Perceptual Grouping from Motion Cues Using Tensor Voting

1. Research Team

Project Leader: Prof. Gérard Medioni, *Computer Science*

Graduate Students: Mircea Nicolescu, Changki Min

2. Statement of Project Goals

The project addresses the problem of visual motion analysis by establishing a voting-based computational framework. We present a novel approach for matching and motion capture that recovers the dense velocity field, motion boundaries and regions from a sequence of images, based on a 4-D layered representation of data, and a voting scheme for token affinity communication. We then interpret these layers in order to generate a full dense 3-D structure of the scene, with independently moving objects segmented. Our current work involves extending the formalism to multiple frames, and improving computational efficiency.

3. Project Role in Support of IMSC Strategic Plan

Our ability to extract velocity information, as well as motion regions and boundaries from image sequences can be very useful to other research projects at IMSC. It can be used by other projects that involve motion capture, avatars, face and body animation, and video compression.

4. Discussion of Methodology Used

We propose a novel approach based on a layered 4-D representation of data, and a voting scheme for token communication. Our methodology is formulated as a 4-D Tensor Voting computational framework. The position (x,y) and velocity (v_x,v_y) of each token are encoded as a 4 tuples. By propagating this information through voting, distinct moving regions emerge as smooth surface layers in the 4-D space of image coordinates and pixel velocities. Although a pair of images is enough in most cases, multiple images also can be used to increase robustness and to solve some difficult cases such as occlusion by incorporating temporal smoothness; a pair of images allows only spatial smoothness, whereas multiple images enable us to take advantage of temporal smoothness.

4.1. Motion Analysis in an Image Pair

4.1.1. Generating Candidate Matches

We take as input two image frames that involve general motion – an input image is shown in Figure 1(a). For every pixel in the first image, the goal at this stage is to produce candidate matches in the second image. We use a normalized cross-correlation procedure, where all peaks of correlation are retained as candidates. Each candidate match is represented as a (x, y, v_x, v_y) point in the 4-D space of image coordinates and pixel velocities, with respect to the first image.

Since we want to increase the likelihood of including the correct match among the candidates, we repeat this process at multiple scales, by using different correlation window sizes.

The resulting candidates appear as a cloud of (x, y, v_x, v_y) points in the 4-D space. Figure 1(b) shows a 3-D view of the candidate matches - the 3 dimensions shown are x and y (in the horizontal plane), and v_x (the height). The motion layers can be already perceived as their tokens are grouped in smooth surfaces surrounded by noisy matches.

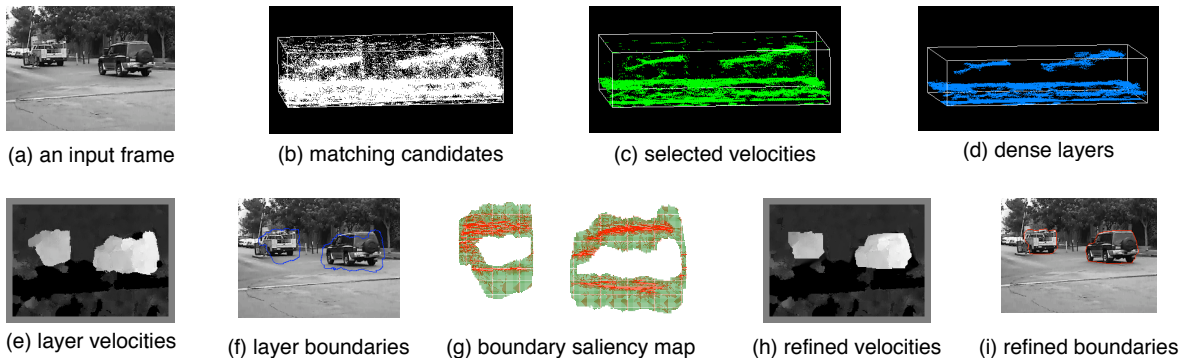


Figure 1. Barrier sequence

4.1.2. Extraction of Motion Layers

Within our 4-D representation, the smoothness constraint is embedded in the concept of *surface saliency* exhibited by the data. By letting the tokens communicate their mutual affinity through voting, noisy matches are eliminated as they receive little support, and distinct regions are extracted as smooth, salient surface layers.

Selection. Since no information is initially known, each potential match is encoded into a 4-D ball tensor. Then each token casts votes by using the corresponding ball voting field. During voting there is strong support between tokens that lie on a smooth surface (layer), while isolated tokens receive little or no support. For each pixel we retain the candidate match with the highest surface saliency, and we reject the others as outliers. Selected matches are shown in Figure 1(c).

Orientation refinement. In order to obtain an estimation of the layer orientations as accurate as possible, we perform an orientation refinement through another voting process, but now with the selected matches only. After voting, the eigenvectors give the local layer orientations at each token. The remaining outliers are also rejected at this step, based on their low surface saliency.

Densification. Since the previous step created holes (i.e., pixels where no velocity is available), we infer this information from the neighbors by using a smoothness constraint. This is performed through an additional dense voting step, by generating discrete velocity candidates, collecting votes at each such location, and retaining the candidate with maximal surface saliency. By following this procedure at every image location we generate a dense velocity field. A 3-D view of the dense layers (the height represents v_x) is shown in Figure 1(d).

4.1.3. Boundary Inference

After grouping the tokens into regions, based on the smoothness of both velocities and layer orientations, it becomes apparent that the extracted layers may still be over or under-extended along the true object boundaries. Figure 1(e) illustrates the recovered v_x velocities within layers (dark corresponds to low velocity), and Figure 1(f) shows the layer boundaries superimposed on the first input image.

This situation typically occurs in areas subject to occlusion, where the initial correlation procedure may generate wrong matches that are consistent with the correct ones, and therefore could not be rejected as outlier noise. However, the key observation is that one should not only rely on *motion cues* in order to perform motion segmentation. Examining the original images reveals a multitude of *monocular cues*, such as intensity edges, that can aid in identifying the true object boundaries.

The boundaries of the extracted layers give us a good estimate for the position and overall orientation of the true boundaries. We combine this knowledge with monocular cues (intensity edges) from the original images in order to build a boundary saliency map within the uncertainty zone along the layers margins. The smoothness and continuity of the boundary is then enforced through a 2-D voting process, and the true boundary is extracted as the most salient curve within the saliency map. Finally, pixels from the uncertainty zone are reassigned to regions according to the new boundaries, and their velocities are recomputed. Figure 1(g) shows the boundary saliency map, while Figure 1(h) shows the refined velocities within layers, and Figure 1(i) shows

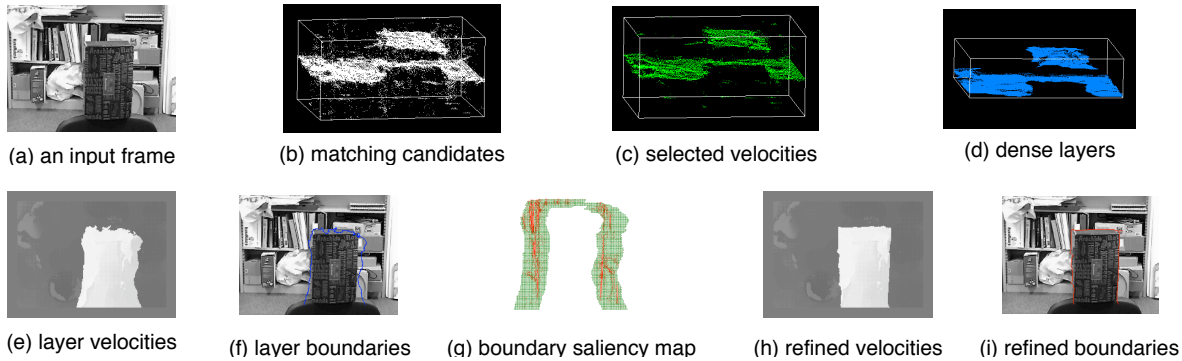


Figure 2. Candy box sequence

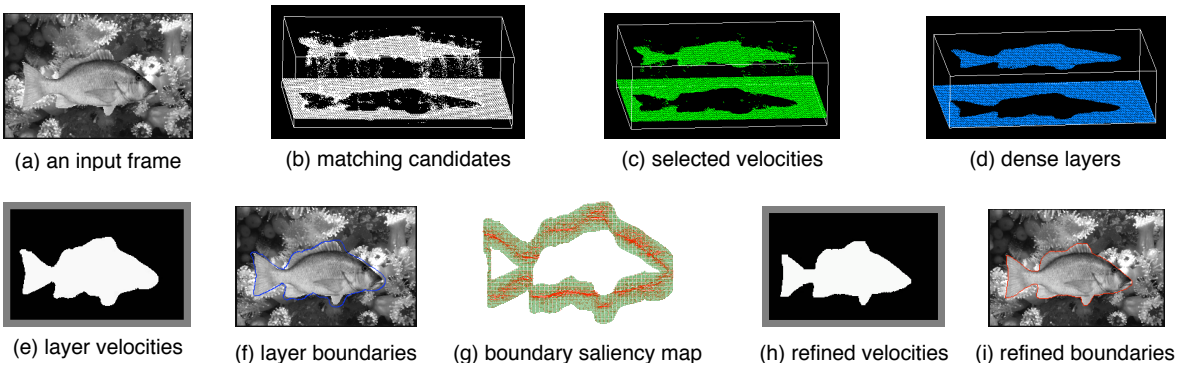


Figure 3. Fish sequence

the refined motion boundaries, that indeed correspond to the actual objects.

4.1.4. Motion Interpretation

There are many approaches to recover the epipolar geometry and 3-D scene reconstruction from the given two images, and most of them require the following two conditions:

- the set of matches must contain no outlier noise
- the scene must be rigid.

The first condition easily fails because no method can perfectly find accurate correspondences mainly due to occlusion or lack of texture, and the second condition is also hard to be held since most scenes have multiple objects, which move independently.

Within our approach both conditions can be easily satisfied. After the tensor voting processes, outliers are eliminated from the initial matches so that the resulting matches are very reliable. Since the correspondence and segmentation processes take place prior to 3-D interpretation, the second condition also can be satisfied.

Here, we use the algorithm introduced by McReynolds and Lowe [12] to verify the potential rigidity. The rigidity test is performed on a subset of matches within each object, to identify potential non-rigid objects, and also across objects, to merge those that move rigidly together but have distinct image motions due to depth discontinuities. It is also worth mentioning that the rigidity test is actually able to only guarantee the non-rigidity of a given configuration. Indeed, if the rigidity test fails, it means that the image motion is not compatible to a rigid 3-D motion, and therefore the configuration must be non-rigid. If the test succeeds, it only asserts that a possible rigid 3-D motion exists, that is compatible to the given image motion.

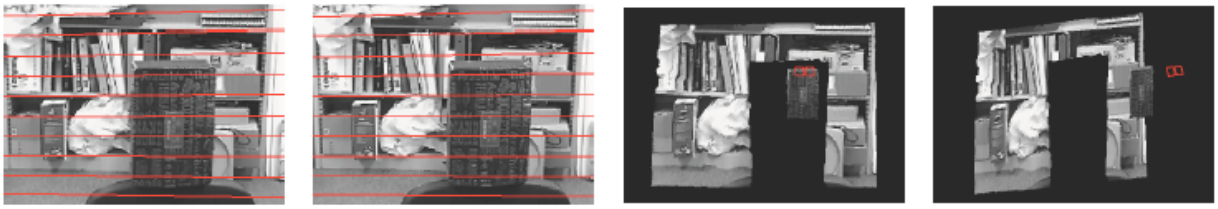
The remaining task at this stage is to determine the object (or camera) motion, and the scene structure. Since wrong matches have been eliminated, and correct matches are already grouped according to the segmented objects in the scene, standard methods for reconstruction can be reliably applied. For increased robustness, we choose to use RANSAC [13] to recover the epipolar geometry for each rigid object, followed by an estimation of camera motion and projective scene structure.

4.1.5. Results (Image Pair)

The examples in Figure 2 and Figure 3 illustrate the performance of our approach for boundary inference in a cluttered environment, where texture edges strongly compete with the true object edges. Through voting, the saliency of the spurious texture edges has been diminished by the overall dominance of saliency and orientation of the correct object edges.

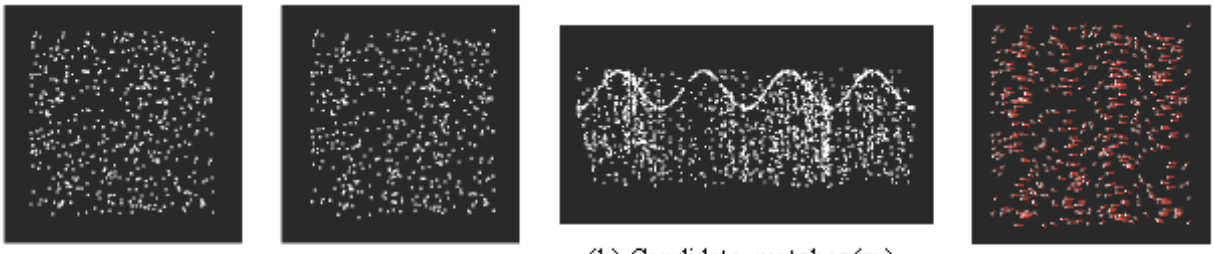
Figure 4 is an example of ‘single rigid motion’. This is the stereo case where the scene is static and the camera is moving. Due to the depth disparity between the box and the background, their image motions do not satisfy the smoothness constraint together, and thus they have been segmented as two separate objects. However, the rigidity test shows that the two objects form a rigid configuration, and therefore are labeled as a single object. The epipolar geometry estimation and scene reconstruction are then performed on the entire set of matches.

Figure 5 shows the case of non-rigid motion. This flag scene is a synthetic sequence where sparse random dots from the surface of a waving flag are displayed in two frames. The configuration is recognized as non-rigid, and therefore no reconstruction is attempted. However, since the image motion is smooth, our framework is still able to determine correct correspondences, extract motion layers, segment non-rigid objects, and label them as such.

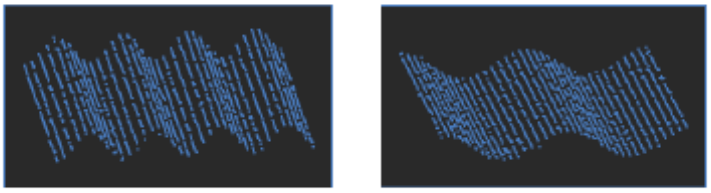


(a) Epipolar lines (b) 3-D structure and motion

Figure 4. Reconstruction of the candy box sequence



(a) Input images (b) Candidate matches (v_x) (c) Velocities



(d) Dense layers (v_x) (e) Dense layers (v_y)

Figure 5. Reconstruction of the flag scene

4.2. Image Sequence

4.2.1. Generating Candidate Matches and Selection

A real image sequence might have huge amount of frames so that processing all the images at once is not an efficient approach when we process multiple images. Rather, by using a sliding window whose size is much smaller than the number of total images, we can do sequential batch processing for the given image sequence.

Assume that the sliding window size is t, and the following analysis will assume only those t images contained in one instance of the sliding window. The image in the middle (temporal sense) is set to the reference image, and $(t-1)$ pairs of images are generated by pairing the reference image with the rest of the images. Figure 6 shows the set of images of the Teddy bear [11]. The sequence consists of 7 images and the 3rd one is the reference image. The camera moves horizontally and the movements are equidistant meaning that each pixel in the reference image moves at its constant speed.

The same *candidate match* and *selection* processes described in 4.1.1. and 4.1.2., respectively, are separately applied to each pair of images to enforce spatial smoothness in the same way. As a result, every pixel (x,y) of the reference image should have a single match in each of other $(t\text{-}1)$ images after the *selection* process. Figure 7 shows the resulting selected matches; the leftmost one is the match between the reference image and the first image, and the next one is the match between the reference image and the second image, and so forth. We can observe that image border areas and object boundaries are quite noisy due to occlusions, and low texture areas such as the right side of the teddy bear also produce many wrong matches.

Figure 6. Input image sequence (Teddy bear)

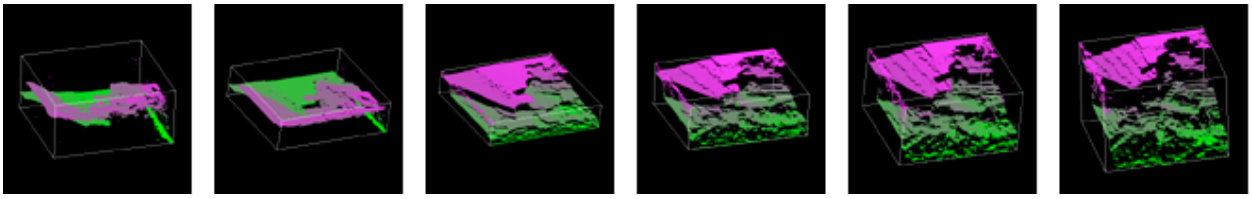


Figure 7. (x,y,vx) point sets after the 'selection' process

4.2.2. Motion Layer Refinement

The temporal smoothing constraint is implemented as follows. Each pixel (x,y) in the reference image is uniquely coupled with $(t\text{-}1)$ pixels of the rest of $(t\text{-}1)$ images after the *selection* process. With the assumption of smooth motion the set of matches forms a smooth curve in (x,y,t) space. Since not all matches are correct due to, for example, occlusion or low texture around a given pixel, the initial curve may not appear smooth. The temporal smoothness is enforced by finding the best curve, which fits to the set of $(t\text{-}1)$ points in (x,y,t) space for each pixel of the reference image.

The outlier rejection process is quite different from the case of a pair of images. Instead of setting a global saliency threshold which determines whether the given match is outlier or not, the outlier is determined within the set of $(t\text{-}1)$ matches at each (x,y) and the threshold is computed based on the statistics of the $(t\text{-}1)$ saliency values. After removing the outlier matches, curve fitting is performed to the remaining matches. Experimentally, three different approaches were implemented; 1) weighted least square, 2) weighted spline, 3) RANSAC with first or second order polynomial curve. In the case of 1) and 2), the saliency values are used as weights. If a smooth curve is found, then correct matches are estimated based on the curve.

Figure 8 illustrates the smoothing process for some pixels in the reference image. The horizontal axis is the image number, each of which is coupled with the reference image, and the vertical axis is the displacement. The blue line with asterisk marks indicates the initially selected matches, the red line with circle marks indicates the smooth curve (here, RANSAC with first order polynomial curve model was used), and the green square indicates the ground-truth match

(only the pair of the reference image and the last image has ground-truth data). Note that each smooth curve is found after removing some outliers as described above meaning that the wrong matches (blue asterisks which obviously appear as outliers in the following figure) do not affect the process of finding a smooth curve, since they have low saliencies.

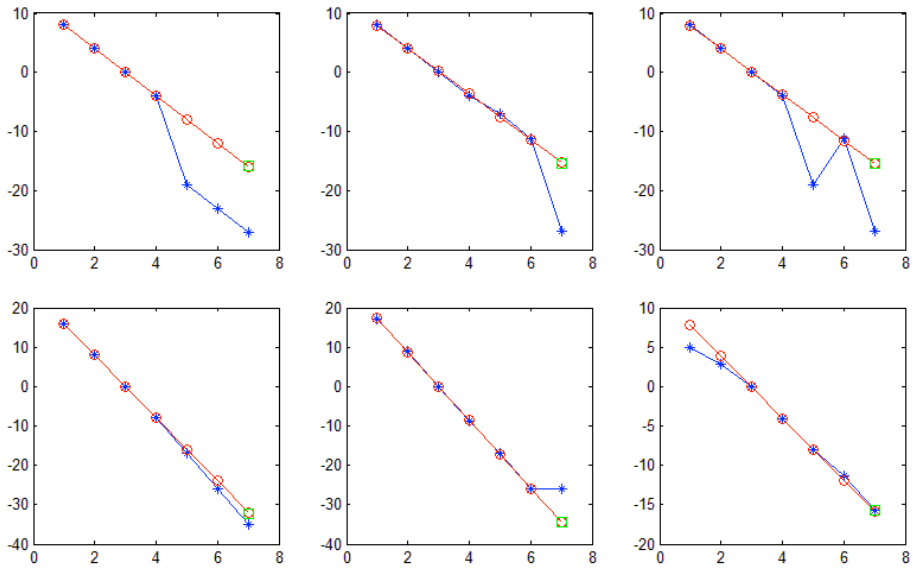


Figure 8. Smoothing process

4.2.3. Results (Image Sequence)

For the case of an image sequence, we do not perform boundary inference process. Rather, the disparity map between the reference image and the last image is computed, and it is compared with the given ground-truth data. Figure 9 shows the disparity maps; the first one is the ground-truth, the second one is the initial selection, the third one is the one after the temporal smoothing process (white areas indicate undefined disparities). It is worth noting that the noisy object boundaries and the large occluded area in the left side of the image have been significantly improved by the temporal smoothing. Table 1 also shows the quantitative error rate comparison with other approaches [10][11]. Although other methods use only two images, the error rate of the proposed multiple image approach shows a promising result.

Figure 9. Disparity maps (ground-truth, initial selected matches, temporally smoothed)

Approaches	Error Rate
Tensor Voting with multiple images (TV-M)	**8.1 %**
Tensor Voting with two images (TV-2)	15.4 %
Sum of Squared Differences (SSD)	26.5 %
Dynamic Programming (DP)	30.1 %
Graph Cuts (GC)	29.3 %

Table1. Error rate comparison (disparity error larger than 1 pixel)

5. Short Description of Achievements in Previous Years

During the initial phase of this research, the core of the voting-based computational framework had been developed and implemented. This includes finding a suitable representation for the motion layers, as smooth surfaces in 4-D, and establishing an efficient mechanism for token communication. Also we developed a consistent methodology for integrating both monocular and motion cues within the tensor voting computational framework. Combining motion information with intensity cues, such as texture edges, is crucial for the accurate segmentation of real image sequences.

5a. Details of Accomplishments During the Past Year

To increase the accuracy and robustness, a new approach which uses multiple images rather than a pair has been experimented. By using multiple images, we could enforce temporal smoothness along with the spatial smoothness. As a result, most occlusions could be handled in a proper way and the accuracy of the matches could be improved as well.

6. Other Relevant Work Being Conducted and How this Project is Different

Optical flow techniques [4] rely on local, raw estimates of the flow field to produce a partition of the image. However, the flow estimates are very poor at motion boundaries and cannot be obtained in uniform areas. Past approaches have also used Markov Random Fields [5] or regularization techniques to handle discontinuities [6].

Significant improvements have been achieved by using layered representations [7]. The difficulties range from a severe restriction in motion representation (as rigid or planar), to over-fitting and instability due to high-order parameterizations.

A computational framework that successfully enforces the smoothness constraint in a unified manner, while preserving smoothness discontinuities is Tensor Voting [8]. The first to propose using Tensor Voting for motion analysis were Gaucher and Medioni [9]. They employ successive steps of voting, first to determine the boundary points as tokens with maximal motion uncertainty, then to locally refine velocities on each side of the boundary. However, their voting communication is essentially a 2-D process that does not inhibit neighboring elements with different velocities from influencing each other.

Our approach proposes a 4-D space for a layered representation, and a voting-based scheme for token communication. This allows for a spatial separation of the points according to both velocities and image coordinates. Consequently, the proposed framework allows tokens from the

same layer to strongly support each other, while inhibiting influence from other layers or from isolated tokens. Despite the high dimensionality, our voting scheme is both time and space efficient. It is non-iterative and the only free parameter is scale, which is an inherent characteristic of human vision.

7. Plan for the Next Year

Although the proposed multiple image approach shows good results, it is still in an experimental state, and needs more work on the smoothing process. Better methods for detecting outliers and finding smooth curves, as well as a more *efficient* algorithm will be studied in the next year. Also the method which allows us to use layers in (*t-1*) pairs for better motion interpretation will be investigated. The performance of the new algorithm will also be carefully analyzed by comparing it with other approaches.

8. Expected Milestones and Deliverables

After we refine the temporal smoothing procedure and make it more efficient, we are going to apply this algorithm to real image sequences. Therefore, we expect more publications not only in computer vision areas but in multimedia-related conferences and journals. Also we expect that the tensor voting framework which is optimized in the efficiency sense will contribute to other vision applications.

9. Member Company Benefits

N/A

10. References

[1] M. Nicolescu and G. Medioni, "Layered 4-D Representation and Voting for Grouping from Motion", to appear in the *IEEE Trans. on PAMI - Special Issue on Perceptual Organization in Computer Vision*, 25:4, pp. 492 – 501, 2003.
[2] M. Nicolescu and G. Medioni, "4-D Voting for Matching, Densification and Segmentation into Motion Layers", *ICPR*, vol. 3, pp. 11-15, 2002.
[3] M. Nicolescu and G. Medioni, "Perceptual Grouping from Motion Cues Using Tensor Voting in 4-D", *ECCV*, vol. III, pp. 423-437, 2002.
[4] J. Barron, D. Fleet, S. Beauchemin, "Performance of Optical Flow Techniques", *IJCV*, 12:1, pp. 43-77, 1994.
[5] F. Heitz, P. Bouthemy, "Multimodal Estimation of Discontinuous Optical Flow Using Markov Random Fields", *PAMI*, 15:12, pp. 1217-1232, December 1993.
[6] S. Ghosal, "A Fast Scalable Algorithm for Discontinuous Optical Flow Estimation", *PAMI*, 18:2, pp. 181-194, 1996.
[7] S. Hsu, P. Anandan, S. Peleg, "Accurate Computation of Optical Flow by Using Layered Motion Representations", *ICPR*, pp. 743-746, 1994.
[8] G. Medioni, Mi-Suen Lee, Chi-Keung Tang, "A Computational Framework for Segmentation and Grouping", *Elsevier Science*, 2000.

[9] L. Gaucher, G. Medioni, “Accurate Motion Flow Estimation with Discontinuities”, *ICCV*, pp. 695-702, 1999.
[10] Daniel Scharstein and Richard Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light", *CVPR*, pp. 195-202, 2003.
[11] http://cat.middlebury.edu/stereo/data.html
[12] D. McReynolds, D. Lowe, “Rigidity Checking of 3D Point Correspondences Under Perspective Projection”, Trans. *PAMI*, 18:12, pp. 1174-1185, 1996.
[13] P.H.S. Torr, D.W. Murray, “The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix”, *IJCV*, 24:3, pp. 271-300, 1997.