Layered Motion Segmentation with a Competitive Recurrent Network

Julian Eggert, Joerg Deigmoeller, and Volker Willert

Honda Research Institute (HRI) Europe GmbH,
Carl-Legien-Str. 30, 63073 Offenbach/Main, Germany
{julian.eggert,joerg.deigmoeller}@honda-ri.de
Technical University of Darmstadt, Control Theory and Robotics Lab,
Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany
volker.willert@rtr.tu-darmstadt.de

Abstract. Using local motion information data such as that obtained from optical flow, we present a network for a multilayered segmentation into motion regions that are governed by affine motion patterns. Using an energy-based competitive multilayer architecture based on non-negative activations and multiplicative update rules, we show how the network can perform segmentation tasks that require a combination of affine estimation with local integration and competition constraints.

Key words: Motion segmentation, layering, affine, competitive multilayer networks, multiplicative gradient descent

1 Introduction

Motion-based segmentation and motion-based layer separation are essential steps for the decomposition of dynamic visual scenes. The topic has been investigated in a series of early publications by commonly either clustering locally estimated motion models [3, 5] or estimating the spatial support of mixture models [1, 4]. The latter frequently appears in the realm of probabilistic approaches, formulating the problem in terms of maximum-likelihood of the observed data.

A common starting point of the approaches is a motion field estimation as a preprocessing step. Subsequently, they introduce parameterized models for describing subregions in the motion field, where almost all of these approaches assume affine motion models. In order to consider spatial constraints, Markov Random Fields (MRFs) are commonly used to support regions of similar motion [2, 6]. As such approaches optimize conditions in a pixels neighbourhood, they lack in considering model competition at certain pixel positions.

In this paper, we used a dynamic neural network approach to combine the spatial distribution of labels (*intra-label integration*) with further constraints on *inter-label competition*. Such a model is motivated by previous work on competitive layer models [7]. The network dynamics within this work are deterministic and follow a gradient-descent-like update rule, updating motion region *parameter estimation* and motion region *labeling* in alternating steps. The dynamics

can be implemented very efficiently as block operations on a labeling grid. Compared to MRFs which rely on stochastic techniques, this deterministic relaxation method is a computational much more efficient approach.

Sec. 2 introduces the network for multi-region motion segmentation in detail. As a proof-of-concept, we present results from two motion sequences in Sec. 3. These include the MPEG Flower Garden sequence as well as an example of a moving hand filmed by a moving observer. Finally, we conclude the paper in Sec. 4.

2 Dynamic Neural Network for Motion Segmentation

In this paper, we start with a sequence of 2D images. For each consecutive pair of images, an approximation of the 2D *motion field* in the images is obtained by calculation of a dense optical flow in the form of velocity vectors $\mathbf{v}(\mathbf{p})$ at positions $\mathbf{p}$. In our particular implementation, we use a spatiotemporally integrating, patch-based method for calculating optical flow [8].

In this section, we introduce the main components of our motion estimation network. In Sec. 2.1, we briefly explain the calculation of parameters for the affine description of motion regions by applying weighted regression on a motion field. In Sec. 2.2, we present a competitive recurrent network for dynamically updating activations that encode the tendency of each image position to belong to the different motion models. In Sec. 2.3, we combine both approaches within a single energy function that drives the entire system.

2.1 Motion Fields Described by Affine Models

We assume the motion field to be composed of large spatial regions that can be approximated by *affine homographies*. This is valid if the images recorded by a camera are rectified and the 3D scene contains planar surfaces where changes in depth between objects and camera are small compared to their distance.

In the following, we introduce image coordinates $\mathbf{p} = \{p_x, p_y\}^T$ and motion vectors $\mathbf{v} = \{v_x, v_y\}^T$, as well as homogeneous image coordinates $\hat{\mathbf{p}} = \{p_x, p_y, 1\}^T$ and homogeneous motion vectors $\hat{\mathbf{v}} = \{v_x, v_y, 1\}^T$. The goal is to find N_A affine matrices A_k, where each matrix describes the motion field for a certain region. In other words, an assumed affine homography is suitable for describing a region of the image when a large amount of the measured motion vectors $\mathbf{v}(\hat{\mathbf{p}})$ can be approximated by motion vectors $\mathbf{v}_{A_k}(\hat{\mathbf{p}})$:

$$\mathbf{v}(\hat{\mathbf{p}}) \approx \mathbf{v}_{A_k}(\hat{\mathbf{p}}) = A_k \hat{\mathbf{p}} = \begin{pmatrix} a_{k,11} & a_{k,12} & a_{k,13} \\ a_{k,21} & a_{k,22} & a_{k,23} \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} . \tag{1}$$

Assuming that the motion field has been measured at $N_{\mathbf{p}}$ positions $\mathbf{p}_i := \{p_{i,x}, p_{i,y}\}^T$, we have $\mathbf{v}_i := \mathbf{v}(\mathbf{p}_i)$. Since we want to describe the entire motion field by all N_A affine models simultaneously, we introduce weight factors $w_{i,k} \geq 0$ which indicate the *affiliation* of a motion vector $\mathbf{v}_i$ to an affine model A_k. For

fixed weight factors, and assuming the underlying parametric model of residuals $r_{i,k}$ to be Gaussian, we can formulate a cost function $G(\{A_k\})$ that is the weighted Euclidean distance between measured and expected motion:

$$\begin{aligned} G(\{A_k\}) &= \sum_k G_k(A_k) = \sum_k \sum_i w_{i,k} \|\mathbf{v}(\hat{\mathbf{p}}_i) - \mathbf{v}_{A_k}(\hat{\mathbf{p}}_i)\|^2 \\ &= \sum_k \sum_i w_{i,k} \underbrace{\|\mathbf{v}(\hat{\mathbf{p}}_i) - A_k \hat{\mathbf{p}}_i\|}_{:=r_{i,k}}^2. \end{aligned} \tag{2}$$

For this case, a weighted linear regression can be used for the calculation of the affine models. For each affine model A_k describing a region of the motion field characterized by the affiliation weights $w_{i,k}$, the affine parameters are then estimated by minimizing the cost function $G(\{A_k\})$:

$$A_k^* = \arg\min_{A_k} G(\{A_k\}) = \arg\min_{A_k} G_k(A_k). \tag{3}$$

The cost function gets minimal if:

$$\begin{aligned} \nabla_{A_k} \Bigg\{ & \sum_i w_{i,k} [v_{i,x} - (a_{k,11}\ p_{i,x} + a_{k,12}\ p_{i,y} + a_{k,13})]^2 \\ & + \sum_i w_{i,k} [v_{i,y} - (a_{k,21}\ p_{i,x} + a_{k,22}\ p_{i,y} + a_{k,23})]^2 \Bigg\} = 0. \end{aligned} \tag{4}$$

This in turn leads to a linear equation system in the coefficients of the A_k's, which can be solved analytically in a straightforward way (see Appendix).

Nevertheless, this only applies for *fixed* affiliation weights $w_{i,k}$. If these are given by some preprocessing step (like segmentation) we would be done. However, we want to simultaneously estimate A_k *and* $w_{i,k}$. The first step towards this is to consider that the affiliation weights act as a kind of affiliation probability, i.e. the different models are loosely coupled via their affiliations by a normalization condition $\sum_k w_{i,k} = 1$ and $w_{i,k} > 0$. Furthermore, beyond the normalization condition, we let the models A_k compete explicitly for their affiliations, i.e., a model A_k which best describes the motion field at location $\mathbf{p}_i$ receives a significant affiliation weight $w_{i,k}$. In the following section, the implementation of this competition by means of a recurrent neural network is presented.

2.2 Extraction of Motion Layers with a Recurrent Neural Network

We have seen that the considerations of the previous section assumed the affiliation weights to be fixed or determined by external means. To make affiliation weights compete for their models, we apply a competitive neural network which consists of a grid-like arrangement of neurons at image positions $\mathbf{p}_i$ with activities $a_{i,k}$ for all positions and all models A_k.

The network dynamics are determined by energy function $E(\{a_{i,k}\})$ of all neuronal activities $\{a_{i,k}\}$. It can be considered as a layered neural network with

a fixed number of N_A layers (indexed here by $k = 1, ..., N_A$), and each layer consisting of $N_{\mathbf{p}}$ positions (indexed here by $i = 1, ..., N_{\mathbf{p}}$, but which for practical purposes should be arranged in x and y coordinates so as to map with the input image).

Each neuron receives two sources of input. One, $h_{i,k}$, is a driving input, which originates from outside of the network and conveys a kind of "sensory" support for the neuron $a_{i,k}$. The second is an input originating from recurrent connections from within the network itself, i.e., from the other neurons. This input serves to trigger a competition between the neurons on different layers, and for imposing a spatial coupling between different positions. The energy function therefore comprises three terms, one for the driving input (d), one for the layer competition (l) and one for the spatial coupling term (c):

$$\begin{aligned} E(\{a_{i,k}\}) &= E_d(\{a_{i,k}\}) + E_l(\{a_{i,k}\}) + E_c(\{a_{i,k}\}) \qquad (5) \\ &= -\lambda_1 \sum_i \sum_k h_{i,k} a_{i,k} + \lambda_2 \frac{1}{2} \sum_i \sum_k \sum_{k'} W_k^{k'} a_{i,k'} a_{i,k} - \lambda_3 \frac{1}{2} \sum_i \sum_{i'} \sum_k K_i^{i'} a_{i,k} a_{i',k}. \end{aligned}$$

Subsequently, we assume a positivity constraint $a_{i,k} > 0$ and $h_{i,k} > 0$. The energy function should be minimized which is the case e.g., for high activities at those neurons with a large (positive) driving input $h_{i,k}$, and which are consistent with the layer competition and the spatial coupling conditions.

In Eq. (5), we have restricted ourselves to a layer competition term which acts over all layers (corresponding to the motion models A_k) but exclusively on neurons at the same position $\mathbf{p}_i$. This is parameterized by the kernel $W_k^{k'}$, which quantifies the competition between the activities $a_{i,k'}$ and $a_{i,k}$. Similarly, we use a kernel $K_i^{i'}$ to express the spatial coupling within one layer. This segregation into inter-layer and spatial coupling is not strictly necessary so that a fully connected network may also be used, but for the purpose presented here it is sufficient. Both kernels $W_k^{k'}$ and $K_i^{i'}$ are chosen to be symmetric and positive. The structure of the neural network, including layerwise and spatial coupling is illustrated in Fig. 1.

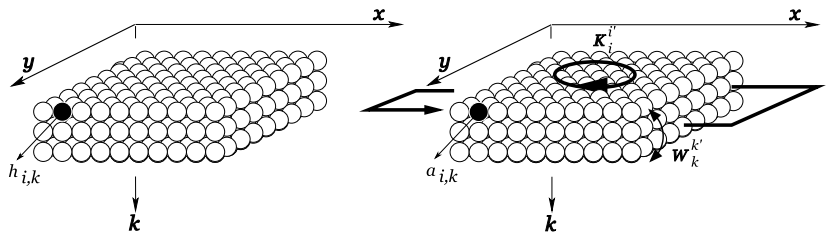

Fig. 1. Structure of the recurrent neural network. Each layer of neurons represents the affiliation to a specific motion model through normalized activities $a_{i,k}$ (right). The network is initialized by the driving input $h_{i,k}$ (left).

The dynamics of the activities are motivated by standard gradient descent considerations. We therefore obtain $a_{i,k} \sim -\nabla_{a_{i,k}} E$ with:

$$\nabla_{a_{i,k}} E = -\lambda_1 h_{i,k} + \lambda_2 \frac{1}{2} \sum_{k'} W_k^{k'} a_{i,k'} - \lambda_3 \frac{1}{2} \sum_{i'} K_i^{i'} a_{i',k} \ . \tag{6}$$

For accurate motion model separation, we would like the activities to be interpretable as affiliation probabilities to different motion models. This implies that the activities should always remain positive and they should always be normalized over the layer index, so that $\sum_k a_{i,k} = 1$. Neither condition is fulfilled by dynamics according to Eq. (6).

We impose the first condition, positivity, by using a multiplicative update rule motivated by exponentiated gradient descent and non-negative matrix factorization (NMF) techniques, similar to [9]. In our case, we separate positive and negative terms of the gradient from Eq. (6) according to:

$$\nabla_{a_{i,k}} E := \nabla_{a_{i,k}} E^+ - \nabla_{a_{i,k}} E^- \tag{7}$$

and express the dynamics by the fixpoint condition:

$$a_{i,k} \leftarrow a_{i,k} \ \frac{\nabla_{a_{i,k}} E^-}{\nabla_{a_{i,k}} E^+} \ . \tag{8}$$

Intuitively, as the dynamics approaches the minimum of the energy function, $\nabla_{a_{i,k}} E^+ \rightarrow \nabla_{a_{i,k}} E^-$ such that $\frac{\nabla_{a_{i,k}} E^-}{\nabla_{a_{i,k}} E^+} \rightarrow 1$ and the activities approach a static state. In addition to the advantage of positivity, the multiplicative update rule does not depend on a step size as gradient descent does.

The second condition, having normalized activities, is not trivial to impose. We cannot modify the activities according to Eq. (8) and then simply normalize the activities at each time step because normalization changes the overall energy $E(\{a_{i,k}\})$ in an unpredictable way, leading for example to a potential energy increase (instead of a decrease). Instead, we can either calculate the gradient and then modify the activities by projecting it onto the "normalized energy submanifold" or we can search for dynamics that have a continuous normalization condition built-in. The latter is the case for energy $E(\{\bar{a}_{i,k}\})$ based on position-wise (resp. columnwise) normalized activities

$$\bar{a}_{i,k} := \frac{a_{i,k}}{\sum_{k'} a_{i,k'}} \ . \tag{9}$$

Now, we are searching for the dynamics of the activities that minimizes $E(\{\bar{a}_{i,k}\})$ (instead of $E(\{a_{i,k}\})$). This can be done by the multiplicative update rule according to Eq. (8) except that activities are now normalized:

$$a_{i,k} \leftarrow a_{i,k} \ \frac{\nabla_{a_{i,k}} E^-(\{\bar{a}_{i,k}\})}{\nabla_{a_{i,k}} E^+(\{\bar{a}_{i,k}\})} \ . \tag{10}$$

Accordingly, the derivative of the energy function changes to $\nabla_{a_{i,k}} E(\{\bar{a}_{i,k}\}) = \sum_{k'} \nabla_{\bar{a}_{i,k'}} E(\{\bar{a}_{i,k}\}) \nabla_{a_{i,k}} \bar{a}_{i,k'}$.

In summary, if we apply the update rule (Eq. (10)) followed by activity normalization (Eq. (9)), we get an activity dynamics that minimizes the energy function under the constraints of positive and normalized activities.

In our implementation, we further used a layer competition kernel $W_k^{k'} := \delta(k, k')$. The detailed activity dynamics from Eq. (10) is then:

$$a_{i,k} \leftarrow a_{i,k} \frac{\lambda_1 h_{i,k} + \lambda_2 \sum_{k'} \bar{a}_{i,k'}^2 + \lambda_3 \sum_{i'} K_i^{i'} \bar{a}_{i',k}}{\lambda_1 \sum_{k'} h_{i,k'} \bar{a}_{i,k'} + \lambda_2 \bar{a}_{i,k} + \lambda_3 \sum_{k'} \sum_{i'} K_i^{i'} \bar{a}_{i',k'} \bar{a}_{i,k'}} \tag{11}$$

which at each update should be followed by normalization Eq. (9). Therefore, Eq. (11) is an iterative descent towards the minimization of the energy $E(\{\bar{a}_{i,k}\})$.

2.3 Combined Segregation and Affine Model Estimation

In Sec. 2.1 we explained how to estimate a number of affine models to describe partial motion fields of an image sequence for pre-set model affiliation weights $w_{i,k}$. In Sec. 2.2 we introduced a modified recurrent, layered network to let the affiliation weights compete for their models, triggered by the driving input $h_{i,k}$. The weights are encoded in different layers and incorporate through spatial coupling constraints. In this section, we fuse the energy functions of Sec. 2.1 and Sec. 2.2 by combining the affiliation weights and the driving input. This allows iterative calculations to estimate the motion models and the best affiliation probabilities for the models.

We assume the affiliation weights to be represented directly by the activities of the layered network, i.e., $w_{i,k} \equiv a_{i,k}$. Since by construction the activities $a_{i,k}$ remain positive and normalized if deployed according to the neuronal dynamics from Sec. 2.2, they fulfill the conditions postulated for the weights in Sec. 2.1.

Furthermore, we assume the driving input to originate from the consideration of how well a given model A_k serves to describe a motion flow $\mathbf{v}$ at the positions $\mathbf{p}_i$ indicated by, and weighted with, the model affiliation probabilities $a_{i,k}$. Using the considerations from Sec. 2.1, the driving input should be large when the measured flow and the model-based flow match, in our case by using:

$$h_{i,k} \sim e^{-\frac{1}{2\sigma^2} ||\mathbf{v}(\hat{\mathbf{p}}_i) - A_k \hat{\mathbf{p}}_i||^2} \ . \tag{12}$$

Finally, the complete energy equation becomes:

$$E(\{a_{i,k}\},\{A_k\}) = -\lambda_1 \sum_i \sum_k e^{-\frac{1}{2\sigma^2}||\mathbf{v}(\hat{\mathbf{p}}_i)-A_k\hat{\mathbf{p}}_i||^2} a_{i,k}$$
$$+\lambda_2 \frac{1}{2} \sum_i \sum_k \sum_{k'} W_k^{k'} a_{i,k'} a_{i,k} - \lambda_3 \sum_i \sum_{i'} \sum_k K_i^{i'} a_{i,k} a_{i',k} \ . \quad (13)$$

This energy function now has to be solved simultaneously for the models A_k and the activities $a_{i,k}$, with the additional constraints of positive and normalized activities. We then proceed as before with gradient descent, taking

$$\nabla_{a_{i,k}} E(\{\bar{a}_{i,k}\},\{A_k\}) \quad (14)$$

and

$$\nabla_{A_k} E(\{\bar{a}_{i,k}\},\{A_k\}) \quad (15)$$

to update the activities and the models in alternating steps for fixed models and activities, respectively. As a shortcut, during the model update according to Eq. (15) we assume that at positions where the layer activity is large, the corresponding model is already matching well (which is the case close to the minimum of the energy function). This means that $\mathbf{v}(\hat{\mathbf{p}}_i) \approx A_k \hat{\mathbf{p}}_i$ and hence we can approximate

$$h_{i,k} \approx 1 - \frac{1}{2\sigma^2} ||\mathbf{v}(\hat{\mathbf{p}}_i) - A_k \hat{\mathbf{p}}_i||^2 \quad (16)$$

so that

$$E(\{\bar{a}_{i,k}\},\{A_k\}) \approx -\lambda_1 N_{\mathbf{p}} \ + \ \lambda_1 \sum_i \sum_k \frac{1}{2\sigma^2} ||\mathbf{v}(\hat{\mathbf{p}}_i) - A_k \hat{\mathbf{p}}_i||^2 \ \bar{a}_{i,k}$$
$$+\lambda_2 \frac{1}{2} \sum_i \sum_k \sum_{k'} W_k^{k'} \bar{a}_{i,k'} \bar{a}_{i,k} - \lambda_3 \sum_i \sum_{i'} \sum_k K_i^{i'} \bar{a}_{i,k} \bar{a}_{i',k} \ . \quad (17)$$

Therefore, we get a contribution to the energy function from the driving input which is identical in form to Eq. (2), and hence can be solved using Eq. (4) and Eq. (18).

The full algorithm then reads:

1. Initialize the activities $\{a_{i,k}\}$
2. At each time step:
   (a) Get the motion vector field $\{\mathbf{v}_i\}$
   (b) Calculate the models $\{A_k\}$ according to Eq. (18)
   (c) Calculate the driving input $\{h_{i,k}\}$ according to Eq. (12)
   (d) Update the activities $\{a_{i,k}\}$ according to Eq. (11)
   (e) Normalize the activities $\{a_{i,k}\}$ to 1 according to Eq. (9)
   (f) While a desired energy decrease has not been reached, go to 2 (b)
3. Optional: Warp the current activities $\{a_{i,k}\}$ with the calculated affine parameters A_k as prediction for the next time step
4. Go to step 2

The warping step (3) allows to move the activities along with the motion field, which is very useful for temporally persistent, coherent motion. In this case, the affiliation probabilities represented by the activities $a_{i,k}$ are shifted with the stimulus, which requires less repetitions of steps 2(b)-2(f).

3 Results

Below, results of the proposed algorithm for two image sequences are presented. The first is the well known MPEG flower garden sequence, available at e.g. [11]. It consists of several planes shifting horizontally due to a moving observer: the tree in front moves fastest, the flower bed moves at intermediate speed and the house moves very slowly. To compute the driving input for the recurrent network, motion vector fields are estimated by the method described in [8]. The second video shows a moving hand in front of a moving background available at [10], including a manually annotated flow field.

For both sequences, the activities of the algorithm have been initialized by zero-mean Gaussian noise. For relaxation, 30 iterations were used for each motion vector field, where iterations 1, 15 and 30 were plotted in Figs. 2 and 3. The first row shows the input images and following rows each represent a motion layer. The λs to weight the energy terms have been set to $\lambda_1 = 0.6$, $\lambda_2 = 0.3$ and $\lambda_3 = 0.1$ for all sequences. These heuristically evaluated values represent a good parameterization for a variety of examples.

For the flower garden sequence, the number of models was set to three. It can be seen from the image sequence that after 15 iterations the system starts to converge (2nd column) and already after 30 iterations all models can be clearly separated (3rd column). As described in the previous chapter, activities are warped here to predict the activities for the next input image (see columns three to four). This avoids starting from the scratch for every new incoming image.

For the hand sequence, two models were assumed to be present in the flow field. Again, after 30 iterations the method is able to clearly separate the two layers. For this example, the fitted vector fields are plotted into the layers. This illustrates the close interaction between activities and affine model parameters.

4 Conclusion

We have presented a model for motion-based image segmentation into multiple affine motion layers. In contrast to many approaches, we employ a strictly positive dynamic neural network to address the problem of gaining the affiliation parameters for each layer. This allows us to directly combine conditions for layer competition and spatial coherence in a single energy function.

The energy optimization for normalized, positive activities provides a computationally efficient way to minimize the total energy. This allows an effective implementation to make the system employable to practical applications.

The proposed framework makes an important contribution to interpreting and understanding visual scenes containing rigid moving objects. The capability of the algorithms to successfully separate motion in real-world images has been shown for two video sequences.

As the evaluation has shown, the system provides accurate results. This indicates a solid basis for more complex scenarios. Of course, in more challenging

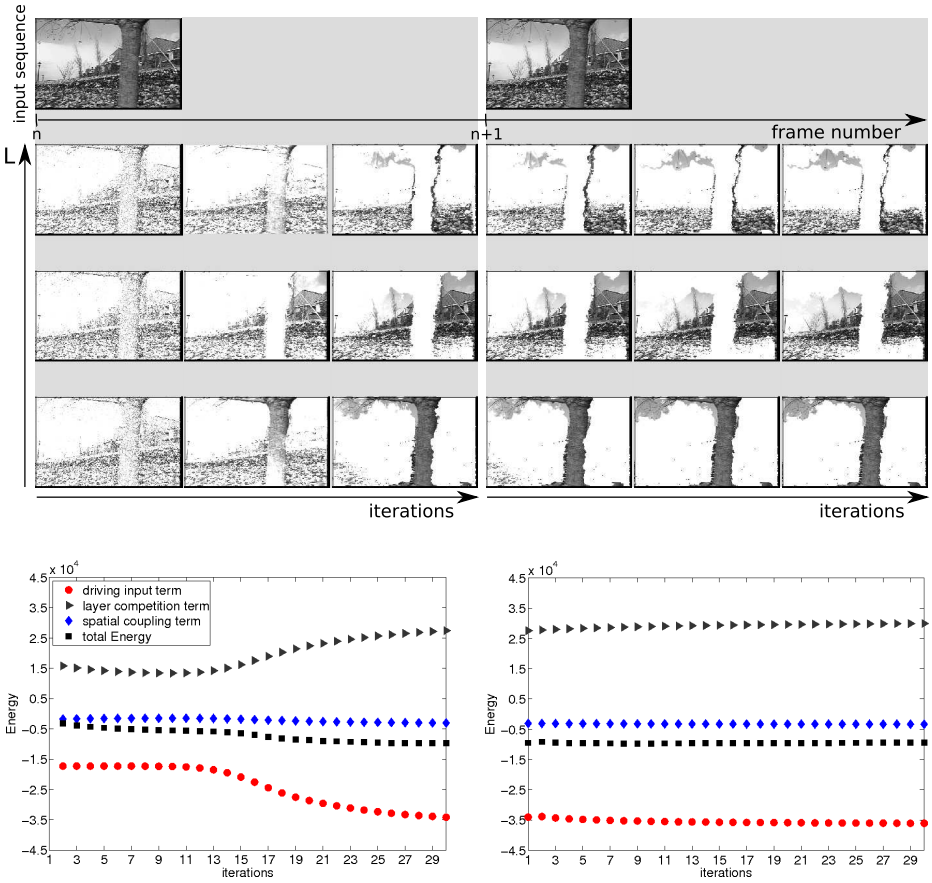

Fig. 2. Layer separation for the MPEG flower garden sequence. The first row shows the input images and the next three rows represent the layer activities. Each column of a layer illustrates the activities for the 1st, 15th and 30th iteration step. After the last iteration of the first input image, the activities are warped to predict activities for the next input image as initialization.

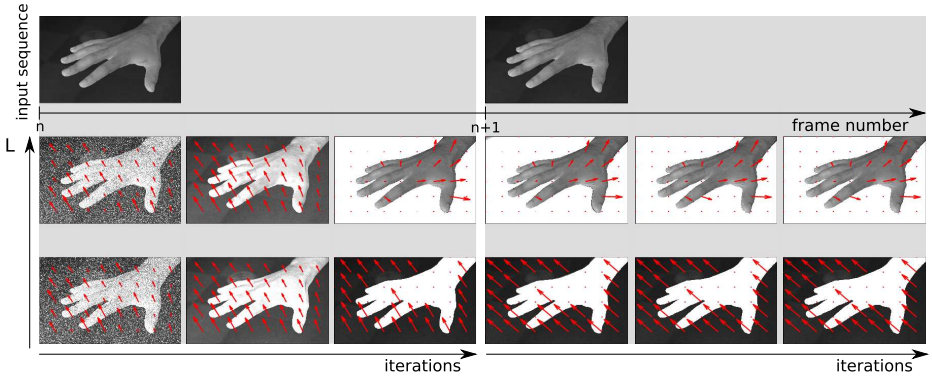

Fig. 3. The hand sequence, available at [10]. The first row shows the input images and the next two rows represent the motion layers. Each column of a layer illustrates the activities for the 1st, 15th and 30th iteration step. For each layer, the fitted vector field is plotted as well.

scenarios we have to cope with less reliable measured motion. In such cases, the regression analysis using the Euclidean norm might not suffice to satisfactorily fit the motion models. Now, as we have shown the consistency and correct functionality of the implementation, the next step is to prepare the algorithm to be able to deal with outliers and larger uncertainties in the motion data.

Appendix: Solution of the weighted regression

$$\arg\min_{A_k} G_k(A_k) = A_k^* = \begin{pmatrix} a_{k,13}^* & a_{k,23}^* \\ a_{k,11}^* & a_{k,21}^* \\ a_{k,12}^* & a_{k,22}^* \end{pmatrix} = \tag{18}$$

$$\begin{pmatrix} \sum_i w_{i,k} & \sum_i w_{i,k} p_{i,x} & \sum_i w_{i,k} p_{i,y} \\ \sum_i w_{i,k} p_{i,x} & \sum_i w_{i,k} p_{i,x}^2 & \sum_i w_{i,k} p_{i,x} p_{i,y} \\ \sum_i w_{i,k} p_{i,y} & \sum_i w_{i,k} p_{i,x} p_{i,y} & \sum_i w_{i,k} p_{i,y}^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_i w_{i,k} v_{i,x} & \sum_i w_{i,k} v_{i,y} \\ \sum_i w_{i,k} p_{i,x} v_{i,x} & \sum_i w_{i,k} p_{i,x} v_{i,y} \\ \sum_i w_{i,k} p_{i,y} v_{i,x} & \sum_i w_{i,k} p_{i,y} v_{i,y} \end{pmatrix}$$

References

1. Weiss, Y., Adelson, E. H.: Perceptually organized EM: A framework for motion segmentation that combines information about form and motion. M.I.T. Media Lab Perceptual Computing Section Technical Report No. 315, Cambridge (1995)
2. Weiss, Y., Adelson, E. H.: A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, Ca. (1996)
3. Wang, J. Y. A., Adelson, E. H.: Representing moving images with layers. IEEE Transactions on Image Processing, vol.3, no. 5, (1994)
4. Ayer, S., Sawhney, H. S.: Layered Representation of Motion Video using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding. Proceedings of the Fifth International Conference on Computer Vision, Washington, (1995)
5. Ju, S. X., Black, M. J., Jepson, A.D.: Skin and bones: multi-layer, locally affine, optical flow and regularization with transparency. IEEE Conference on Computer Vision and Pattern Recognition, San Francisco (1996)
6. Odobez, J.-M., Bouthemy, P.: Direct incremental model-based image motion segmentation for video analysis. Signal Processing, Volume 66, Issue 2 (1998)
7. Wersing, H., Steil, J. J., Ritter, H.: A Competitive Layer Model for Feature Binding and Sensory Segmentation. Neural Computation, (2001)
8. Willert, V., Eggert, J., Adamy, J., Korner, E.: Non-Gaussian velocity distributions integrated over space, time, and scales. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (2006)
9. Eggert, J., Koerner, E: Sparse coding and NMF. IEEE International Joint Conference on Neural Networks, Budapest, (2004)
10. Human-Assisted Motion Annotation, `http://people.csail.mit.edu/celiu/motionAnnotation/index.html`
11. Image Sequences, `http://www.cs.brown.edu/~black/images.html`