rest of the cluser in Figure 7, but is between shot 88 and shots 90–92.

Finally, shot 96 shows how the use of multiple keyframes can help in certain situations. The set of keyframes used in shot 96 includes one containing the boy in the orange shirt. It is likely that this keyframe is reasonably well matched to shots 90-92. Another keyframe in shot 96 includes the white couch and the bookshelf in the background. This keyframe is probably a good match for the upper right keyframe in shot 88.

4 Conclusion

We have presented an algorithm to cluster video shots by the location in which they were captured. We have systematically investigated each component in the algorithm: (1) how to repesent a shot (Section 2.1), (2) the similarity measure between a pair of keyframes (Section 2.2), (3) the cluster model (Section 2.3), and (4) the addition of a temporal prior (Sec-

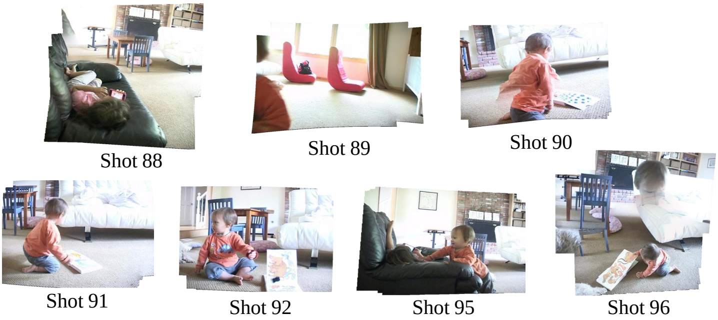


Figure 7: **An example cluster** illustrating the main points of our algorithm. See text for more details.

tion 2.4). At each step, we empirically compared the alternatives and chose the best one. The combination of all the enhancements is a significant improvement over baseline algorithms such as k-means and spectral clustering [12] which do not use those enhancements.

One of the apparent difficulties for clustering by location is the presence of transient foreground objects, primarily people. The same people wearing the same clothing may well appear in multiple locations, adding distractors to any inter-keyframe similarity measure. We carried out some preliminary experiments trying to model the foreground and background, and explicitly segment out the foreground. However, experiments showed that even masking the foreground by hand (surprisingly) actually deteriated the performance. One possible explanation is that masking out the people results in different areas of the background being masked in different shots. On the other hand, giving extra weight in the texton histograms to the borders of the image and reducing the impact of the center resulted in a small performance improvement. Future work can almost certainly result in further improvements.

References

[1] A. Aner and J. Kender. Video Summaries through Mosaic-Based Shot and Scene Clustering. In *European Conference on Computer Vision*, 2002.

[2] Apple, Inc. http://www.apple.com/ilife/iphoto/.

[3] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[5] Google, Inc. http://picasaweb.google.com/.

[6] B. Guensel, A. Ferman, and A. Tekalp. Temporal Video Segmentation Using Unsupervised Clustering and Semantic Object Tracking. In *SPIE*, 1998.

[7] M. Heritier, S. Foucher, and L. Gagnon. Key-places detection and clustering in movies using latent aspects. In *International Conference on Image Processing*, pages II: 225–228, 2007.

[8] L. Hubert and P. Arabie. Comparing Partitions. In *Journal of Classification*, 1985.

[9] S. Kamvar, D. Klein, and C. Manning. Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based Approach. In *International Conference on Machine Learning*, 2002.

[10] J.R. Kender and B.L.Yeo. Video scene segmentation via continuous video coherence. In *Computer Vision and Pattern Recognition*, 1998.

[11] J. Matas, O. Chum, M. Urba, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 384–396, 2002.

[12] A.Y. Ng, M. Jordan, and Y. Weiss. On spectal clustering: Analysis and an algorithm. *Neural Information Processing Systems*, 14, 2002.

[13] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic Location Recognition. In *Computer Vision and Pattern Recognition*, 2008.

[14] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition*, 2006.

[15] R. Potts. Some generalized order-disorder transformation. *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

[16] W. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, pages 846–850, 1971.

[17] F. Schaffalitzky and A. Zisserman. Automated Location Matching in Movies. In *Computer Vision, Image Understanding*, 2003.

[18] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[19] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In *International Conference on Computer Vision*, 2003.

[20] M.M. Yeung and B.L. Yeo. Time-constrained clustering for segmentation of video into story units. In *International Conference on Pattern Recognition*, 1996.