Dense and Deformable Motion Segmentation for Wide Baseline Images

Juho Kannala, Esa Rahtu, Sami S. Brandt and Janne Heikkilä

Machine Vision Group, University of Oulu, Finland
{jkannala,erahtu,sbrandt,jth}@ee.oulu.fi

Abstract. In this paper we describe a dense motion segmentation method for wide baseline image pairs. Unlike many previous methods our approach is able to deal with deforming motions and large illumination changes by using a bottom-up segmentation strategy. The method starts from a sparse set of seed matches between the two images and then proceeds to quasi-dense matching which expands the initial seed regions by using local propagation. Then, the quasi-dense matches are grouped into coherently moving segments by using local bending energy as the grouping criterion. The resulting segments are used to initialize the motion layers for the final dense segmentation stage, where the geometric and photometric transformations of the layers are iteratively refined together with the segmentation, which is based on graph cuts. Our approach provides a wider range of applicability than the previous approaches which typically require a rigid planar motion model or motion with small disparity. In addition, we model the photometric transformations in a spatially varying manner. Our experiments demonstrate the performance of the method with real images involving deforming motion and large changes in viewpoint, scale and illumination.

1 Introduction

The problem of motion segmentation typically arises in a situation where one has a sequence of images containing differently moving objects and the task is to extract the objects from the images using the motion information. In this context the motion segmentation problem consists of the following two subproblems: (1) determination of groups of pixels in two or more images that move together, and (2) estimation of the motion fields associated with each group [1].

Motion segmentation has a wide variety of applications. For example, representing the moving images with a set of overlapping motion layers may be useful for video coding and compression as well as for video mosaicking [2, 1]. Furthermore, the object-level segmentation and registration could be directly used in recognition and reconstruction tasks [3, 1].

Many early approaches to motion segmentation assume small motion between consecutive images and use dense optical flow techniques for motion estimation [2, 4]. The main limitation of optical flow based methods is that they are not suitable for large motions. Some approaches try to alleviate this problem by using feature point correspondences for initializing the motion models [5, 6, 1]. However, the implementations described in [5] and [6] still require that the motion is relatively small and approximately planar. The approach in [1] can deal with large planar motions.

Fig. 1. An example image pair, courtesy of [3], and the extracted motion components (middle) with the associated geometric and photometric transformations (right).

In this work, we address the motion segmentation problem in the context of wide baseline image pairs. This means that we consider cases where the motion of the objects between the two images may be very large due to non-rigid deformations and viewpoint variations. Another challenge in the wide baseline case is that the appearance of objects usually changes with illumination. For example, spatially varying illumination changes, such as shadows, occur frequently in wide baseline imagery and may further complicate object detection and segmentation. In order to address these challenges we propose a bottom-up motion segmentation approach which gradually expands and merges the initial matching regions into smooth motion layers and finally provides a dense assignment of pixels into these layers. Besides segmentation, the proposed method provides the geometric and photometric transformations for each layer.

The previous works closest to ours are [1, 7, 8]. In [1] the problem statement is the same as here, i.e., two-view motion segmentation for large motions. However, the solution proposed there requires approximately planar motion and does not model varying lighting conditions. The problem setting in [7] and [8] is slightly different than here since there the main focus is on object recognition. Nevertheless, the ideas of [7] and [8] can be utilized in motion segmentation and we develop them further towards a dense and deformable two-view motion segmentation method. In particular, we use the quasi-dense matching technique of [8] for initializing the motion layers. This allows us to avoid the planar motion assumption and makes the initialization more robust to extensive background clutter. In order to get the pixel level segmentation, we use graph cut based optimization together with a somewhat similar probabilistic model as in [7]. However, unlike in [7], we do not use any presegmented reference images but detect and segment the common regions automatically from both images. Furthermore, we propose a spatially varying photometric transformation model which is more expressive than the global model in [7].

In addition to the aforementioned publications, there are also other recent works related to the topic. For example, [9] describes an approach for computing layered motion segmentations of video. However, that work uses continuous video sequences and hence avoids the problems of large geometric and photometric transformations which make the wide baseline case difficult. Another related work is [10] which describes a layered image formation model for motion segmentation. Nevertheless, [10] does not address the problem of model initialization which is essential for large motions.

Algorithm 1: Outline of the method
Input:
two images I and I' and a set of seed matches
Algorithm:
1. Grow and group the seed matches [8]
2. Verify the grown groups of matches
3. Initialize motion layers
4. Perform dense segmentation of both images
5. Enforce the consistency of segmentations
Output:
a dense assignment of pixels to layers which define the motion for each pixel

Algorithm 2: Dense motion segmentation
Input:
• the image to be segmented (I) and the other image (I')
• a set of motion layers ($\mathcal{L}_j$) with geometric and photometric transformations ($\mathcal{G}_j$ and $\mathcal{F}_j$)
• initial segmentation S
Algorithm:
1. Update the photometric transformations $\mathcal{F}_j$
2. Update the geometric transformations $\mathcal{G}_j$
3. Update the segmentation S
4. Repeat steps 1-3 until S does not change

2 Overview

This section gives a brief overview of our approach whose main stages are summarized in Algorithm 1. The particular focus of this paper is on the dense segmentation method which is described in Algorithm 2 and detailed in Section 3.

2.1 Hypothesis generation and verification

First, given a pair of images and a sparse set of seed matches between them, we compute our motion hypotheses by region growing and grouping. That is, we first use the match propagation technique [8] to obtain more matching pixels in the spatial neighborhoods of the seed matches, which are acquired using standard region detectors and SIFT-based matching [11]. After the propagation, the coherently moving matches are grouped together by using a similar approach as in [8], where the neighboring quasi-dense matches, connected by Delaunay triangulation, are merged to the same group if the triangulation is consistent with the local affine motions estimated during the propagation. However, instead of the heuristic criterion in [8], we use the bending energy of locally fitted thin plate splines [12] to measure the consistency of triangulations.

Then, the grouped correspondences are verified in order to reject incorrect matches. The idea is to improve the precision of keypoint based matching by examining the grown regions, as in [3, 8, 13, 14]. In our current implementation the verification is based on the size of the matching regions [8] but also other decision criteria could be used in the proposed framework (cf. [14]). Finally, the verified groups of correspondences are used to initialize the tentative motion layers illustrated in Fig. 2.

2.2 Motion segmentation

The tentative motion layers are refined in the dense segmentation stage (Step 4, Alg. 1) where the assignment of pixels to layers is first done separately for each image whereafter the final layers are obtained by checking the inverse consistency of the two assignments as in [1] (Step 5, Alg. 1). The segmentation procedure (Alg. 2) iterates the following steps: (1) estimation of photometric transformations for each color channel, (2) estimation of geometric transformations, and (3) graph cut based segmentation of pixels to layers. The details of the iteration are described in Sect. 3 but the core idea is the following: when the segmentation is updated some pixels change their layer to a

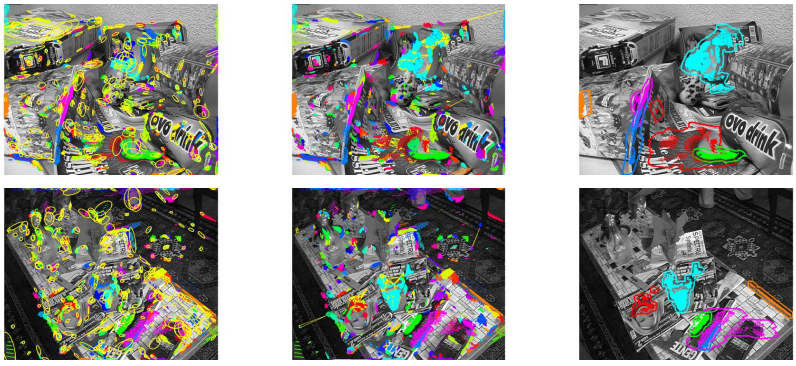

Fig. 2. Left: the seed regions (yellow ellipses) and the propagated quasi-dense matches. Middle: the grouped matches (each group has own color, the yellow lines are the Delaunay edges joining initial groups [8]). Right: the six largest groups and their support regions.

better one and this allows to improve the estimates for the geometric and photometric transformations of the layers (which then again improves the segmentation and so on).

The final motion layers for the example image pair of Fig. 2 are illustrated in the last column of Fig. 1 where the meshes illustrate the geometric transformations and the colors visualize the photometric transformations. The colors show how the gray color, shown on the background layer, would be transformed from the other image to the colored image. The result indicates that the white balance is different in the two images. Note also the shadow on the corner of the foremost magazine in the first image.

3 Dense and deformable motion segmentation

3.1 Layered model

Our layer-based model describes each one of the two images as a composition of layers which are related to the other image by different geometric and photometric transformations. In the following, we assume that image I is the image to be segmented and I' is the reference image. The other case is obtained by changing the roles of I and I'.

The model consists of a set of motion layers, denoted by $\mathcal{L}_j$, $j = 0, \ldots, L$. The segmentation of image I is defined by the label matrix S which has the same size as I (i.e. $m \times n$). So, $S(\mathbf{p}) = j$ means that the pixel $\mathbf{p}$ in I is labeled to layer j. The layer $j = 0$ is the background layer reserved for those pixels which are not visible in I'. The label matrix S is sufficient for representing the final assignment of pixels to layers. However, it is not sufficient for the initialization of our iterative segmentation method since some of the tentative layers may overlap as shown in Fig. 2. Therefore, for later use, we introduce additional label matrices S_j so that $S_j(\mathbf{p}) = 1$ if $\mathbf{p}$ belongs to layer j and $S_j(\mathbf{p}) = 0$ otherwise.

The geometric transformation associated to layer j ($j \neq 0$) is denoted by $\mathcal{G}_j$. In detail, the motion field $\mathcal{G}_j$ transforms the pixels in I to the other image and is represented by two matrices of size $m \times n$ (one for each coordinate). Thus, $\mathcal{G}_j(\mathbf{p}) = \mathbf{p}'$ means that pixel $\mathbf{p}$ is mapped to position $\mathbf{p}'$ in the other image if it belongs to layer j.

The photometric transformation of layer j ($j \neq 0$) is denoted by $\mathcal{F}_j$ and its parameters define an affine intensity transformation for each color channel at every pixel.

Hence, if the number of color channels is K, then $\mathcal{F}_j$ is represented by a set of $2K$ matrices each of which has size $m \times n$. So, the modeled intensity for color channel k at pixel $\mathbf{p}$ is defined by

$$\hat{I}_j^k(\mathbf{p}) = \mathcal{F}_j^k(\mathbf{p}) \cdot I'^k(\mathcal{G}_j(\mathbf{p})) + \mathcal{F}_j^{(K+k)}(\mathbf{p}), \quad (1)$$

where the superscript of $\mathcal{F}_j$ indicates which ones of the $2K$ transformation parameters correspond to channel k.

Given the latent variables S, $\mathcal{G}_j$, $\mathcal{F}_j$ and the reference image I', the relation (1) provides a generative model for I. In fact, the goal in the dense segmentation stage is to determine the latent variables so that the resulting layered model would explain well the observed intensities in I. This is acquired by minimizing an energy function which is introduced in Sect. 3.3. However, first, we describe how the layered model is initialized.

3.2 Model initialization

The motion hypotheses which pass the verification stage are represented as groups of two-view point correspondences and each of them is used to initialize a motion layer.

First, the initialization of the label matrices S_j is obtained directly from the support regions of the grouped correspondences. That is, we give a label $j > 0$ for each group and assign $S_j(\mathbf{p}) = 1$ for those pixels $\mathbf{p}$ that are inside the support region of group j. At this stage there may be pixels which are assigned to several layers. However, these conflicting assignments are eventually solved when the final segmentation S is produced (see Sect. 3.4).

Second, the initialization of the motion fields $\mathcal{G}_j$ is done by fitting a regularized thin-plate spline to the point correspondences of each group [12]. The thin-plate spline is a parametrized mapping which allows extrapolation, i.e., it defines the motion also for those pixels that are outside the particular layer. Thus, each motion field $\mathcal{G}_j$ is initialized by evaluating the thin-plate spline for all pixels $\mathbf{p}$.

Third, the coefficients of the photometric transformations $\mathcal{F}_j$ are initialized with constant values determined from the intensity histograms of the corresponding regions in I and I'. In fact, when $\mathcal{F}_j^k(\mathbf{p})$ and $\mathcal{F}_j^{K+k}(\mathbf{p})$ are the same for all $\mathbf{p}$, (1) gives simple relations for the standard deviations and means of the two histograms for each color channel k. Hence, one may estimate $\mathcal{F}_j^k$ and $\mathcal{F}_j^{K+k}$ by computing robust estimates for the standard deviations and means of the histograms. The estimates are later refined in a spatially varying manner as described in Sect. 3.5.

3.3 Energy function

The aim is to determine the latent variables $\boldsymbol{\theta} = \{S, \mathcal{G}_j, \mathcal{F}_j\}$ so that the resulting layered model explains the observed data $\mathcal{D} = \{I, I'\}$ well. This is done by maximizing the posterior probability $P(\boldsymbol{\theta}|\mathcal{D})$, which is modeled in the form $P(\boldsymbol{\theta}|\mathcal{D}) = \psi \exp(-E(\boldsymbol{\theta}, \mathcal{D}))$, where the normalizing factor ψ is independent of $\boldsymbol{\theta}$ [9]. Maximizing $P(\boldsymbol{\theta}|\mathcal{D})$ is equivalent to minimizing the energy

$$E(\boldsymbol{\theta}, \mathcal{D}) = \sum_{\mathbf{p} \in \mathcal{P}} U_{\mathbf{p}}(\boldsymbol{\theta}, \mathcal{D}) + \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{N}} V_{\mathbf{p},\mathbf{q}}(\boldsymbol{\theta}, \mathcal{D}), \quad (2)$$

where $U_{\mathbf{p}}$ is the unary energy for pixel $\mathbf{p}$ and $V_{\mathbf{p},\mathbf{q}}$ is the pairwise energy for pixels $\mathbf{p}$ and $\mathbf{q}$, $\mathcal{P}$ is the set of pixels in image I and $\mathcal{N}$ is the set of adjacent pairs of pixels in I.

The unary energy in (2) consists of two terms,

$$\sum_{\mathbf{p}\in\mathcal{P}} U_{\mathbf{p}}(\boldsymbol{\theta},\mathcal{D}) = \sum_{\mathbf{p}\in\mathcal{P}} -\log P_{\mathbf{p}}(I|\boldsymbol{\theta},I') - \log P_{\mathbf{p}}(\boldsymbol{\theta}) = \sum_{j=0}^{L} \sum_{\mathbf{p}|S(\mathbf{p})=j} -\log P_{\mathrm{l}}(I(\mathbf{p})|\mathcal{L}_j,I') - \log P(S(\mathbf{p})=j), \quad (3)$$

where the first one is the likelihood term defined by P_l and the second one is the pixelwise prior for $\boldsymbol{\theta}$. The pairwise energy in (2) is defined by

$$V_{\mathbf{p},\mathbf{q}}(\boldsymbol{\theta},\mathcal{D}) = \gamma(1-\delta_{S(\mathbf{p}),S(\mathbf{q})}) \exp\left(\frac{-\max_k |\nabla I^k(\mathbf{p}) \cdot \frac{\mathbf{p}-\mathbf{q}}{||\mathbf{p}-\mathbf{q}||}|^2}{\beta}\right), \quad (4)$$

where $\delta_{\cdot,\cdot}$ is the Kronecker delta function and γ and β are positive scalars. In the following, we describe the details behind the expressions in (3) and (4).

Likelihood term The term $P_{\mathbf{p}}(I|\boldsymbol{\theta},I')$ measures the likelihood that the pixel $\mathbf{p}$ in I is generated by the layered model $\boldsymbol{\theta}$. This likelihood depends on the parameters of the particular layer $\mathcal{L}_j$ to which $\mathbf{p}$ is assigned and it is modeled by

$$P_{\mathrm{l}}(I(\mathbf{p})|\mathcal{L}_j,I') = \begin{cases} \kappa & j=0 \\ P_{\mathrm{c}}(I(\mathbf{p})|\hat{I}_j) P_{\mathrm{t}}(I(\mathbf{p})|\hat{I}_j) & j\neq 0 \end{cases} \quad (5)$$

Thus, the likelihood of the background layer ($j=0$) is κ for all pixels. On the other hand, the likelihood of the other layers is modeled by a product of two terms, P_{c} and P_{t}, which measure the consistency of color and texture between the images I and $\hat{I}_j$, where $\hat{I}_j$ is defined by $\mathcal{G}_j$, $\mathcal{F}_j$, and I' according to (1). In other words, $\hat{I}_j$ is the image generated from I' by $\mathcal{L}_j$ and $P_{\mathrm{l}}(I(\mathbf{p})|\mathcal{L}_j,I')$ measures the consistency of appearance of I and $\hat{I}_j$ at $\mathbf{p}$.

The color likelihood $P_{\mathrm{c}}(I(\mathbf{p})|\hat{I}_j)$ is a Gaussian density function whose mean is defined by $\hat{I}_j(\mathbf{p})$ and whose covariance is a diagonal matrix with predetermined variance parameters. For example, if the RGB color space is used then the density is three-dimensional and the likelihood is large when $I(\mathbf{p})$ is close to $\hat{I}_j(\mathbf{p})$.

Here the texture likelihood $P_{\mathrm{t}}(I(\mathbf{p})|\hat{I}_j)$ is also modeled with a Gaussian density. That is, we compute the normalized grayscale cross-correlation between two small image patches extracted from I and $\hat{I}_j$ around $\mathbf{p}$ and denote it by $t_j(\mathbf{p})$. Thereafter the likelihood is obtained by setting $P_{\mathrm{t}}(I(\mathbf{p})|\hat{I}_j) = N(t_j(\mathbf{p})|1,\nu)$, where $N(\cdot|1,\nu)$ is a one-dimensional Gaussian density with mean 1 and variance ν.

Prior term The term $P_{\mathbf{p}}(\boldsymbol{\theta})$ in (3) denotes the pixelwise prior for $\boldsymbol{\theta}$ and it is defined by the probability $P(S(\mathbf{p})=j)$ with which $\mathbf{p}$ is labeled with j. If there is no prior information available one may here use the uniform distribution which gives equal probability for all labels. However, in our iterative approach, we always have an initial estimate $\boldsymbol{\theta}_0$ for the parameters $\boldsymbol{\theta}$ while minimizing (2), and hence, we may use the initial estimate S_0 to define a prior for the label matrix S. In fact, we model the spatial

distribution of labels with a mixture of two-dimensional Gaussian densities, where each label j is represented by one mixture component, whose portion of the total density is proportional to the number of pixels with the label j. The mean and covariance of each component are estimated from the correspondingly labeled pixels in S_0.

The spatially varying prior term is particularly useful in such cases where the colors of some uniform background regions accidentally match for some layer. (This is actually quite common when both images contain a lot of background clutter.) If these regions are distant from the objects associated to that particular layer, as they usually are, the non-uniform prior may help to prevent incorrect layer assignments.

Pairwise term The purpose of the term $V_{\mathbf{p},\mathbf{q}}(\boldsymbol{\theta}, \mathcal{D})$ in (2) is to encourage piecewise constant labelings where the layer boundaries lie on the intensity edges. The expression (4) has the form of a generalized Potts model [15], which is commonly used in segmentation approaches based on Markov Random Fields [1, 7, 9]. The pairwise term (4) is zero for such neighboring pairs of pixels which have the same label and greater than zero otherwise. The cost is highest for differently labeled pixels in uniform image regions where ∇I^k is zero for all color channels k. Hence, the layer boundaries are encouraged to lie on the edges, where the directed gradient is non-zero. The parameter γ determines the weighting between the unary term and the pairwise term in (2).

3.4 Algorithm

The minimization of (2) is performed by iteratively updating each of the variables S, $\mathcal{G}_j$ and $\mathcal{F}_j$ in turn so that the smoothness of the geometric and photometric transformation fields, $\mathcal{G}_j$ and $\mathcal{F}_j$, is preserved during the updates. The approach is summarized in Alg. 2 and the update steps are detailed in the following sections.

In general, the approach of Alg. 2 can be used for any number of layers. However, after the initialization (Sect. 3.2), we do not directly proceed to the multi-layer case but first verify the initial layers individually against the background layer. In detail, for each initial layer j, we run one iteration of Alg. 2 by using uniform prior for the two labels in S_j and a relatively high value of γ. Here the idea is that those layers j, which do not generate high likelihoods $P_1(I(\mathbf{p})|\mathcal{L}_j, I')$ for a sufficiently large cluster of pixels, are completely replaced by the background. For example, the four incorrect initial layers in Fig. 2 were discarded at this stage. Then, after the verification, the multi-label matrix S is initialized (by assigning the label with the highest likelihood $P_1(I(\mathbf{p})|\mathcal{L}_j, I')$ for ambiguous pixels) and the layers are finally refined by running Alg. 2 in the multi-label case, where the spatially varying prior is used for the labels.

3.5 Updating the photometric transformations

The spatially varying photometric transformation model is an important element of our approach. Given the segmentation S and the geometric transformation $\mathcal{G}_j$, the coefficients of the photometric transformation $\mathcal{F}_j$ are estimated from linear equations by using Tikhonov regularization [16] to ensure the smoothness of solution.

In detail, according to (1), each pixel $\mathbf{p}$ assigned to layer j provides a linear constraint for the unknowns $\mathcal{F}_j^k(\mathbf{p})$ and $\mathcal{F}_j^{(K+k)}(\mathbf{p})$. By stacking the elements of $\mathcal{F}_j^k$ and $\mathcal{F}_j^{(K+k)}$ into a vector, denoted by $\mathbf{f}_j^k$, we may represent all these constraints, generated

by the pixels in layer j, in matrix form $\mathbf{M}\mathbf{f}_j^k = \mathbf{b}$, where the number of unknowns in $\mathbf{f}_j^k$ is larger than the number of equations. Then, we use Tikhonov regularization and solve

$$\min_{\mathbf{f}_j^k} ||\mathbf{M}\mathbf{f}_j^k - \mathbf{b}||^2 + \lambda ||\mathbf{L}\mathbf{f}_j^k||^2, \tag{6}$$

where λ is the regularization parameter and the difference operator $\mathbf{L}$ is here defined so that $||\mathbf{L}\mathbf{f}_j^k||^2$ is a discrete approximation to

$$\int ||\nabla \mathcal{F}_j^k(\mathbf{p})||^2 + ||\nabla \mathcal{F}_j^{(K+k)}(\mathbf{p})||^2 d\mathbf{p}. \tag{7}$$

Since the number of unknowns is large in (6) (i.e. two times the number of pixels in I) we use conjugate gradient iterations to solve the related normal equations [16]. The initial guess for the iterative solver is obtained from the current estimate of $\mathcal{F}_j$. Since we initially start from a constant photometric transformation field (Sect. 3.2) and our update step aims at minimizing (6), thereby increasing the likelihood $P_1(\mathbf{p}|\hat{I}_j)$ in (3), it is clear that the energy (2) is decreased in the update process.

3.6 Updating the geometric transformations

The geometric transformations $\mathcal{G}_j$ are updated by optical flow [17]. Given S and $\mathcal{F}_j$ and the current estimate of $\mathcal{G}_j$, we generate the modeled image $\hat{I}_j$ by (1) and determine the optical flow from I to $\hat{I}_j$ in a domain which encloses the regions currently labeled to layer j [17] (color images are transformed to grayscale before computation). Then, the determined optical flow is used for updating $\mathcal{G}_j$. However, the update is finally accepted only if it decreases the energy (2).

3.7 Updating the segmentation

The segmentation is performed by minimizing the energy function (2) over different labelings S using graph cut techniques [15]. The exact global minimum is found only in the two-label case and in the multi-label case efficient approximate minimization is produced by the α-expansion algorithm of [15]. Here the computations were performed using the implementations provided by the authors of [15, 18–20].

4 Experiments

Experimental results are illustrated in Figs. 3 and 4. The example in Fig. 3 shows the first and last frame from a classical benchmark sequence [2, 4], which contains three different planar motion layers. Good motion segmentation results have been obtained for this sequence by using all the frames [2, 6, 9]. However, if the intermediate frames are not available the problem is harder and it has been studied in [1]. Our results in Fig. 3 are comparable to [1]. Nevertheless, compared to [1], our approach has better applicability in cases where (a) only a very small fraction of keypoint matches is correct, and (b) the motion can not be described with a low-parametric model. Such cases are illustrated in Figs. 1 and 4.

The five examples in Fig. 4 show motion segmentation results for scenes containing non-planar objects, non-uniform illumination variations, multiple objects, and deforming surfaces. For example, the recovered geometric registrations illustrate the 3D shape

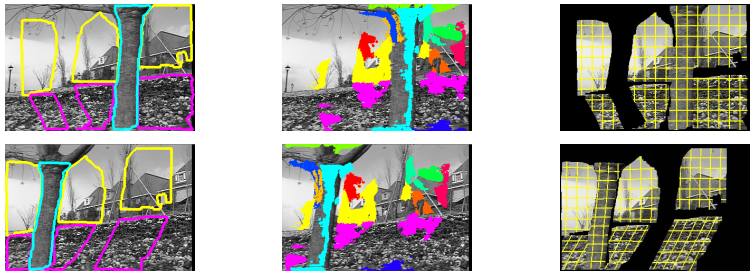

Fig. 3. Left: two images and the final three-layer segmentation. Middle: the grouped matches generating 12 tentative layers. Right: the layers of the first image mapped to the second.

of the toy lion and the car as well as the bending of the magazines. In addition, the varying illumination of the toy lion is correctly recovered (the shadow on the backside of the lion is not as strong as elsewhere). On the other hand, if the changes of illumination are too abrupt or if some primary colors are not present in the initial layer (implying that the estimated transformation may not be accurate for all colors), it is difficult to achieve perfect segmentation. For example, in the last column of Fig. 4, the letter "F" on the car, where the intensity is partly saturated, is not included in the car layer.

Besides illustrating the capabilities and limitations of the proposed method, the results in Fig. 4 also suggest some topics for future improvements. Firstly, improving the initial verification stage might give a better discrimination between the correct and incorrect correspondences (the magenta region in the last example is incorrect). Secondly, some postprocessing method could be used to join distant coherently moving segments if desired (the green and cyan region in the fourth example belong to the same rigid object). Thirdly, if the change in scale is very large, more careful modeling of the sampling rate effects might improve the accuracy of registration and segmentation (magazines).

5 Conclusion

This paper describes a dense layer-based two-view motion segmentation method, which automatically detects and segments the common regions from the two images and provides the related geometric and photometric registrations. The method is robust to extensive background clutter and is able to recover the correct segmentation and registration of the imaged surfaces in challenging viewing conditions (including uniform image regions where mere match propagation can not provide accurate segmentation). Importantly, in the proposed approach both the initialization stage and the dense segmentation stage can deal with deforming surfaces and spatially varying lighting conditions, unlike in the previous approaches. Hence, in the future, it might be interesting to study whether the techniques can be extended to multi-frame image sequences.

References

1. Wills, J., Agarwal, S., Belongie, S.: A feature-based approach for dense segmentation and estimation of large disparity motion. IJCV **68** (2006) 125–143
2. Wang, J.Y.A., Adelson, E.H.: Representing moving images with layers. IEEE Transactions on Image Processing **3**(5) (1994) 625–638

Fig. 4. Five examples. The bottom row illustrates the geometric and photometric registrations.

3. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation from single or multiple model views. IJCV **67** (2006) 159–188
4. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: CVPR. (1997)
5. Torr, P.H.S., Szeliski, R., Anandan, P.: An integrated bayesian approach to layer extraction from image sequences. TPAMI **23**(3) (2001) 297–303
6. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. TPAMI **27** (2005) 1644–1659
7. Simon, I., Seitz, S.M.: A probabilistic model for object recognition, segmentation, and non-rigid correspondence. In: CVPR. (2007)
8. Kannala, J., Rahtu, E., Brandt, S.S., Heikkilä, J.: Object recognition and segmentation by non-rigid quasi-dense matching. In: CVPR. (2008)
9. Kumar, M.P., Torr, P.H.S., Zisserman, A.: Learning layered motion segmentations of video. IJCV **76** (2008) 301–319
10. Jackson, J.D., Yezzi, A.J., Soatto, S.: Dynamic shape and appearance modeling via moving and deforming layers. IJCV **79** (2008) 71–84
11. Lowe, D.: Distinctive image features from scale invariant keypoints. IJCV **60** (2004) 91–110
12. Donato, G., Belongie, S.: Approximate thin plate spline mappings. In: ECCV. (2002)
13. Vedaldi, A., Soatto, S.: Local features, all grown up. In: CVPR. (2006)
14. Čech, J., Matas, J., Perd'och, M.: Efficient sequential correspondence selection by cosegmentation. In: CVPR. (2008)
15. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI **23**(11) (2001) 1222–1239
16. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems. SIAM (1998)
17. Horn, B.K.P., Schunk, B.G.: Determining optical flow. Artificial Intelligence (1981)
18. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. TPAMI **26**(9) (2004) 1124–1137
19. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? TPAMI **26**(2) (2004) 147–159
20. Bagon, S.: Matlab wrapper for graph cut. http://www.wisdom.weizmann.ac.il/∼bagon (2006)