Live Feature Clustering in Video Using Appearance and 3D Geometry

Adrien Angeli
a.angeli@imperial.ac.uk
Andrew Davison
ajd@doc.ic.ac.uk

Department of Computing
Imperial College London
London
SW7 2AZ

Now that good solutions to real-time monocular SLAM exist (e.g. [3]), attention is turning towards what additional 3D information can be extracted live from the same video stream. In this paper our aim is to show that the application of sensibly formulated online feature clustering to the points generated by online SLAM allows useful determination of groups of features which are closely related both in geometry and appearance. We consider this a problem in data-driven perceptual grouping, and the clusters may or may not correspond to individual objects in the scene; however they definitely correspond to characteristic, repeatable structure and could aid in automatic scene understanding, labelling and recognition. Our clustering method proceeds via interleaved local and global processes which permit scalable real-time operation in scenes with thousands of feature points, and has the potential to become part of the standard toolbox of live 3D camera tracking and reconstrution systems.

Single keyframe clustering

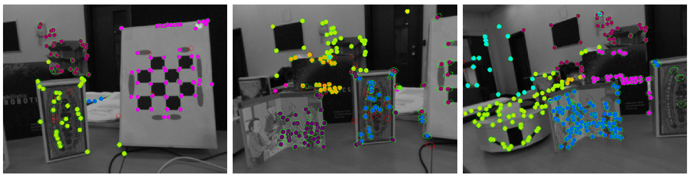


Figure 1: Independent single keyframe clustering results for 3 different keyframes of an indoor video sequence.

We use the monocular SLAM system described by Strasdat *et al.* in [4] which follows the main design of Klein and Murray's PTAM [3]. The output is a live camera pose estimate, and a continuously updated 'map' of the 3D positions of thousands of point features and poses of a representative set of historical keyframes.

In our algorithm, first the features from each single keyframe are clustered, taking into account both appearance and geometry information. Each feature's appearance is characterised using a 1D RGB histogram computed over the 15×15 pixel image patch extracted around the image position of that feature. For geometry information, we compare two features i and j with 3D positions x_i and x_j using the 3D distance between them normalised by the distance to the camera, meaning that the level of cluster detail depends on the distance between the scene and the camera. Following the line of work of [2], we combine appearance and geometry information into a one dimensional similarity measure between features:

$$s(i,j) = \exp\Big(-\frac{d_{\mathrm{RGB}}(i,j)^2}{\sigma_{\mathrm{RGB}}^2} - \alpha\frac{d_{\mathrm{Geom}}(i,j)^2}{\sigma_{\mathrm{Geom}}^2}\Big) , \tag{1}$$

where $d_{\mathrm{RGB}}(i,j)$ is the χ^2 distance between the two histograms. This similarity measure can be interpreted as the probability that any two features i and j belong to the same cluster given their geometrical closeness and similarity in appearance. The complexity of spectral clustering methods as used in [2] currently prevents their use in live video analysis. We therefore have designed a new clustering algorithm in the spirit of Constrained Agglomerative Clustering (CAC) [5]. First, a divisive procedure separates the input data into a set of small yet very consistent clusters. Then an agglomerative algorithm merges the obtained clusters recursively, proceeding by pairs at each iteration until a suitable number of clusters is obtained. We determine the number of clusters automatically using model selection. Figure 1 shows single keyframe clustering results.

Sequence clustering

Next, we use the output of each single keyframe clustering process to maintain, over the whole sequence, a matrix of pairwise co-occurrence probabilities between all the features in the scene. We can then use the *Markov Cluster* (MCL) algorithm [1], one of a class of *Random Walks* methods for determining strongly connected regions of a graph. At convergence, clusters naturally appear in the resulting graph as separate connected components, thus removing the need to specify the numbers of clusters beforehand. The implementation of MCL provided by Dongen [1] takes advantage of the sparseness of the matrix of co-occurrence probabilities to achieve efficient sequence clustering, only requiring a couple of seconds even when the scene contains a few thousand features. Figure 2 shows sequence clustering results, notably showing the consistency of the clustering of a cluttered scene across different viewpoints: as the camera gets closer to the objects, more features are added to the already existing clusters associated with these objects.

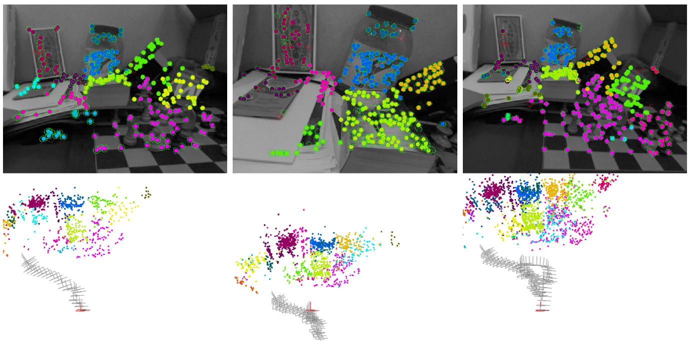
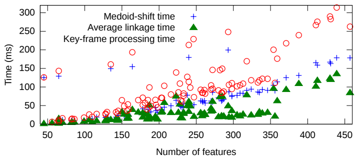


Figure 2: Global, incrementeally computed sequence clustering results for a cluttered scene as the live camera explores.

Computational Performance

All results were obtained in 'live' conditions with all processing on a standard laptop. On average, single keyframe clustering took 99ms, while in the worst case sequence clustering, which is able to run amortized in the background, took 5855ms (with more than 2000 features in the scene). More detailed computation times are given in Figure 3.



Num. of features	Time (ms)
2084	5855
1125	1835
1254	4512
695	1340
566	546
911	3131

Figure 3: **Left:** keyframe computation time vs. number of features in the keyframe. **Right:** sequence clustering computation time vs. total number of features in the scene.

[1] S. Dongen. *Graph clustering by flow simulation*. PhD thesis, University of Utrecht, 2000.

[2] F.J. Estrada, P. Fua, V. Lepetit, and S. Süsstrunk. Appearance-based keypoint clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[3] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[4] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Scale drift-aware large scale monocular SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[5] Y. Zhao and G. Karypis. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10:141–168, 2005.