Using Co-occurrence and Segmentation to Learn Feature-based Object Models from Video

Thomas Stepleton
Robotics Institute
Carnegie Mellon University
tss@ri.cmu.edu

Tai Sing Lee
Center for the Neural Basis of Cognition
Carnegie Mellon University
tai@cnbc.cmu.edu

Abstract

A number of recent systems for unsupervised feature-based learning of object models take advantage of co-occurrence: broadly, they search for clusters of discriminative features that tend to coincide across multiple still images or video frames. An intuition behind these efforts is that regularly co-occurring image features are likely to refer to physical traits of the same object, while features that do not often co-occur are more likely to belong to different objects. In this paper we discuss a refinement to these techniques in which multiple segmentations establish meaningful contexts for co-occurrence, or limit the spatial regions in which two features are deemed to co-occur. This approach can reduce the variety of image data necessary for model learning and simplify the incorporation of less discriminative features into the model.

1. Introduction and related work

Co-occurrence is a powerful tool for discovering relationships between heterogeneous collections of attributes or events. Typically, if two such *features* frequently co-occur throughout a dataset, it is proposed that they correspond to traits of the same object, concept, or process. Co-occurrence is already a popular inference tool in language technologies such as machine translation [1] and searching and indexing [2], where co-occurring words are assumed to be semantically similar.

In computer vision, co-occurrence forms the basis for several techniques which automatically, or with minimal supervision, extract and model objects in video or image sets. Features that share frequent spatio-temporal co-occurrence are deemed to arise from the same object, and collections of these features form the basis for a model of that object. We propose that many of these techniques incorporate some or all of these four key processing steps:

1. **Feature tokenization**—a discretization of the feature space allowing consistent feature recognition throughout the dataset.
2. **Co-occurrence context establishment**—where spatial and temporal limits describe contexts in which co-occurrence is meaningful, reducing noise.
3. **Co-occurrence measurement**—where the frequency of feature co-occurrence is determined across the data.
4. **Feature clustering**—where strongly coincident feature sets are clustered into discrete object models.

Previous efforts closely resembling this schema include Sivic and Zisserman's unsupervised aggregation of affine-invariant feature sets from video sequences [3] and the unsupervised association of caption text keywords with image features [4]. In Sivic and Zisserman's case, steps 2 and 3 are implicit, manifesting themselves in the creation and filtering of spatially limited feature configurations.

Divergent but still relatable are Rothganger et al.'s automated discovery of 3D object models from video sequences [5], which uses affine-invariant feature matching and structure from motion; and Schmid's learning of 2D visual models from weakly-labeled training data [6].

In this paper, we discuss a refinement of the second step, in which a collection of segmentations is used to determine the pairwise co-occurrence likelihood of features. Aggressive context delineation is beneficial in cases where most of the image data look the same, such as a single, short video clip: here, a learner that groups objects appearing in roughly the same spatio-temporal location is likely to place all features into a single cluster. Segmentation can also limit the degree to which relatively non-discriminative features, appearing across wide swaths of an image, are judged to co-occur with nearby, unrelated features on a different object. We demonstrate a system that uses several segmentations to learn object models consisting of two different kinds of features of varying discriminative strengths. Our input data consist of 151 frames of a 30 frame-per-second 400x267 pixel video sequence, stills of which are seen in Figure 1.

Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (WACV/MOTION'05)
0-7695-2271-8/05 $ 20.00 IEEE

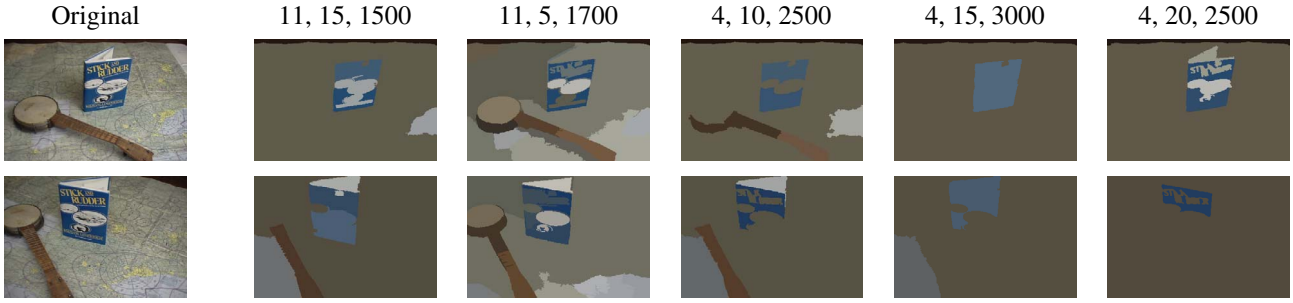

Figure 1. Two video frames and their five mean-shift segmentations. The three parameters for the mean-shift segmentation system—spatial bandwidth h_s, color range bandwidth h_r, and minimum region size—are displayed in corresponding order atop each segmentation column. Each segmentation provides information about the object boundaries in the image, but no single one finds all the appropriate boundaries, and several create spurious boundaries between different-colored regions of the same object.

Applications of unsupervised object model learning include search and indexing of video and image data, automated annotation and captioning, and the automated discovery of real-world objects by robots. Furthermore, these learning methods are often themselves showcases for applications of a variety of lower-level vision techniques, such as feature detection, segmentation, and structure from motion.

2. Model learning

We describe our learning mechanism in terms of the four-step schema presented in the introduction. The output of our system is similar to that of [3]: a collection of feature sets corresponding to objects without information on the relative spatial locations of the features.

2.1. Feature tokenization

We begin by extracting a collection of features from each frame in the video sequence. Here, we used two kinds of features: SIFT keypoint features [7], which are extremely discriminative, and small, localized color patches isolated from an oversegmentation of the image, comparable to the "superpixels" described in [8]. These superpixels are not very discriminative at all. Aside from demonstrating the robustness of the algorithm, the reasons for using such weak features were threefold. First, these color features can provide useful information in the absence of texture, which is required by SIFT. Second, by describing simple image attributes, they are often more intuitable to humans. Third, a heterogeneous feature set represents a more total use of the information present in the image.

Each SIFT descriptor in the collection is a point in a continuous 128-dimensional space, and each color patch is represented as a point in a 3-dimensional LUV colorspace. To greatly simplify matching features between frames, both spaces were vector quantized with K-Means clustering, as in [3]. SIFT descriptor space was reduced to 1,600 clusters, with each keypoint now assigned to the nearest cluster; LUV colorspace was likewise reduced to 100 clusters. As Sivic and Zisserman note, this process effectively renders the continuous video data into a stream of discrete tokens with spatio-temporal locations.

We perform a further consolidation of our SIFT features, which are simply scale and not affine-invariant. When a new SIFT feature is detected in a video frame, the underlying pixel data are tracked in future frames using an affine Lucas-Kanade tracker [9]. SIFT features that appear along the tracker trajectory are deemed to be the same as the original SIFT feature, and their corresponding K-Means clusters are also bound together as identical.

Finally, these unified SIFT-derived feature clusters are filtered for persistence and uniqueness. Those appearing in fewer than half of the images, or appearing more than once in an image in over five percent of their showings, are culled. For our sequence, 83 SIFT-derived clusters remained after this final filtering.

Despite this grooming procedure, it should be made clear that this technique is designed to be feature agnostic, given a modest discriminative ability on behalf of feature types, the capability for meaningful tokenization, and feature persis-

tence across a reasonable number of images. We acknowledge that the features chosen for this demonstration are ad-hoc, and anticipate a more diverse and considered variety of features in future efforts.

2.2. Co-occurrence context establishment

Our data now consist of a set of frames containing spatially located feature tokens. We use segmentation to propose contexts in which the features are deemed to co-occur.

An ideal segmentation for this application divides the image sequence into a set of contiguous spatio-temporal regions; features occupying the same region are said to co-occur. Methods for such segmentations exist [10], but for this first effort, we treat each frame as independent and use per-frame image segmentation for co-occurrence contexts. In effect, our system only uses spatial coincidence to estimate feature co-occurrence, since the "spatio-temporal" clusters are only a single frame long. Parts of an object which can't be seen simultaneously (e.g. the front and back of a box) are therefore unlikely to be grouped by this initial method.

We used mean-shift image segmentation [11][12] to generate five segmentations for each frame. Each segmentation used different spatial and range bandwidths and minimum region sizes, rendering partitions with a range of coarseness (Figure 1b). Mean-shift segmentation with reduced spatial and range bandwidth parameters was also used to generate the superpixel features in 2.1.

Just as this technique is designed to be feature agnostic, it is also designed to be segmentation agnostic, given a tendency for the segmentations to reflect some actual object boundaries within the data. It seems probable that segmentation wide diversity will promote the discovery of a greater number of meaningful spatio-temporal boundaries. In the future we intend to incorporate affine motion segmentation and other segmentation strategies into this framework.

2.3. Co-occurrence measurement

We can now compute the pairwise co-occurrence likelihood for two features given the per-frame segmentations. Let $f_1 \heartsuit f_2$ denote that features f_1 and f_2 correspond to physical traits co-occurring on the same real-world object. For a particular spatio-temporal segmentation S and input video data D, we estimate the likelihood of $f_1 \heartsuit f_2$ as follows:

$$P(f_1 \heartsuit f_2 | S, D) = \begin{cases} 1 & \text{if } \exists\ t_{f_1}, t_{f_2} \in D \text{ and } r \in S \\ & \text{s.t. } t_{f_1} \in r \text{ and } t_{f_2} \in r, \\ 0 & \text{otherwise,} \end{cases}$$

where t_{f_n} is a feature token locating an observation of feature f_n at some spatio-temporal coordinate in the data, and r is a spatio-temporal region in the segmentation. This is a simple mathematical restatement of the co-occurrence judgment summarized in 2.2.

We compute the pairwise co-occurrence likelihood for a *segmentation type* ST, in this case the full set of per-frame segmentations corresponding to a single configuration of mean-shift parameters, by integrating out each segmentation in ST:

$$P(f_1 \heartsuit f_2 | ST, D) = \sum_{S \in ST} P(f_1 \heartsuit f_2 | S, D) P(S|D).$$

Here, the size of each ST is N, the number of frames in the image sequence, since each frame has its own segmentation. We assume $P(S|D)$ to be uniform, i.e. $1/N$.

The next step is to find an approximation of the pairwise co-occurrence likelihood by integrating out the segmentation types themselves:

$$P(f_1 \heartsuit f_2 | D) \approx \sum_{ST} P(f_1 \heartsuit f_2 | ST, D) P(ST|D).$$

This final approximation is only exact when every possible image segmentation is tried, and when the exact probabilities $P(S|D)$ and $P(ST|D)$ are known. Finding segmentation likelihoods is known to be a difficult task, though there exists some progress in this area [13]. In practice, we find it is effective to assign uniform likelihoods to each segmentation type. In future systems, it may be worthwhile for these likelihoods to be user-configurable, letting the user indicate which type of visual information is more indicative of object boundaries in their data.

The final output of the co-occurrence measurement steps is a symmetric pairwise feature affinity matrix A, where $A_{i,j} \approx P(f_i \heartsuit f_j | D)$.

2.4. Feature clustering

To isolate discrete objects from the feature affinity matrix, we find clusters of strongly connected feature pairs. Spectral techniques exist for clustering based on affinity matrices [14], but here we used the following greedy sampling algorithm to find clusters:

Algorithm FindClusters(F, A)
1 $C \leftarrow$ randomly chosen feature in F
2 **repeat** until C stops changing
3 $F_{\not\heartsuit} \leftarrow$ random permutation of all $F \notin C$
4 **for each** $f_f \in F_{\not\heartsuit}$
5 $J \leftarrow \{f_c \mid f_c \in C \text{ and } P(f_f \heartsuit f_c | D) > \alpha\}$
6 **if** $|J| < \beta |C|$
7 **then** $C \leftarrow \{C, f_f\}$
8 **end for each**
9 **end repeat**

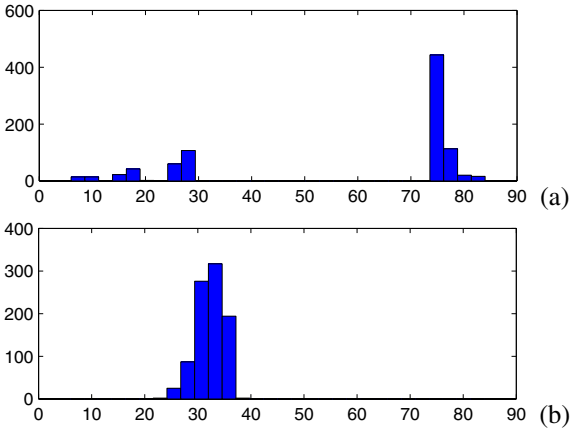

Figure 2. Histograms showing the number of models (y axis) containing a particular number of features (x axis). Plot (a) shows models made using multiple segmentations to establish co-occurrence context; plot (b) shows models made using inter-feature spatial proximity.

Here, α is a connection strength threshold, and β is a connectedness threshold. Both are between 0 and 1. The algorithm incrementally grows the cluster C, adding new members f_f from the set of remaining free features if those features are strongly co-occurrent with a significant fraction of those already in C. In our experiment, α was 0.55 and β was 0.78.

One advantage of this simple method is that it allows clusters to overlap, or share features. This is appropriate when weakly discriminative features like color patches are used, since they can appear on multiple objects. A disadvantage, however, is that different objects are sampled at a frequency proportional to their number of constituent features, due to the random initialization of C. Large objects with lots of appearance variety are more likely to be found than smaller ones, so it is necessary to run FindClusters several times to recover all objects. This is demonstrated in Figure 2a, a histogram showing the number of features in object models found in 1,000 runs of FindClusters our test dataset. Nearly 60% of all recovered clusters describe the same 75-feature object, which has about three times more features than the next largest model.

Just as the output models are distinctively sized in this dataset, it is also the case that they contain distinctive sets of features. Those in the smaller two bumps contain features present on the banjo ukulele; those in the middle group contain features common to the book, and those in the large 75-feature group describe the aviation sectional map laid out on the tabletop. For the detection tasks that follow we selected individual clusters from each group; in future work, we intend to devise a system that analyzes FindClusters output and automatically generates a final set of objects extracted from the training data.

Figure 3 shows feature clusters selected from each group detected and plotted on one of the training set images. Some of the map features are detected on the book and banjo: these features tend to correspond to colors that the objects have in common, or belong to parts of the book and banjo that the segmentations frequently joined with parts of the map. We believe that adding more types of segmentation, such as motion segmentation, will address the latter problem by making difficult-to-find inter-object edges more prominent. Nevertheless, the highest density of map features occurs on the map itself.

2.5. The benefits of segmentation

We can now demonstrate the benefit of using segmentations to establish meaningful contexts for co-occurrence. We remove the co-occurrence probability calculation described in 2.3 and estimate $P(f_1 \diamond f_2|D)$ as the fraction of frames in which any tokens of f_1 and f_2 are within some spatial distance l of each other. In this case, the co-occurrence context is always the spatial vicinity.

Figure 2b shows a histogram of the number of features in the object models discovered by FindClusters when l was set to 50 pixels. Every model contained approximately 33 features corresponding to those in the vicinity of the initializing feature. These features have a low correspondence to any particular object, instead covering portions or all of several objects in the test images (not shown). Setting l to larger or smaller radii increased or decreased the number of features in the object models respectively, though generally this did not improve model/object correspondence.

3. Detection

The main goal of this paper is to show that segmentation can assist in the unsupervised creation of feature-based object models consisting of heterogeneous features. Our object models, consisting chiefly of weakly-discriminative color features, happen to be rather poor for object detection. The banjo ukulele in particular lacks much distinctive texture; its model contains zero SIFT features, and many other objects in the world are made of wood. Detection performance is not necessarily an intrinsic fault or credit of the learning system presented here: by being feature agnostic, it can cluster features of any discriminative strength. Nevertheless, we present some preliminary detection results from using our learned models on test image data.

Figure 3. Original video frame and the frame overlaid with detected feature locations for the three object models. Some map features (lower right) overlap parts of the book and the banjo due to similar colors and incorrect image segmentation.

For each test image we generate a set of feature tokens, as was generated for each image in the training data. We then sample x, y locations in the image, finding the set of features $F_{x,y}$ in a 50-pixel radius of the sample point. For each object O with cluster model C_O, we compute $P(F_{x,y}|O)$ as $|F_{x,y} \cap C_O|/|C_O|$, or the fraction of model features the feature set and the model have in common. No effort was made to weight particular features according to their discriminative power or commonality across object models.

Image intensity maps showing $P(F_{x,y}|O)$ for the three object models are shown in Figure 4. The book detector is particularly effective, since that object contains a small number of bold colors and strong texture cues. The map detector is less so; its colors are common to many real-world objects, and it contains so many features that $|F_{x,y}|$ is usually much smaller than $|C_{\text{map}}|$.

We acknowledge that finding $P(F_{x,y}|O)$ is only a component calculation of practical feature detection, and that our estimation of even this quantity is unsophisticated. Future efforts will incorporate more principled detection strategies.

4. Conclusions and future work

We have demonstrated an unsupervised feature-based object model learning system that uses multiple image segmentations to improve its estimation of feature co-occurrence. Additionally, we presented a four-step schema we hope may be useful for describing further endeavors in this area.

There is room for considerable improvement of the presented method. To enumerate some potential refinements, we return to the four processing stages:

1. **Feature tokenization**. A more diverse array of image features may be used to capture more information about objects.

2. **Co-occurrence context establishment**. Instead of per-image segmentation, actual video segmentation should be used to isolate true spatio-temporal volumes. A wider variety of segmentations should be considered. Finally, it may be worthwhile to find the segmentation likelihoods $P(S|D)$ instead of considering them uniform.

3. **Co-occurrence measurement**. Online estimation of the affinity matrix A, or exploitation of A's sparseness, may be worthwhile investigations.

4. **Feature clustering**. A less heuristic approach to feature clustering may be more robust, and a final step that returns a small set of object clusters is necessary.

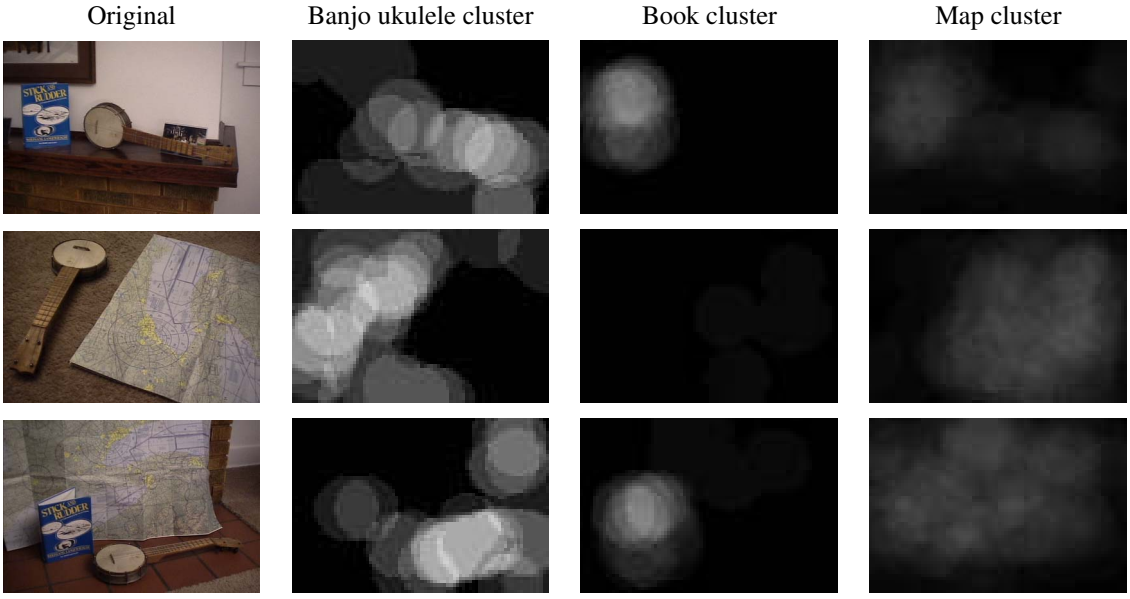

Figure 4. Preliminary detection results for the three learned object clusters. The intensity map images show the estimated per-pixel values of $P(F_{x,y}|O)$; all of them share the same intensity scale. The banjo ukulele detections are strongest since its model contains the least features; the map model contains a large number of features, meanwhile, and its response is weaker.

Acknowledgments

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Thanks to Dave Tolliver for valuable advice.

References

[1] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation." *Computational Linguistics*, 19(2), pp. 263-311, 1993.

[2] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the Society for Information Science*, 41(6), pp. 391-407, 1990.

[3] J. Sivic and A. Zisserman, "Video data mining using configurations of viewpoint invariant regions," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. I, pp. 488-495, 2004.

[4] P. Duygulu, K. Barnard, J. G. G. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: learning a lexicon for a fixed image vocabulary," In *Proc. ECCV*, 2002.

[5] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "Segmenting, modeling, and matching video clips containing multiple moving objects," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. II, pp. 914-921, 2004.

[6] C. Schmid, "Weakly supervised learning of visual models and its application to content-based retrieval," *International Journal of Computer Vision*, 56(12), pp. 7-16, 2004.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60(2), pp. 91-110, 2004.

[8] X. Ren and J. Malik, "Learning a classification model for segmentation," In *Proc. 9th Int. Conf. Computer Vision*, Vol. I, pp. 10-17, 2003.

[9] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *International Journal of Computer Vision*, 56(3), pp. 221-255, 2004.

[10] R. Megret, D. DeMenthon. "A survey of spatio-temporal grouping techniques," LAMP-TR-094, University of Maryland, College Park, USA, 2002.

[11] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 2002.

[12] C. M. Christoudias and B. Georgescu, *Edge Detection and Image Segmentation System (EDISON)*, Source code, http://www.caip.rutgers.edu/riul/research/code/EDISON/, version as of 16 July 2004.

[13] Z. Tu and S. Zhu, "Image segmentation by data-driven markov chain monte carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 2002.

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.