Grouping from motion using the medoid shift and topological relations

Alexey Chernyavskiy
State Research Institute of Aviation Systems (FGUP GosNIIAS), Moscow, Russia
achern@gosniias.ru

Abstract

We consider the problem of extracting various moving objects on an image by analyzing the topological relations induced by the positions of a sparse set of putative matching points on two images of one scene. Topological relations among triples of matched point pairs are found, and we formulate the task as a hypergraph clustering problem which is then performed by the medoid shift, a non-parametric mode-seeking algorithm similar to mean shift. The method finds the number of clusters automatically and filters out outliers. We report the performance of our method on a synthetic dataset.

Keywords: *topology, medoid shift, clustering, hypergraph, motion segmentation, image matching.*

1. INTRODUCTION

This work addresses the problem of analysis of visual motion. This problem arises in video surveillance as well in other cases when one needs to detect the image changes in order to establish correspondences between pixels across several frames. The image changes can then be interpreted and the 3D structure of the scene can be analyzed. In video tracking the optical flow field is usually available from the analysis of several consecutive frames. The situation is different when the observer moves as well, and when the period of time that elapses between two images is longer.

A common way of finding several independently moving regions of an image is to detect motions by fitting a motion model, such as a homography or fundamental matrix, to the sets of matching points [7]. The sets of points which are explained by the motion model are then removed from consideration and the process is repeated until all the motions are found [2]. In case of a large number of mismatches this approach is not feasible. Whenever a subset of points is removed from consideration, the ratio of noisy matches relative to all the remaining matches increases, and robust filtering methods such as RANSAC [6] become less and less efficient.

The formulation of the problem that is being solved in this work is the following. Given two images and a sparse set of possibly corresponding points cluster the points into groups that belong to separately moving regions of an image. We do so by finding topological relations among pairs of points. By analyzing the topological relations among points one can detect parallax effects which are important depth cues. The topological relations are written down as an affinity matrix of the point pairs, and this matrix is clustered using the medoid shift algorithm. Several independent motions can be recovered simultaneously and the method is robust to outliers.

In Section 2 we formulate the topological relations among pairs of possible point matches, many of which may be outliers. We show that triples of point pairs may be thought of as vertices of a hypergraph. The penalties assigned to a triple whenever at least one point pair within it violates the *sidedness* constraint are the weights of the hypergraph edges. In Section 3 we describe how the *medoid shift* [8], a feature space analysis method similar to the *mean* shift [4], may be used to cluster the affinity matrix derived from the hypergraph using distance values only. The application of medoid shift to hypergraph clustering is given in Section 4. Results on synthetic data are shown in Section 5.

2. TOPOLOGICAL RELATIONS

Topological relations are one of the most stable relations among points of an image. These relations do not change under a wide range of affine transformations that may be applied to an image, such as scaling, translation or rotation. That is why it has been suggested in [5], in the context of image matching, to check whether topological constraints are satisfied among pairs of points and detect outliers. Suppose that we are given two images to be compared and matched, and a set of N corresponding points $\Phi = \{(\mathbf{x}_1^i, \mathbf{x}_2^i)\}_{i=1,\dots,N}$, which are projections of some points in three-dimensional space onto two image planes. For each triple of points $(\mathbf{x}_m^1, \mathbf{x}_m^2, \mathbf{x}_m^3)$ belonging to the same m-th image a *sidedness* function may be computed:

$$side(\mathbf{x}_m^i, \mathbf{x}_m^j, \mathbf{x}_m^k) = \operatorname{sgn}\left(\det\begin{bmatrix} x_m^k - x_m^j & x_m^i - x_m^j \\ y_m^k - y_m^j & y_m^i - y_m^j \end{bmatrix}\right), \qquad (1)$$

where $\mathbf{x}_m^i = (x_m^i, y_m^i)$ and the subscript denotes the number of the image (1 or 2), while the superscript denotes the number of the point. This function assumes the value of +1 if the point $\mathbf{x}_m^i$ is located to the right side with respect to the vector pointing from $\mathbf{x}_m^j$ to $\mathbf{x}_m^k$; otherwise the function takes the value of -1. If, for a triple of points $(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k)$ belonging to the first image and for the triple of corresponding points $(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k)$ located in another image, $side(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k) \neq side(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k)$, we say that the sidedness constraint is violated. In this way, it is possible to derive a penalty function for each point pair:

$$p(i) = \sum_{j,k \in \Phi \setminus i;\, j<k} \left| side(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k) - side(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k) \right| . \qquad (2)$$

By denoting

$$h(\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k) = \left| side(\mathbf{x}_1^i, \mathbf{x}_1^j, \mathbf{x}_1^k) - side(\mathbf{x}_2^i, \mathbf{x}_2^j, \mathbf{x}_2^k) \right| , \qquad (3)$$

the formula (2) may be rewritten as

$$p(i) = \sum_{j,k \in \Phi \setminus i;\, j<k} h(\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k) \ . \qquad (4)$$

The penalty function can be normalized by the maximal number of possible sidedness constraints that each point pair can violate: $p_N(i) = p(i)/[(N-1)(N-2)]$. As a result, the penalty of each pair will belong to [0,1]. An iterative topological filtering procedure has been proposed in [5], in which point pairs (putative matches) having the largest penalty are marked as outliers and removed from consideration. The penalty of the remaining point pairs is then recomputed and the process is repeated until all the pairs' penalties are below a user-specified threshold. Topological filtering has been successfully applied for solving various image matching problems ([5], [3]).

In this work we propose to use topological constraints for segmenting objects that have moved between the moments when two images of one scene have been taken. We assume that the motions which are present on the image may be considered as piecewise-rigid, and so, when one considers pairs of points belonging the surfaces that are moving in a similar way (automobiles, buildings), there are few cases of sidedness violations among such pairs of points. On the contrary, if for a triple of points some points belong to a moving object while other points are part of a structure undergoing a separate movement, then the sidedness constraint has a high likelihood of being violated. Therefore, it seems reasonable to introduce a measure of distance based on topological relations. Point pairs belonging to two views of the same moving will exhibit good topological intra-class affinity, while points belonging to differently moving segments of an image will show a high inter-class dissimilarity.

As can be seen from (3), the topological relations involve triples of points. Since one is able to compute the sidedness violations for all the sets of point triples, the information about the relations that exist among triples can be represented as a hypergraph. A *hypergraph* is a generalization of a graph, where edges can connect any number of vertices instead of just two as in an ordinary graph. Formally, a weighted undirected hypergraph is a pair $H = (X, h)$ where X is a set of vertices, and subsets of X containing k vertices are called hyperedges. The function $h : X^k \to \mathrm{R}^+$ associates non-negative weights to each hyperedge consisting of k vertices. Since the hypergraph is undirected the function h does not depend on the order of its arguments. While there are numerous ways of finding sub-structures within a hypergraph, a common practice is to find a way to reduce the hypergraph to an ordinary weighted undirected graph $G = (X, g)$ over the same set of vertices, with edge weights given by a function $g : X^2 \to \mathrm{R}^+$ of two variables. One of the methods used to approximate a hypergraph with a graph is called clique expansion [1]. For each hyperedge z consisting of k vertices, a k-clique (a completely connected graph on k vertices; $k = 3$ in the case of triadic topological relations) is considered. The task of approximating $h(z)$ is reduced to the task of assigning weights to each edge within the k-clique associated with hyperedge z. According to [1], the weights of the ordinary graph G are given by the formula

$$g(\mathbf{x}^i, \mathbf{x}^j) = \sum_{\mathbf{x}^i, \mathbf{x}^j \in z; \forall \mathbf{x}^k \notin \{\mathbf{x}^i, \mathbf{x}^j\}} h(z) \Big/ \mu(N, k), \tag{5}$$

where $\mu(N,k) = \dfrac{(N-2)!}{(k-2)!(N-k)!}$ is the number of hyperedges that contain a particular pair of vertices. In our case $k = 3$, and therefore $\mu(N,k) = N-2$. The information about the edge weights can be written as a $N \times N$ symmetric matrix $\mathbf{G}$. Having the distance matrix $\mathbf{G}$ on hand, the affinity matrix is defined by

$$\mathbf{W}(i,j) = \exp(-\mathbf{G}^2(i,j)/2\sigma^2) \tag{6}$$

with σ a free parameter.

In the next section a novel algorithm is demonstrated, which allows to cluster data based on its topology-based affinity and remove outliers using the medoid shift.

3. GRAPH CLUSTERING USING MEDOID SHIFT

In this work the task of motion segmentation from two images of a scene is solved by clustering putative matching points into separate sets using an affinity measure computed by using topological relations among triples of matched point pairs. The points in the images can be thought of as vertices of a weighted undirected graph, and the affinity matrix consists of weights associated with the edges of this graph. The graph clustering problem involves data clustering based on the affinity matrix.

There are many efficient methods for data segmentation that make use of eigendecompositions of the affinity matrix (see [9] for a review and analysis). Many of them require specifying the number of clusters explicitly, while in typical computer vision tasks the number of clusters is not known a priori. Although it is possible to infer the number of clusters by analyzing the structure of eigenvectors of the affinity matrix, it involves a complicated numerical method **Error! Reference source not found.**. That is why we turned to a non-parametric mode-seeking technique, the medoid shift, which does not require the number of clusters to be known. Instead, it automatically finds the number of clusters during execution.

The medoid shift [8] is similar to the mean shift, an algorithm which is widely used for data clustering in computer vision problems [4]. We will briefly review both methods. Given N samples denoted by the set $\{\mathbf{x}^i\} \in \mathrm{R}^d$, $i = 1,...,N$, Parzen kernel density estimation is used to evaluate the underlying distribution function at a point by

$$f(\mathbf{x}) = c_o \sum_{i=1}^{N} \Phi\left(\left\|(\mathbf{x} - \mathbf{x}^i)/\sigma\right\|^2\right), \tag{7}$$

where $\Phi(\cdot)$ is a radially symmetric kernel function [4], c_0 is a positive scalar, and $\sigma > 0$ is the bandwidth. In addition, $\Phi(x)$ is the *shadow* of the kernel $\varphi(x)$, i.e. $\varphi(x) = -\Phi'(x)$.

Mode-seeking is the process of finding local maxima of the density of the data. It is assumed that modes are good candidates for being centers of clusters. During mode-seeking, each point is initially denoted by $\mathbf{y}^0$, and the set of intermediate points traversed on the way to the mode is denoted by $\{\mathbf{y}^k\} \in \mathrm{R}^d$, $k = 1,...,K$. Each step of mean shift moves along the direction of highest gradient from the current point. Given the current point $\mathbf{y}^k$, its position on the next iteration of the method is denoted by $\mathbf{y}^{k+1}$ and found according to

$$\mathbf{y}^{k+1}_{mean} = \arg\min_{\mathbf{y}} \sum_{i=1}^{N} \left\|\mathbf{x}^i - \mathbf{y}\right\|^2 \varphi\left(\left\|(\mathbf{x}^i - \mathbf{y}^k)/\sigma\right\|^2\right). \tag{8}$$

By differentiating the above equation and setting the first derivative to zero, one can obtain the formula for the position update of a point:

$$\mathbf{y}_{mean}^{k+1} = \sum_i \mathbf{x}_i \varphi\left(\left\| (\mathbf{x}^i - \mathbf{y}^k)/\sigma \right\|^2 \right) \Big/ \sum_i \varphi\left(\left\| (\mathbf{x}^i - \mathbf{y}^k)/\sigma \right\|^2 \right). \qquad (9)$$

In this way, the updated position of a point is the weighted mean of the sample points. The mean shift method converges when the position $\mathbf{y}_{mean}^{k+1}$ does not change over the course of several iterations. By applying the mean shift procedure to each of the sample points, one can obtain modes of the data. Points that converged to the same modes are said to be part of the same clusters. The data on which the mean shift (and medoid shift) operate do not have to be linearly separable or form clusters of spherical shape with sharp boundaries. This allows the mean shift to work well in various tasks of computer vision, such as image segmentation, discontinuity preserving smoothing [4], and object tracking.

The medoid shift [8] is the extension of mean shift. An advantage of the medoid shift over the mean shift is that it can be applied to cases when the notion of mean is not defined and/or the mean of data points cannot be readily computed. Medoid shift finds modes of the data even when only a distance measure between samples is defined. This is precisely the case in the current work. The medoid, which is an extension of the median in the one-dimensional case (it is defined as the most centrally located point in a set of samples), is always part of the initial dataset. The update rule (8) is transformed and becomes

$$\mathbf{y}_{medoid}^{k+1} = \arg\min_{\mathbf{y}\in\{\mathbf{x}_i\}} \sum_{i=1}^{N} \left\| \mathbf{x}^i - \mathbf{y} \right\|^2 \varphi\left(\left\| (\mathbf{x}^i - \mathbf{y}^k)/\sigma \right\|^2 \right). \qquad (10)$$

The difference between (8) and (10) is that in the former case the updated position is a data point taken from the *d*-dimensional search space, while in the latter case $\mathbf{y}_{medoid}^{k+1}$ is the data *sample* that minimizes the function. In other respects, the mode-seeking procedure is the same as in the mean shift algorithm. The medoid shift provides a straightforward way of handling the outliers. Typically, outliers are located 'far' (in terms of topology-based affinity) from other data, and they do not merge with any clusters other than the ones they are centers of. A simple threshold T on the cardinality of a cluster allows differentiating between good clusters, consisting of many elements, and small clusters (containing only one data point in the extreme case) which are deemed as outliers. In this work we used the value $T = 10$. Numerically, the medoid shift is much faster than the mean shift because the search space is greatly reduced. The numerical implementation of the medoid shift is straightforward (see [8] for details). We use (6) with $\sigma = 0.2$ for computing the matrix $\mathbf{W}$.

4. PROPOSED METHOD

Before proceeding to the outline of the proposed algorithm, it should be made clear that formula (4) is not optimal for computing the topology-based distances between pairs of putative matches because it is highly affected by contributions from point pairs that are located far away from the point pair under consideration. Suppose that one needs to compute the distance $g(i, j)$ between pairs $\mathbf{x}^i$ and $\mathbf{x}^j$. Suppose that these pairs belong to the same moving object or to the image background and are close to each other. A third point $\mathbf{x}_1^k$ and its probable match $\mathbf{x}_2^k$ may be outliers (mismatches) or pseudo-outliers (true matches that belong to another moving object) and this may lead to sidedness violation and affect the affinity between $\mathbf{x}^i$ and $\mathbf{x}^j$.

Assuming that moving surfaces that are to be segmented from the images are local, we propose to weigh points' contribution to the affinity measure according to their geometrical distance from the pair under consideration. Precisely, the modified formula for $g(i, j)$ is the following:

$$g(\mathbf{x}^i, \mathbf{x}^j) = \sum_{\mathbf{x}^i, \mathbf{x}^j \in z; \forall \mathbf{x}^k \notin \{\mathbf{x}^i, \mathbf{x}^j\}} f\left[h(z)\right] \Big/ (N-2), \qquad (11)$$

where

$$f\left[h(z)\right] = \begin{cases} 0, & \text{if } d_m^{ijk}/d_m^{ij} > t, \\ 1, & \text{if } d_m^{ijk}/d_m^{ij} < t, \quad \forall m \in \{1,2\} \\ h(z), & \text{otherwise,} \end{cases} \qquad (12)$$

and $d_m^{ij} = \left\| \mathbf{x}_m^i - \mathbf{x}_m^j \right\|$, $d_m^{ijk} = \min(d_m^{ik}, d_m^{jk})$, $m \in \{1,2\}$.

In other words, the sidedness violations arising from the points which are far away from the pair for which the topological distance is computed are ignored. The modification given by (11) and (12) enhances the structure of the affinity matrix, leading to better intra-class similarity and larger inter-class dissimilarity. In this work we have used the value $t = 2$. The distance matrix for various values of t is shown in Figure 2 for a test problem described in the next section.

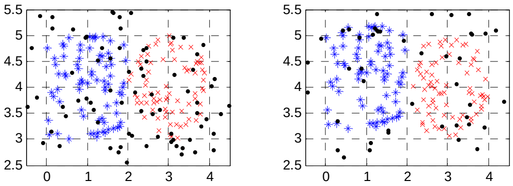

Figure 1: A synthetic example involving a square and a circle undergoing translation and rotation.

The outline of our algorithm is the following. Given σ, T, t, and N putative point matches of two images of the same scene :

1. Compute sidedness violations (3) for each triple of corresponding point pairs.
2. Compute the distance matrix using the sidedness violations and formulas (11) and (12). Compute the affinity matrix using (6).
3. Find modes of the data by running the medoid shift algorithm on the affinity matrix $\mathbf{W}$. Data points that converged to the same mode are part of a common cluster. Prune the clusters that contain fewer than T elements. They are considered as outliers. Clusters that contain more than T elements will correspond to surfaces of objects that undergo various movements.

5. RESULTS

We created a synthetic example consisting of two moving objects shown in Figure 1. The first object consists of 75 points randomly placed within a square of size 2. The second object is consists of 75 points randomly placed inside a circle of radius 1 centered at coordinates (3,4). In a second image, the square is moved by 0.2 in the y-direction, while the circle is rotated by 45° clockwise about its center. Our goal was to segment these two motions in the presence of 75 noisy points, a noise level of 33%.

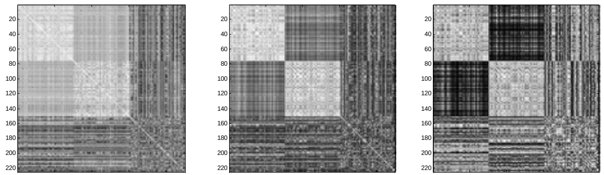

Figure 2: Distance matrix for the model from Figure 1 with various values of t **:** $t = 2$ (center) and $t = 1$ (right). The threshold t was not used in the left image. Lighter color indicates higher similarity among points.

In Figure 2 we illustrate the impact of the parameter t on the structure of the distance matrix. For illustration purposes, the points are numbered according to the number of object they are part of. When t is not used (Figure 2, left), the points exhibit a high similarity in-between clusters which makes it hard to differentiate between objects. When t is very small (Figure 2, right), the inter-cluster similarity is low, but the intra-cluster similarity becomes lower as well, since sidedness violations for even nearby points are ignored. We demonstrate in Figure 3 the two segmented objects (only the right frame is shown). Notice that while some points were incorrectly segmented, their number is low, and the number of correct points that were not thrown out is quite high. For comparison, we also show the results produced by the mean shift. For the mean shift, each data sample consisted of four parameters: the coordinates of each point in the first frame, and the two velocity components (difference of coordinates between the putative matches). The search domain for each parameter was normalized to lie between 0 and 1, and $\sigma = 0.1$ was used. A total of seven clusters were found, but much more outliers remain. This happens because both translation and rotation are present in the image, but the mean shift fails to take it into account.

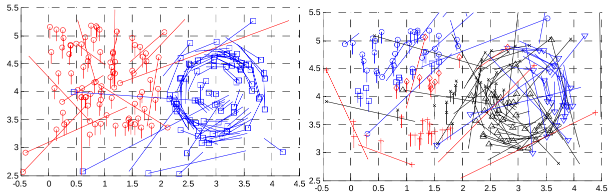

Figure 3: Two structures segmented by our algorithm (left) and seven clusters found by the mean shift (right).

6. CONCLUSION

In this work we have presented a method to extract various moving regions from topological relations among pairs of putative matches found using two images of a scene. The main contribution of our work is that we have formulated the topological relations in the contest of a hypergraph. The affinity matrix of the corresponding ordinary graph is then clustered using medoid shift. Clusters correspond to areas of the image that undergo independent motions, which is indicative of independently moving objects. Outliers (wrong matches) are detected as well. Compared to [5], the topological clustering introduced in this work leads to a smaller number of pairs that are wrongly marked as outliers, since it does not assume global scene rigidity.

The method was compared to a mean shift implementation that took into account only the shift vectors (velocities) of the matched points, without using the topological relations. Our method outperformed the mean shift implementation on synthetic data.

In the future we plan to validate our method on real datasets, as well as investigate the role of parameters σ and t which should be chosen dynamically based on local dataset density.

7. REFERENCES

[1] S. Agarwal, Lim Jongwoo, L. Zelnik-Manor, P. Perona, D. Kriegman and S. Belongie: Beyond pairwise clustering, In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, pp. 838-845, 2005.

[2] P. Bhat, K.C. Zheng, N. Snavely, A. Agarwala, M. Agrawala, M.F. Cohen, M.F. Curless and B. Curless: Piecewise Image Registration in the Presence of Multiple Large Motions, In *IEEE Conference on Computer Vision and Pattern Recognition*, New York, USA, pp. 2491-2497, 2006.

[3] Yu.B. Blokhinov, D.A. Gribov and A.S. Chernyavskiy: Image matching problem for certain cases of perspective photography, *Journal of Computer and Systems Sciences International*, vol. 47, pp. 959-973, 2008.

[4] D. Comaniciu and P. Meer: Mean shift: a robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603-619, 2002.

[5] V. Ferrari, T. Tuytelaars and L. Van Gool: Wide-baseline multiple view correspondences, In *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, USA, pp. 718-725, 2003.

[6] M.A. Fischler and R.C. Bolles: Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, vol. 24, pp. 381-395, 1981.

[7] R. Hartley and A. Zisserman: Multiple View Geometry in Computer Vision. – Cambridge University Press, 2004. - 672 p.

[8] Y.A. Sheikh, E.A. Khan and T. Kanade: Mode-seeking via Medoidshifts, In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, pp. 1-8, 2007.

[9] Y. Weiss: Segmentation using eigenvectors: a unifying view, In *IEEE International Conference on Computer Vision*, Kerkyra, Greece, pp. 975-982, 1999.

About the author

Alexey Chernyavskiy received his specialist degree in Applied Mathematics from Moscow State University in 2000. In 2003 he completed his M.S. degree in Geophysics at the University of Utah, USA. He now works at the State Research Institute of Aviation Systems (GosNIIAS) in Moscow, Russia. His scientific interests include image matching in the presence of large motions and pattern recognition. His contact email is achern@gosniias.ru .