

Motion Segmentation with Occlusions on the Superpixel Graph

Alper Ayvaci Stefano Soatto
University of California, Los Angeles
{ayvaci, soatto}@cs.ucla.edu

Abstract

We present a motion segmentation algorithm that partitions the image plane into disjoint regions based on their parametric motion. It relies on a finer partitioning of the image domain into regions of uniform photometric properties, with motion segments made of unions of such “superpixels.” We exploit recent advances in combinatorial graph optimization that yield computationally efficient estimates. The energy functional is built on a superpixel graph, and is iteratively minimized by computing a parametric motion model in closed-form, followed by a graph cut of the superpixel adjacency graph. It generalizes naturally to multi-label partitions that can handle multiple motions.

1. Introduction

Changes of vantage point and the motion of objects in a scene induce a deformation of the domain of the image, and estimating this motion field has been a central problem in Computer Vision for many years. Horn and Schunck’s pioneering work [15] proposed approximating such “optical flow” by a smooth vector field on the entire image domain. More modern renditions of their program include robust regularization that enable preserving motion discontinuities [31, 5], as well as motion segmentation techniques that partition the domain into regions where the motion is well approximated by a low-dimensional parametric model [11]. Although in principle such techniques can handle an arbitrary number of regions, in practice the optimization becomes non-convex and rather laborious as soon as there are more than two independently moving objects in the scene [10]. There has also been work combining motion with intensity and texture cues [3].

In this manuscript, we show that one can, without loss of generality, approximate a given motion field to an arbitrary degree by a partition of the image domain within which motion is constant, and then show that such a domain can be described as the union of a finer partition based on photometric characteristics of a static image. Then we formulate the ensuing problem, that yields a multi-phase partitioning

of the image domain into regions of constant motion, as a graph optimization problem, and exploit recent techniques that yield fast and accurate results. This framework enables us to handle multiple moving objects at no additional cost.

2. Motion Estimation and Segmentation

Let $I(x, t) : D \subset \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ denote a time-varying image. Assuming a Lambertian scene seen under constant illumination, it is common to model the domain deformation induced by scene motion by assuming the existence of a motion field $v : D \rightarrow \mathbb{R}^2; x \mapsto v(x)$ such that $I(x, t + dt) = I(x + v(x)dt, t) + n(x, t)$, where the \mathbb{L}^2 -norm of the residual n is assumed to be small. In the limit where $n = 0$ and $dt \rightarrow 0$, this is called the *brightness constancy constraint* (BCC). As usual in functional approximation problems, one has the choice of approximating v globally (on the entire image domain D) with a very complex function, for instance a general (infinite-dimensional) diffeomorphism [28], or restricting the class of models significantly (e.g. translational $v(x) = d \forall x$ or affine fields $v(x) = Ax + d$), and then partitioning the domain into regions, which we call Ω_i , in which the model fits the data within a specified accuracy. So, as long as the field is square-integrable, either choice can make the residual n arbitrarily small (for instance in the sense of \mathbb{L}^2). The first approach has been pioneered by Horn and Shunck [15], while the second approach has been dubbed *motion competition* [11]. Even more popular are local optical flow approaches that arbitrarily select both the model class and the domain partition Ω_i , typically made of rectangular windows of fixed size [2]. These approaches, while attractive from a computational standpoint, do not offer any guarantee of approximating the original motion field, assuming there was a “true” one to begin with. Indeed, typically there is not. In fact, even the previous two approaches are only valid on a subset of the image domain D , away from occluding boundaries. Occluded regions $\mathcal{O} \subset D$ are those visible in one image but not in the other. In these regions, $v(x)$, $x \in \mathcal{O}$ is simply not defined, as there is no possible correspondence. Various approaches have been proposed to address this issue [17, 27, 4], along with techniques to preserve discontinuities

in the vector field (by using Total Variation as a regularizer, rather than \mathbb{L}^2 , [1, 24, 12, 18, 16]). Most methods for dealing with occlusions that we are aware of can be reduced to the optimization of variants of the following functional:

$$\hat{v} = \arg \min_{v, \mathcal{O}} \underbrace{\int_{D \setminus \mathcal{O}} (I(x, t + dt) - I(x + v(x)dt, t))^2 dx}_{BCC} + \lambda_1 \underbrace{\int_D \|\nabla v\|_1 dx}_{\text{regularization}} + \lambda_2 \underbrace{\int_{\mathcal{O}} (\varepsilon + \|\nabla I\|_2^2) dx}_{\text{occlusion}}. \quad (1)$$

where $\nabla I \doteq [\frac{\partial I}{\partial x_1} \ \frac{\partial I}{\partial x_2}]$ denotes the gradient of the image (a row vector), $I_t \doteq \frac{\partial I}{\partial t}$, $\lambda_1, \lambda_2, \varepsilon$ are weighting coefficients. There are several possible variants of this model depending on the choice of regularization or occlusion penalties including just the area $\int_{\mathcal{O}} \varepsilon dx$, rather than the area weighted by the gradient of the image.

The alternative model would simultaneously attempt to partition the domain into regions Ω_i , with $\cup_i \Omega_i = D$, within which the vector field is constant, say $v = v_i$:

$$\{\hat{v}_i\}_{i=1}^N = \arg \min_{v_i, \Omega_i, \mathcal{O}} \sum_{\Omega_i \cap \mathcal{O} = \emptyset} \int_{\Omega_i} (I(x, t + dt) - I(x + v_i dt, t))^2 dx + \lambda_2 \int_{\mathcal{O}} (\varepsilon + \|\nabla I\|_2^2) dx. \quad (2)$$

Note that there is no need for an explicit regularization term, assuming N is finite. The advantage of this formulation is that, since each v_i is constant in Ω_i , it can be solved for in closed-form given the region Ω_i ; omitting the arguments for simplicity, and using the derivative notation to denote first differences, we have

$$\int_{\Omega_i} (\nabla I v_i + I_t)^2 dx = v_i^T \underbrace{\left(\int_{\Omega_i} \nabla I \nabla I^T dx \right)}_{G(\Omega_i)} v_i + 2 \underbrace{\int_{\Omega_i} I_t \nabla I dx}_{E(\Omega_i)} v_i + \underbrace{\int_{\Omega_i} I_t^2 dx}_{F(\Omega_i)} \quad (3)$$

so $v_i(\Omega_i) = G^{-1}(\Omega_i)E(\Omega_i)$, where G is called the *second-moment matrix*, which has to be invertible, as we discuss shortly. So, the problem becomes that of finding the partition $\{\Omega\}$ and the occlusion \mathcal{O} . This can be cast as a region-based segmentation problem, and solved using variational techniques in the framework of Level Set methods [21]. This is easy to do for the case of two regions (e.g. “foreground” and “background”), because it can be cast as a convex optimization problem [10]. Unfortunately, in the presence of multiple motions (three or more), the problem becomes non-convex, and approaches involving logical combinations of level set functions quickly become unmanage-

able. For this reason, although developed in full generality, variational motion segmentation approaches are mostly used for binary (foreground/background) classification.

Consider a partition of the image domain sub-partitioned into regions that have constant color or gray level to within a specified accuracy ϵ ,

$$S_i \mid \cup_i S_i = D \quad \text{and} \quad \left| I(x, t) - \frac{\int_{S_i} I(x, t) dx}{\int_{S_i} dx} \right| \leq \epsilon$$

where the distance from the mean can be interpreted as a proxy for constancy, and can be generalized to a small gradient norm as a proxy for continuity. This partitioning can be easily obtained with a number of “superpixel” oversegmentation algorithms, for instance [30]. It is immediate to see that the partition $\{\Omega_i\}_{i=1}^N$ must be a super-set of the partition $\{S_i\}_{i=1}^M$, in the sense that, at least away from occluding boundaries, each Ω_i must be a union of S_j 's. If this was not the case, $I(x, t)$ would be constant (approximately within ϵ), and therefore $\nabla I(x, t) \simeq 0 \forall x \in \Omega_i$, and the corresponding v_i would be undefined, in the sense that v_i would have no effect in the BCC component of the cost $\nabla I v_i + I_t = 0 v_i + I_t = I_t$. Therefore, it makes sense to restrict the partition $\{\Omega\}$ to be made of unions of superpixels. Note also that this guarantees that the second-moment matrix $G(\Omega_i)$ is invertible, as previously mentioned.

We now address the issue of occlusions. In general, there will be a subset of superpixels S_j that are entirely occluded, but it is also possible for a superpixel region to be only partially occluded. In other words, at occlusions, the motion regions Ω_i can split the superpixels S_j . However, these are easily detected, for in this case $\nabla I(x, t) = 0$, $x \in S_j$, but at the same time $I(x, t + dt) - I(x + v_j dt, t) > 0$, by the occlusion assumption (unless the occluder region has the same graylevel as the occluded region, in which case it cannot be detected as an occlusion anyway). An accurate solution of the functional (2) will have high residual at occluded regions, therefore, the superpixel S_j is better off contributing to the occlusion region $S_j \subset \mathcal{O}$.

If a finer partition is desired, so partially occluded superpixels are further segmented into an occluded and an unoccluded part, one can compare the intensity at time t with that at $t + 1$ restricted to the superpixels that are detected as occluded; by assumption, the occluding boundaries in each image will be captured by the oversegmentation (illusory contours, or boundaries between regions with identical photometric properties will, again, not be detected).

In the next section, we illustrate our approach to carrying out the minimization of the functional (2) using recently introduced graph-based optimization methods, and in Section 4 we show the results of our approach on representative image data, including quantitative comparisons with *motion competition* of [11].

3. Implementation

In this section, we describe our implementation of the minimization (2) by restricting the regions Ω_i and \mathcal{O} to be unions of superpixels S_j . We have implemented both the case of translational motion as well as affine motion, which we compare in Sect. 4.

3.1. Superpixelization

We use the publicly available Quick Shift [30] to extract superpixels at multiple scales from the input frames. We use a three-dimensional representation of each pixel, the concatenation of the intensity value and its location within the image. Unlike normalized-cut [25] superpixelization techniques, the number of the superpixels are not fixed, since the size and the number of superpixels is directly related to the complexity of the scene. The outcome of the process in [30] is determined by three parameters: γ , the trade-off between the intensity and the position of the pixel, σ , the scale at which the density is estimated, and τ , the maximum distance in feature space between pixels in the same cluster. The parameter used for the experiments are $\sigma = 1, \gamma = 0.25$ and $\tau = 3$. A low value of σ can be used to arrive at a conservative superpixelization.

Once the finer partitioning of the image is computed, the superpixel graph is constructed. Each node in the graph represents a label. An edge is added between any two adjacent neighbors. Furthermore, the features defined on regions such as mean and standard deviation of the pixel intensities are stored on the nodes while the lengths of the shared boundaries between two regions are stored on the edges.

3.2. Motion Segmentation Functional on Superpixel Graph

We now write the data portion of the cost functional (2) directly in terms of the superpixels; we also include the shortest boundary length regularization which separates the partitions:

$$\{\hat{v}_i\}_{i=1}^N = \arg \min_{v_i, \Omega_i} \sum_{\Omega_i} \sum_{S_j \subset \Omega_i} \int_{S_j} \left(\nabla I v_i + \frac{\partial I}{\partial t}(x, t) \right)^2 dx + \sum_{\Omega_i} \int_{\partial \Omega_i} ds, \quad (4)$$

where s denotes arc-length. In the next section we describe the minimization of this functional.

3.3. Energy Minimization

We use an alternating minimization scheme to optimize (4). For each given partition Ω_i , we first compute the motion in closed form as defined in Section-2. For the case of

affine motion, we follow the lines of [11], and solve for the parameters in closed form.

Given motion estimates for all the regions Ω_i , we can address the optimization of $\{\Omega_i\}$ using the techniques developed in [14]. Since both the BCC and boundary regularization terms are sub-modular, we can solve (4) for $\{\Omega_i\}$ as a max-flow/min-cut computation [19] which gives an optimal solution in the case of only two regions, foreground and background. The first term in (4) describes unary terms penalizing the optical flow residual, and the second term penalizes the contour length and can be written in terms of pairwise entities on the edges. Note that if we put the shared boundary length of two neighboring superpixels as an entity on the edges, a minimal-cut computed on the graph gives the total shared boundary length. This value is basically the length of the boundary separating two regions, see Figure 1.

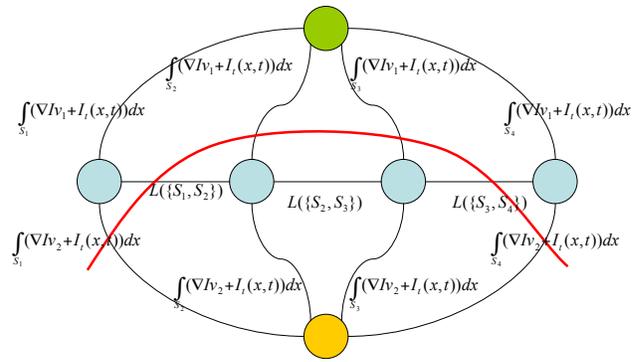


Figure 1: **Graph Cuts optimization:** The red cut separates the superpixels $\{S_1, S_4\}$ from $\{S_2, S_3\}$. It is clear that the cut does the summation through the residuals which is the first term of our motion segmentation functional. In addition, it sums the shared boundary lengths $L(\{S_1, S_2\})$ and $L(\{S_3, S_4\})$ which is the boundary length separating these two partitions.

3.4. Dealing with Multiple Classes

Our optimization scheme generalizes naturally to partitioning of the image plane when multiple motions exist. While α -expansion and α/β -swap algorithms provide an optimal solution for binary graph cuts minimization, they also provide an approximate solution to the multi-labelling problem on graphs [9, 19, 7].

3.5. Occlusion Detection

Theoretically, the superpixels at the occlusion should not be assigned any flow since there is no correspondence. Nevertheless, in practice, we start the minimization of (2) with the assumption that there is no occlusion, and apply the motion segmentation algorithm explained in this section. In the

next step, given $\{\Omega_i, v_i\}$ pairs, occluded regions that minimize the energy (2) can be detected. In the case that a superpixel S_j is fully occluded, it is straightforward to mark it as occluded since any flow vector v_j assigned will lead to high residual. However, for a partially occluded superpixel S_k , the residual computed on the pixels inside S_k can be small unless v_k is the *correct* motion that would have been estimated in the case that S_k was never occluded.

To overcome this problem, we take advantage of intensity as a coupling term in our motion segmentation algorithm. When the intensity similarity term $\exp(-\beta(\mu_j - \mu_k)^2)$, where μ_j and μ_k are the intensity means of the superpixels S_j and S_k and β is the coefficient, weights the edges of superpixel graph, the graph cuts labeling will favor the assignment of the same motion model to the superpixels with similar intensity profile. This approach is similar to that of other occlusion detection algorithms [33, 26, 17] which rely on anisotropic diffusion to recover from the error that occluded regions cause. Once we have a proper motion segmentation, we can detect fully or partially occluded superpixels easily using graph cuts minimization. Subsequently, we split the partially occluded superpixels, and exclude them together with the fully occluded ones from the domain \mathcal{D} . We repeat this procedure, alternating between the motion segmentation and occlusion detection, until the energy converges.

In our experiments, we use area based occlusion penalty $\int_{\mathcal{O}} \varepsilon dx$, rather than the area weighted by the gradient of the image.

4. Experiments

Figure 2 illustrates the features of our algorithm on a synthetic example. In this test video, a puzzle-piece shaped region of a carpet texture is sliding to the left while the rest of the image is moving to the right. The clustering algorithm extracted 1,352 superpixels in the first frame of the test video. The algorithm successfully segments the puzzle piece in 3 iterations, each taking about 10ms, starting from a random initialization in a bounding box. So, assuming that we are given the superpixels, our algorithm finalizes the segmentation in 30ms. Superpixelization on this image takes about 400ms in Matlab, so the overall cost is still a fraction of that of competing motion segmentation schemes.

Such a small number of iterations is made possible by the optimal moves in the graph-cuts minimization and the fast computation of the graph cut that relies on the superpixelization. Also, note that the initial bounding box does not need to be close to the object to have a good segmentation.

In the following experiments, we have used two image sequences recorded by D. Koller and H.-H. Nagel

(KOGS/IAKS, University of Karlsruhe)¹. First, we have tested our algorithm using both a translational and an affine motion model and compared them in Figure 3. In this example, a white car is turning to the right at the corner. Segmentation and motion estimation are shown on the top, relative to the translational model, and on the bottom relative to the affine model. Both are computed in 4 iterations, each taking 10ms for the translational model, and 60ms for the affine one.

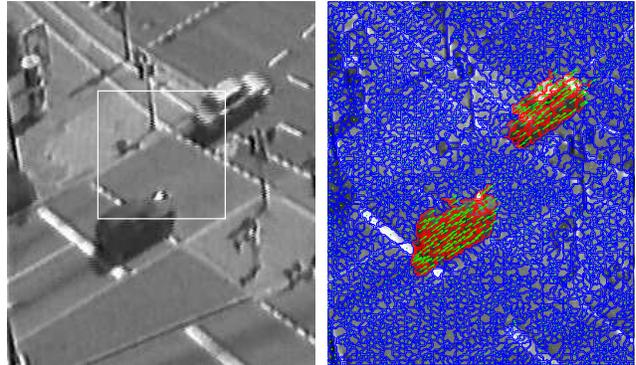


Figure 4: **Topological changes:** Two distinct objects sharing the same motion are shown. Although our algorithm is initialized with a single object, it accurately labels the scene since it allows splitting.

We also test the segmentation of two distinct object moving with the same motion, Figure-4. Even though we initialize our algorithm with a single object, since the graph cuts allows topological changes such as splitting and merging, our motion segmentation algorithm accurately labels these distinct objects.

In the next experiment, we test the estimation of multiple motions. Graph cut minimization in this case provides a more natural way of generalizing the approach to multi-label segmentation, via a multi-way cut. Although there is no polynomial time optimal solution for this problem, an approximate solution is possible using the algorithms proposed by [8].

The multi object motion segmentation for the frame-13 of the Taxi sequence is illustrated at Figure 5. In this frame, the white car is still turning to the right (object 1) while another black car is entering the scene from the left (object 2), and the background is, for the most part, moving with trivial (zero) motion, nevertheless, its motion is estimated (object 3). The segmentation and affine motion estimation are accurate for both of the cars, and are a slight improvement on the translational model.

Observe the “halo” effect, whereby an object includes, as part of its motion, regions belonging to the uniform back-

¹http://i21www.ira.uka.de/image_sequences/

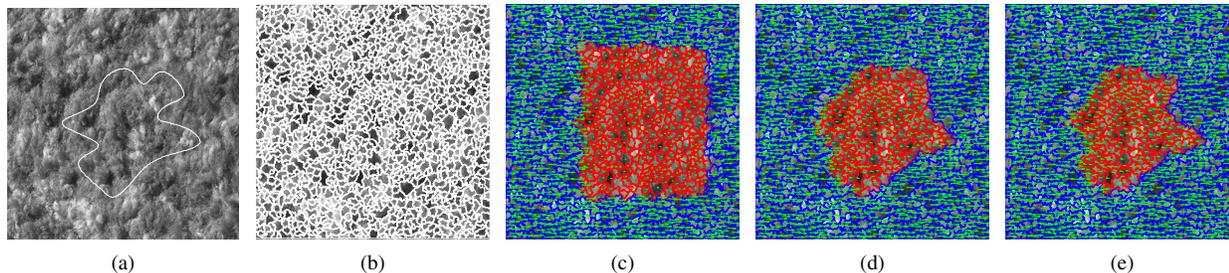


Figure 2: **Synthetic Example:** A puzzle-shaped part of the carpet is sliding to the left while the rest is moving to the right. (a) The puzzle shape. (b) Superpixels (c),(d),(e) Motion segmentation at the first (initialization), second and third (and last) iterations, respectively. The red superpixels describe the foreground while the blue ones describe the background. The green arrows, starting from the centroid of the superpixels, represent the velocity of the superpixels.

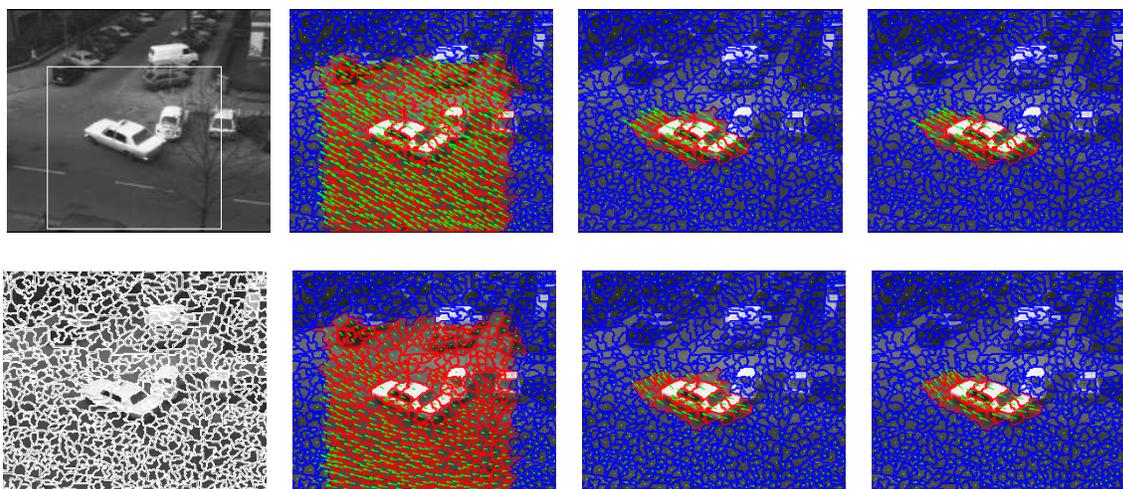


Figure 3: **Translational vs. Affine model:** Motion segmentation for frame-13 of the “Taxi” sequence. (first column) Initial bounding box on the reference frame and superpixels. (column 2,3,4) Evaluation of the segmentation regions and the motion fields for both translational and affine motion models, in iteration one, two, and four. The affine motion model yields a spatially-varying field on the turning taxi, but almost identical segmentation. Note that part of the background pavement is attributed to the car. One could disambiguate split superpixels as described in the text and in Figure- 7, although in this case, due to the constant background, technically one could not know for sure whether the car is indeed traveling with a “halo” underneath.

ground. This can be eliminated by increasing the weight of the boundary length regularizer, as shown in Figure-6. Several authors have addressed this issue, including using combinations of features [23, 22]. However, partially occluded superpixels still remain as a problem which will be solved with the proposed occlusion detection algorithm.

In the next experiment, the proposed occlusion detection method is tested on frame-1 and frame-13 of the Taxi sequence, Figure-7. Since the motion of both cars are slow and the road beneath is smooth, in both cases the superpixels at the boundary are occluded only partially. Our approach successfully detects them, and splits to sub-regions

that are subsequently labeled as occlusions.

In order to arrive at a quantitative comparison, we use [11] as a reference, we generated multiple sequences similar to that in Figure 2 (with different foreground and background texture), by randomizing the shape of the foreground region and its motion under noise levels ranging from 1% to 15%. We measure error by averaging the set-symmetric difference between the estimated foreground region and the ground truth, normalized by the area of the true foreground. This yields a figure of merit between 0 and 1. Our algorithm presents superior robustness against salt and pepper noise compared to [11], Figure 8. While the error of

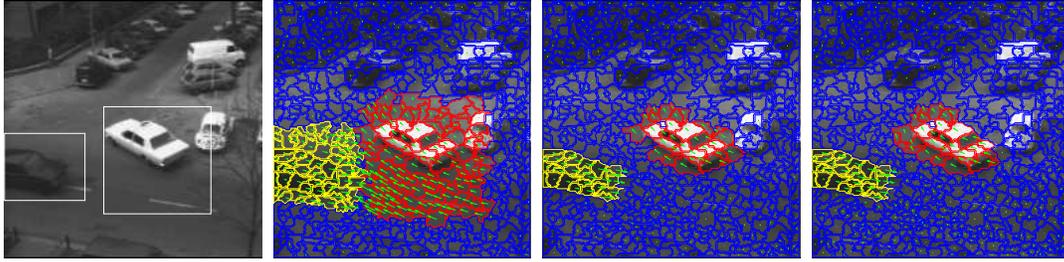


Figure 5: **Multiple Motion Segmentation:** Three objects are shown: The white vehicle, the black vehicle, and the background. This poses no significant difficulty in our approach, that successfully estimates the motion and boundary of the objects.

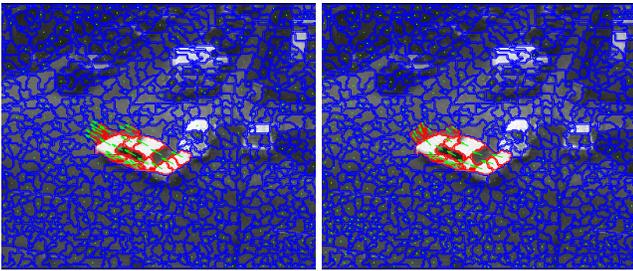


Figure 6: **Effect of boundary regularization:** The “halo” effect discussed in Figure- 3 can also be easily eliminated by increasing the weight on the boundary length regularizer; the effect is shown for frame 1 of the Taxi sequence relative to the translational and affine motion model.

our algorithm is 0.08 ± 0.01 at the 8% noise level, [11] has achieved a poor accuracy with the error of 0.37 ± 0.01 , and could not converge at the higher noise ratios.

Since we adopt a partition-based approach to functional estimation, the accuracy of the estimate is reflected in the partition, and therefore we have chosen to compare the set-symmetric difference, rather than the flow vector as one would do in a plain optical flow algorithm.

Then we have measured the *robustness* to initialization by computing the percentage of trials where the final solution at convergence was within expected error ratio. This clearly depends on the initial conditions. While in our case the algorithm is not particularly sensitive to initialization, [11] exhibited some sensitivity, performing best with an initialization that was partially overlapping the object of interest. In the absence of noise, both algorithms achieved 100% convergence. However, at 3% noise level, [11] converged only 66% of the times within the error of 21%, whereas our algorithm converged 98% of the times within the error of 7%. Furthermore, we have tested our algorithm alone at the noise level 10%, and it has managed to converge at 96% of

the trials.

Finally, as the computational efficiency was discussed earlier, our approach converges to a solution with high accuracy much faster than [11]. It should also be noted that we are restricting this simulation experiment to two regions, otherwise the approach [11] would require multi-phase level set implementation at significantly increased computational complexity.

5. Discussion

It may be at first seem surprising that we can outperform pixel-based motion segmentation schemes, since we constrain the regions to be unions of superpixels. However, the partition imposed by superpixels is non-committal from the point of view of motion estimation, for it is a partition within which the second-moment matrix is not invertible, and therefore there is no added benefit in further subdividing these regions, for instance into pixels. On the other hand, the computational advantage of operating on superpixels is evident in the computational improvements. Although a vast literature on optical flow estimation exists, with benchmark data sets [2], our model is more powerful than any of the (fixed-block partition) models thus described. There are also other approaches based on Markov Random Fields [20] deterministic relaxation [6], normalized cuts [25] and expectation-maximization [32] that are significantly slower than our approach.

Our approach is different than other methods utilizing superpixels for motion segmentation [29] and motion estimation [35]. The algorithm, proposed in [29], exhaustively matches superpixels in two frames and solves the motion segmentation problem on the *pixel* domain where the superpixel matching is a motion prior. In the case of [35], the authors propose a method for constructing consistent segments (superpixels) across the frames where the segments that are matching to each other have similar shape and appearance. Once the consistent segments are constructed, it is straightforward to estimate the motion relying on this cor-

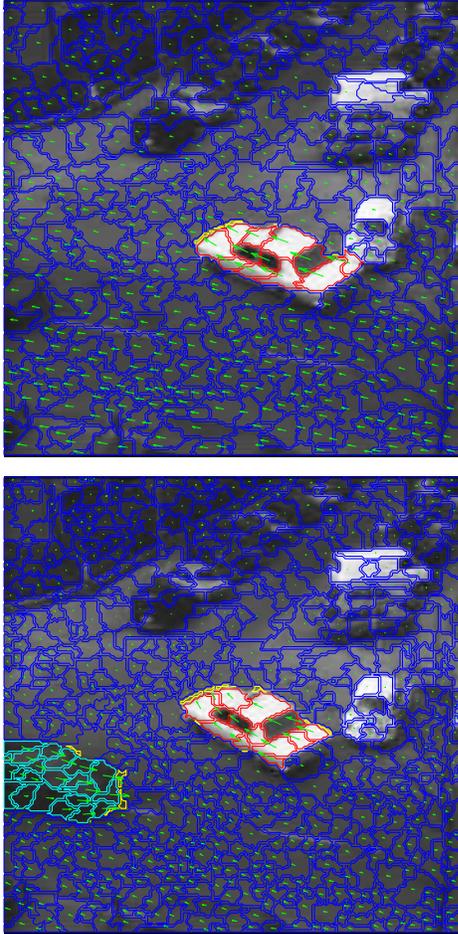


Figure 7: **Handling Occlusions:** Here we illustrate the disambiguation of boundary superpixels that are split by an occlusion, using intensity as a coupling term in the segmentation. We show the final segmentation for frame 1 and frame 13 respectively, which can be seen to eliminate the “halo” underneath the vehicles. The superpixels that are splitted, and then marked as occluded are represented with yellow boundaries.

respondence.

We are able to obtain our result, also thanks to recent advances in combinatorial optimization schemes for solving variational problems of the kind (2) [19, 25, 34, 13], and in particular [14].

Of course, our model has limitations. Since it relies on superpixels, if superpixels fail to (over)segment the image, and instead include within each region S_j significant photometric variability, our algorithm will fail. Therefore, a key to the successful use of our model consists in choosing a conservative threshold for the superpixels.

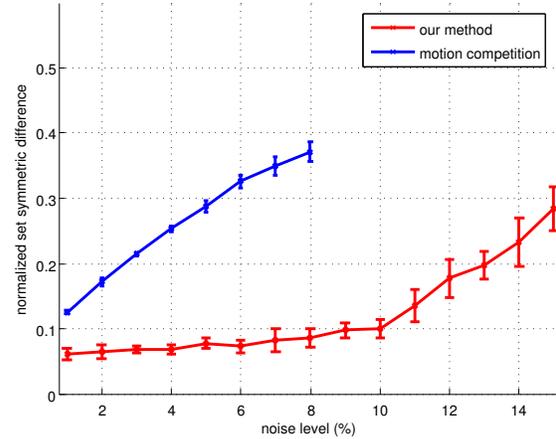


Figure 8: **Accuracy:** This figure presents the comparison between the accuracy of both algorithms under different noise levels.

Acknowledgments

This research is supported by AFOSR FA9550-09-1-0427 and ONR N00014-08-1-0414.

References

- [1] L. Alvarez, J. Weickert, and J. Sánchez. A scale-space approach to nonlocal optical flow calculations. In *Proc. of International Conference on Scale-Space*, pages 235–246, 1999.
- [2] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.
- [3] M. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *Proc. European Conference of Computer Vision*, pages 485–493. Springer, 1992.
- [4] M. Black. Explaining optical flow events with parameterized spatio-temporal models. In *Proc. of Conference on Computer Vision and Pattern Recognition*, volume 1, 1999.
- [5] M. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.
- [6] P. Bouthemy and E. Francois. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, 1993.
- [7] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
- [8] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, 2001.

- [9] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [10] T. Chan and S. Esedoglu. Aspects of Total Variation Regularized L-Function Approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817, 2005.
- [11] D. Cremers and S. Soatto. Motion competition: a variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.
- [12] R. Deriche, P. Kornprobst, and G. Aubert. Optical-flow estimation while preserving its discontinuities: a variational approach. In *Proc. Second Asian Conference on Computer Vision*, pages 290–295, 1995.
- [13] N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the Mumford-Shah model. In *Proc. of the Int. Conf. Visualization, Imaging, and Image Processing*, 2007.
- [14] L. Grady and C. Alvino. Reformulating and optimizing the mumford-shah functional on a graph — a faster, lower energy solution. In *ECCV 2008*, volume 5302, pages 248–261, 2008.
- [15] B. Horn and B. Schunck. Determining optical flow. *Computer Vision*, 17:185–203, 1981.
- [16] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representations. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1621–1626, 1992.
- [17] S. Ince and J. Konrad. Occlusion-aware optical flow estimation. *IEEE Transactions on Image Processing*, 17(8):1443–1451, 2008.
- [18] S. Ju, M. Black, and A. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 307–314, 1996.
- [19] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.
- [20] J. Konrad and E. Dubois. Bayesian estimation of motion vector fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.
- [21] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 1988.
- [22] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.
- [23] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2003.
- [24] C. Schnörr. Computation of discontinuous optical flow by domain decomposition and shape optimization. *International Journal of Computer Vision*, 8(2):153–165, 1992.
- [25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [26] C. Strecha, R. Fransens, and L. Van Gool. A probabilistic approach to large displacement optical flow and occlusion detection. In *Proceedings of Statistical Methods in Video Processin*, 2004.
- [27] J. Sun, Y. Li, S. Kang, and H. Shum. Symmetric stereo matching for occlusion handling. In *Proc. of Conference on Computer Vision and Pattern Recognition*, volume 2, 2005.
- [28] G. Sundaramoorthi, P. Petersen, V. S. Varadarajan, and S. Soatto. On the set of images modulo viewpoint and contrast changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.
- [29] A. Thayananthan, M. Iwasaki, and R. Cipolla. Principled fusion of high-level model and low-level cues for motion segmentation. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [30] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *Proceedings of the European Conference on Computer Vision*, 2008.
- [31] J. Weickert, B. Romeny, and M. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.
- [32] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 520, Washington, DC, USA, 1997. IEEE Computer Society.
- [33] J. Xiao, H. Cheng, S. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *Proceedings of the European Conference on Computer Vision*, 2006.
- [34] X. Zeng, W. Chen, and Q. Peng. Efficiently solving the piecewise constant mumford-shah model using graph cuts. Technical report, Technical report, Dept. of Computer Science, Zhejiang University, PR China, 2006.
- [35] C. Zitnick, N. Jovic, and S. Kang. Consistent Segmentation for Optical Flow Estimation. In *Proc. of International Conference on Computer Vision*, volume 2, 2005.