Submitted to *Sensors*. Pages 1 - 22.

OPEN ACCESS

Article

Segment tracking via a spatiotemporal linking process including feedback stabilization in an n-D lattice model

Babette Dellen[1,2], Eren Erdal Aksoy[3], and Florentin Wörgötter[3]

[1] Bernstein Center for Computational Neuroscience Göttingen, Max-Planck Institute for Dynamics and Self-Organization, Bunsenstrasse 10, Göttingen, Germany

[2] Institut de Robotica i Informatica Industrial (CSIC-UPC), Barcelona, Spain

[3] Bernstein Center for Computational Neuroscience Göttingen, University Göttingen, Bunsenstrasse 10, Göttingen, Germany

E-mail: E-mails: bkdellen@bccn-goettingen.de; eaksoye@bccn-goettingen.de; worgott@bccn-goettingen.de

* Author to whom correspondence should be addressed; bkdellen@bccn-goettingen.de, Phone: ++49 (0)551 5176 526, Fax: ++49 (0)551 5176 449

Version August 6, 2009 submitted to Sensors. *Typesetted by LaTeX using class file mdpi.cls*

Abstract: We extend the concept of spatially synchronous dynamics in spin-lattice models to the spatiotemporal domain to track segments within an image sequence. The method is related to synchronization processes in neural networks and based on superparamagnetic clustering of data. Spin interactions result in the formation of clusters of correlated spins, providing an automatic labeling of corresponding image regions. The algorithm obeys detailed balance. This is an important property as it allows for consistent spin-transfer across subsequent frames, which can be used for segment tracking. In the tracking process the correct equilibrium will, thus, always be found, which is an important advance as compared to other, more heuristic, tracking procedures. In the case of long image sequences, i.e. movies, the algorithm is augmented with a feedback mechanism, further stabilizing segment tracking. Model-free tracking is important for solving tasks such as moving-object tracking and action recognition in cases where no prior object knowledge is available.

Keywords: model-free segment tracking; image motion; image segmentation

1. Introduction

How can me make sense out of a complex visual scene having no or only little prior knowledge about its contents and the objects therein? Such problems occur for example if we wish to learn cause-effects in an hitherto unknown environment. Vice versa, many object definitions are only meaningful within the context of a given scenario and a set of possible actions.

Object tracking, i.e. the assignment of consistent labels to objects in different frames of a video, is important for solving various tasks in the field of computer vision, including automatic surveillance, human-computer interaction, and traffic monitoring [1]. Most object tracking algorithms require that predefined objects of interests are detected in the first frame or in every frame of the movie. In an unknown scenario however, the tracking of image segments, presumbably representing parts of objects, allows to postpone object definition to a later step of the visual-scene analysis. Several approaches for segment tracking have been proposed in the context of video segmentation [2,3,4,5,6,7,8,9]. Some approaches rely on segmenting each frame independently, e.g. by classifying pixels into regions based on similarity in the feature space, followed by a segment matching step based on their low-level features [2,3,4,5]. Other methods use motion projection to link segments, i.e. the position of a segment in a future frame is estimated from its current position and motion features [6,7,8,9].

In this paper, we group pixels based on a feature similarity criterium using a method based on the superparamagnetic clustering of data. Tracking of segments is accomplished through simultaneous segmentation of adjacent frames which are linked using local correspondence information, e.g. computed via standard algorithms for optic flow [10]. Blatt et al. (1998) first formulated the segmentation problem in terms of Potts models of granular ferromagnets or spins [11]. In the superparamagnetic phase, segments, i.e. ordered regions of aligned spins, appear naturally. To further accelerate the relaxation of the spin system, Opara and Wörgötter (1998) introduced an energy-based cluster updating technique (ECU), based on the cluster-updating method of Swendsen and Wang (1987) [12], and applied the algorithm to the problem of image segmentation [13,14]. The relaxation processes which can be emulated in spin-lattice models are often being used as approximations for synchronization processes within neural assemblies [13,14,15,16,17,18].

The motivation for our choice is threefold. First, in superparamagnetic clustering the number of segments is determined by the algorithm itself and thus does not need to be predefined. Second, the method obeys detailed balance, ensuring that the algorithm converges to a stable solutation independent of the initial conditions. Third, the concept of spin-relaxation can be easily extended to space-time by allowing bonds to be formed between spins belonging to different movie frames. Thus, time, i.e. frame number, just takes the role of additional dimensions in the spin-relaxation process, using only energy minimization without further constraints.

The segmentation (or partition) of an image is sensitive to global and local changes of the image, i.e. small changes in illumination, the appearance/disappearance of objects parts, causing the partition to change from one frame to the next. By synchronizing the segmentation process of adjacent frames, these kind of partitioning instabilities can be reduced. Furthermore, segment correspondences can be established without having to employ segment matching. To further stabilize segment tracking in the case of long image sequences, we developed a feedback control mechanism, which allows segmentation instabilities, e.g. sudden disappearences of segments, to be detected and removed by adjusting a control parameter of the segmentation algorithm.

The main contribution of this paper is the development of a model-free tracking algorithm, i.e. no prior object knowledge is required in order to track image segments from frame to frame. Potential applications include model-free moving object detection and tracking, and action recognition and classification from movies [19].

The paper is structured as follows: In Section II, we extend the method of superparamagnetic clustering in spin models to the temporal dimension and introduce the controller algorithm. In Section III, we first verificate the core algorithm using short image sequences because these are more suitable to introduce and test the method. We further investigate the sensitivity of the algorithm to system parameters and noise. Then, we demonstrate that segment tracking can be achieved for real movies. In Section IV, the results are discussed.

2. Algorithmic framework

Segment tracking can be roughly divided into the following subtasks: (i) image segmentation, (ii) linking (tracking), and (iii) stabilization. The third point acknowledges that segments, unlike objects, are not per se stable entities, but are sensitive to changes in the visual scene. Subtasks (i-ii) will be solved using a conjoint spin-relaxation process emulated in an n-dimensional (n-D) lattice, which defines the core algorithm (Section II.A). Local correspondence information for linking is obtained using standard algorithms for either stereo or optic flow [10,20]. The conjoint segmentation approach has the advantage that the spin-relaxation processes of adjacent images synchronize, reducing partioning instabilities.

Since simultaneous segmentation of long image sequences is practically impossible due to the high computational costs, we usually split the image sequence into a sequence of pairs. For example, the subsequent frames t_0, t_1 and t_2 are split into two pairs $\{t_0, t_1\}$ and $\{t_1, t_2\}$, where the last frame of previous pair is identical to the first frame of the next pair. If a segment of the last frame of $\{t_0, t_1\}$ and a segment of the first frame of $\{t_1, t_2\}$ occupy the same image region, we can assign the same segment label to both segments. This way segments can be tracked through the entire sequence. Since the algorithm preserves detailed balance (Section II.B), spins can be transferred from one frame to the next, greatly reducing the number of iterations required to achieve a stable segmentation.

We further stabilize segment tracking by introducing a feedback controller (Section II.C). In long image sequences, partioning instabilities are likely to arise at some point during the tracking process. Thus, segments may be lost due to merging or splitting of segments. The feedback controller detects these kind of instabilities and adjust a control parameter of the core algorithm to recover the original segments.

2.1. *Core algorithm*

The method of superparamagnetic clustering has been previously used to segment single images [13, 14,18]. Applying this framework to image sequences requires spin interactions to take place across frames. Due camera and object motion the images undergo changes during the course of time. To connect different frames, the mapping from one frame to the next needs to be known at least in some approximation. We solve this problem in the following way: Point correspondences, derived using algorithms for disparity or optic-flow computation, can be incorporated into the Potts model by allowing spins belonging to different frames of the image sequence to interact if the respective pixels belong to locally corresponding image points. Then, spins belonging to the different frames of the sequences are relaxed simultaneously, resulting in a synchronized segmentation of the images of the sequence. The inter-frame spin interactions cause the spins of corresponding image regions to align, and, thus, they will be assigned to the same segment. Since the formation of segments is a collective process, the point correspondences do not have to be very accurate nor does the algorithm require point correspondences for each pixel. It is usually sufficient if the available correspondences capture the characteristics of the scene only roughly.

The aim of this work is to find corresponding image regions in image sequences, i.e. stereo pairs and motion sequences. The segment tracking task is formulated as follows. Given an image sequence S containing points $p(x, y, z)$ with coordinates (x, y, z) as elements, where x and y label the position within each image, while z labels the frame number, then we want to find a partitioning $\mathbf{P} = \mathrm{P}_1, .., \mathrm{P}_M$ of S in M groups such that

(i) $\mathrm{P}_i \cap \mathrm{P}_j = 0$ and $\mathrm{P}_i \neq \emptyset$ for all groups

(ii) if point $p \in \mathrm{P}_i$, then $s(p, \mathrm{P}_i) > s(p, \mathrm{P}_j)$, where s is a function measuring the average distance of a point to the elements of a group

(iii) if $p(x_i, y_i, z_i) \in \mathrm{P}_r$, then $p(x_i + \triangle x_i, y_i + \triangle y_i, z_i + 1) \in \mathrm{P}_r$, where $\triangle x_i$ and $\triangle y_i$ are the shifts of point $p(x_i, y_i, z_i)$ along the x and y axes, respectively, from frame z_i to frame $z_i + 1$.

To perform this task, we assign a spin variable σ_i (or label) to each pixel (or site) i of the image sequence. To incorporate constraints in form of local correspondence information, we distinguish between neighbors within a single frame (2D bonds) and neighbors across frame (n-D bonds). We create a 2D bond $(i, k)_{2D}$ between two pixels within the same frame with coordinates (x_i, y_i, z_i) and (x_k, y_k, z_k) if

$$\begin{aligned} |(x_i - x_k)| &\leq \varepsilon_{2D} && (1)\\ |(y_i - y_k)| &\leq \varepsilon_{2D} && (2)\\ z_i &= z_k \quad , && (3) \end{aligned}$$

where ε_{2D} is the 2D-interaction range of the spins, a parameter of the system. Across frames, we create a n-D bond $(i, j)_{nD}$ between two spins i and j if

$$\begin{aligned} |(x_i + d^x_{ij} - x_j)| &\leq \varepsilon_{nD} && (4)\\ |(y_i + d^y_{ij} - y_j)| &\leq \varepsilon_{nD} && (5)\\ z_i &\neq z_j && (6)\\ a_{ij} &> \tau \quad , && (7) \end{aligned}$$

where ε_{nD} is the n-D interaction range. The values d^x_{ij} and d^y_{ij} are the shifts of the pixels between frames z_i and z_j along the axis x and y, respectively, obtained from the optic-flow map or disparity map. The parameters a_{ij} are the respective amplitudes (or confidences), and τ is a threshold, removing all local correspondences having a small amplitude.

We define for every bond on the lattice the distance

$$\triangle_{ij} = |g_i - g_j| \quad , \tag{8}$$

where g_i and g_j are the gray (color) values of the pixels i and j, respectively. The mean distance $\bar{\triangle}$ is obtained by averaging over all bonds. We further define an interaction strength

$$J_{ij} = 1 - \triangle/\bar{\triangle} \quad . \tag{9}$$

The spin model is now implemented such a way that neighboring spins with similar color have the tendency to align. We use a q-state Potts model [11] with the Hamiltonian

$$H = -\sum_{\langle ik\rangle_{2D}} J_{ik}\delta(\sigma_i - \sigma_k) - \sum_{\langle ij\rangle_{nD}} J_{ij}\delta(\sigma_i - \sigma_j) \quad . \tag{10}$$

Here, $\langle ik\rangle_{2D}$ and $\langle ij\rangle_{nD}$ denote that i,k and i,j are connected by bonds $(i,k)_{2D}$ and $(i,j)_{nD}$, respectively. The Kronecker δ function is defined as $\delta(a) = 1$ if $a = 0$ and zero otherwise. The segmentation problem is then solved by finding clusters of correlated spins in the low temperature equilibrium states of the Hamiltonian H. The total number M of segments is then determined by counting the computed segments. It is usually different from the total number q of spin states, which is a parameter of the algorithm. Note that the local correspondences used in the algorithm to create n-D bonds are precomputed and are not altered or optimized when computing the equilibrium state. The computation of local correspondences is not the aim of this paper.

We solve this task by implementing a clustering algorithm. In a first step, “satisfied” bonds, i.e. bonds connecting spins of identical spins $\sigma_i = \sigma_j$, are identified. Then, in a second step, the satisfied bonds are “frozen” with a some probability P_{ij}. Pixels connected by frozen bonds define a cluster, which are updated by assigning to all spins inside the same clusters the same new value [12]. In the method of superparamagnetic clustering proposed by Blatt *et al.* (1996) [18] this is done independently for each cluster. In this paper, we will employ the method of energy-based cluster updating (ECU), where new values are assigned in consideration of the energy gain calculated for a neighborhood of the regarded cluster [13,14]. A schematic of the spin system of an image sequence is depicted in Fig. 1A.

The ECU algorithm computing the equilibrium of H consists of the following steps:

1. Initialization: A spin value σ_i between 1 and q is assigned randomly to each spin i. Each spin represents a pixel of the image sequence.

2. Computing bond freezing probabilities: If two spins i and j are connected by a bond and are in the same spin state $\sigma_i = \sigma_j$, then the bond is frozen with a probability

$$P_{ij} = 1 - \exp(-0.5 J_{ij}/T) \quad . \tag{11}$$

Negative probabilities are set to zero.

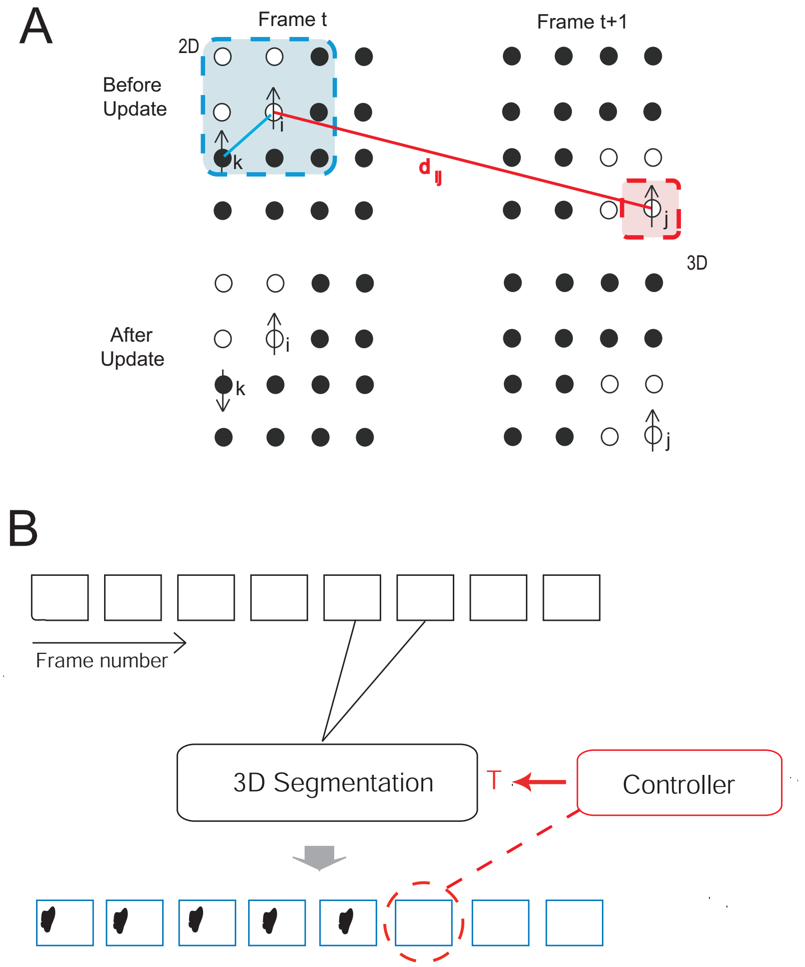

Figure 1: **A** The spin states (upward and downward pointing arrows) of pixels i, k, and j are shown before and after a spin update for two adjacent frames t and $t + 1$ of an image sequence. The white and black circles indicate pixels of small and large gray values, respectively. Pixel i interacts with pixels k and j in its 2D and 3D neighborhood (shaded areas), respectively, which are in the same spin state. **B** Pairwise (3D) segmentation of movies. A feedback controller detects segmentation instabilities and adjusts the control parameter T of the core algorithm (3D segmentation) to recover lost segments.

3. Cluster identification: Pixels which are connected by frozen bonds define a cluster. A pixel belonging to a cluster u has by definition no frozen bond to a pixel belonging to a different cluster v.

4. Cluster updating: We perform a Metropolis update [12,21] that updates all spins of each cluster simultaneously to a common new spin value. The new spin value for a cluster c is computed considering the energy gain obtained from a cluster update to a new spin value w_k. This is done by considering the interactions of all spins in the cluster c with those outside the cluster, assuming that all spins of the cluster are updated to the new spin value w_k, giving an energy

$$E(W_k^c) \;=\; \sum_{i\in c}\Big[-\sum_{\substack{\langle ij\rangle_{2D}\\ c_k\neq c_j}}\eta J_{ij}\delta(\sigma_i-\sigma_j)-\sum_{\substack{\langle ij\rangle_{nD}\\ c_k\neq c_j}}\eta a_{ij}J_{ij}\delta(\sigma_i-\sigma_j)\Big] \tag{12}$$

where $\langle ik\rangle_{2D}, c_k \neq c_j$ and $\langle ij\rangle_{nD}, c_k \neq c_j$ are the noncluster neighborhoods of spin i, and W_k^c symbolizes the respective spin configuration. Here, N is the total number of pixels of the image sequence. The constant η is chosen to be 0.5.

Similar to a Gibbs sampler, the selecting probability $P(W_k^c)$ for choosing the new spin value to be w_k is given by

$$P(W_k^c) = \exp(E(W_k^c))/\sum_{l=1}^{q}\exp(E(W_l^c)) \quad . \tag{13}$$

5. Iteration: The new spin states are returned to step 2 of the algorithm, and steps 2-5 are repeated, until the total number of clusters stabilizes.

6 Segments are defined as groups of correlated spins and can be extracted using a thresholding procedure. All pairs of pixels connected by a bond (i, j) with $c(\sigma_i, \sigma_j) > \theta$ are considered as friends. The function c computes the correlation of the spin states of i and j over several iterations. Then, all mutual friends are assigned to the same segment. Finally, M is determined by counting the total number of segments. In practice, we find it sufficient to take the clusters found in the last iteration as segments.

In an earlier study we had provided evidence that this algorithm obeys *detailed balance*. The full proof shall not be repeated here and can be found in [13]. Detailed balance assures that the proposed algorithm computes an equilibrium spin configuration, i.e. the segmentation, which minimizes the energy function on the labels, and that this is Boltzmann distributed.

The consequence of detailed balance is that spin states can be transferred across image pairs, where spins are being calculated for one pair (the first pair) and then pixels in the next two frames (the second pair) are just assigned these spins from where on a new relaxation process starts (see Fig. 5 for an example). Hence, the relaxation process for the second pair (and all to follow) is much faster when using spin transfer and the system will always arrive at the correct final thermodynamic equilibrium making spin-transfer based segmentation concordant across frames. Note, this property allows consistent segment tracking across many frames without additional assumptions (see Fig. 5), which requires more effort with other methods. This makes this algorithm a possibly very useful and fast enough tool for model free segment tracking applications as will be shown by one example in Fig. 7.

2.2. Feedback control

Segmentation instabilities arising during the tracking process can be partly removed by adjusting the temperature parameter of the core algorithm. The temperature choice affects the formation of segments, hence, a segment which has been lost in a previous frame can sometimes be recovered by increasing the temperature for certain period.

The feedback controller tracks the size of the segments and reacts if the size of a segment changes suddenly. The first controller function

$$P_C^j(t) \;=\; \begin{cases} 1 & \text{if } \triangle S_j(t) < \tau_1, \\ \exp\left[-\triangle S_j(t)/\alpha\right]/\beta & \text{otherwise,} \end{cases} \tag{14}$$

measures the probability of change of segment j, where $S_j(t)$ is the size of segment j at frame t and $\triangle S_j(t) = S_j(t) - S_j(t-1)$, and α, β, τ_1 are parameters. The history of segment j in terms of occurrence is captured by the second controller function

$$P_H^j(t) = 0.4H_j(t-1) + 0.3H_j(t-2) + 0.2H_j(t-3) + 0.1H_j(t-4) \quad , \tag{15}$$

with $H_j(t) = 1$ if $S_j(t) > 0$ and zero otherwise.

Segmentation instabilities may cause a segment to be lost, for example through segment merging or splitting. We define two threshold parameters τ_2 and τ_3. An unexpected segment loss is detected by the controller if the conditions

$$S_j \;=\; 0 \quad , \tag{16}$$

$$P_C \;<\; \tau_2 \quad , \tag{17}$$

$$\text{and } P_H \;>\; \tau_3 \tag{18}$$

are fulfilled. An unexpected segment appearance is detected by the controller if the conditions

$$P_C \;<\; \tau_2 \tag{19}$$

$$\text{and } P_H \;<\; \tau_3 \tag{20}$$

are fulfilled. The identities of the affected segments are stored by the controller. The temperature of the core algorithm is varied using predefined temperature steps $\triangle T$. The segmentation is repeated at the new temperature $T + \triangle T$ for the affected frames. If the lost segments can be recovered at one of these temperatures, the affected segments are relabeled accordingly.

A schematic of the entire system, i.e. core algorithm with feedback control, is presented in Fig. 1B.

3. Results

We apply the algorithm to various synthetic and real image sequences. Unless otherwise indicated, the following parameter values $q = 10$, $\epsilon_{2D} = 1$, and $\epsilon_{nD} = 0$ are used. In Section III.A, the core algorithm is applied to short image sequences and a sensitivity analysis is performed. Temperatures in the range of $T = 0.05 - 0.3$ are chosen for segment tracking, as suggested by the sensitivity analysis. Then, in Section III.B, feedback control is added and the algorithm is applied to movies.

3.1. Verification of core algorithm

We first use stereo image pairs and a three-frame motion sequence to test and verificate the core method (Section II.A) before applying the algorithm to long image sequences, i.e. movies. Stereo images are more suitable to illustrate the basic properties of the algorithm. In the following figures we use spin states instead of cluster labels to limit the total number of colors in the color-coded images to a maximum value of q. Please note that the spin states are not identical to the cluster labels. Spins which belong to the same clusters are always in the same spin state, however, the reverse is not always true. The spin states have to be observed over several iterations to identify clusters as groups of correlated spins.

Illustrating example: artifical solid square

We first demonstrate the algorithm for a synthetic scene which contains a single, solid square, which is shifted by a disparity value of 40 pixels along the x axis, resulting in an image sequence containing two frames, labeled left and right (see Fig. 2A, left and right panel). Each image is of size 100×100 pixels^2. We estimate the disparity of the pixels by applying a stereo algorithm [20], which returns a disparity map D and an amplitude map A, shown in Fig. 2B-C, respectively. The disparities and the respective amplitudes determine whether a pixel in the right frame is a neighbor of a pixel in the left frame. Clustering is performed with $T = 0.01$. The spin states of the spin system are initialized to randomly chosen discrete values between 1 and 10, as depicted in Fig 2D, left and right panel. Then, the system is evolved using the energy-based cluster update algorithm described in the previous section. The spin states after $2, 5, 9, 17, 33, 65$ iterations of the algorithm are shown in Fig. 2E. The process of cluster formation can be easily followed through the iterations. At iteration step 65, the pixels belonging to the square in both the left and the right frame have been assigned to the same cluster, despite incomplete disparity information.

Sensitivity analysis

We investigate the sensitivity of the algorithm in dependence of the parameter T for different levels of Gaussian white noise that we add to the solid-square stereo pair. In Fig. 3A, the ratio of the averaged number of clusters after 100 iterations, computed from 10 runs of the algorithm, and the total number of pixels is plotted as a function of the temperature T for four different realizations of Gaussian noise with standard deviations from the absolute gray-value difference of object and background of 0%, 1%, 10%, and 20%, depicted in red, blue, black, and green, respectively. For this sequence, a perfect segmentation

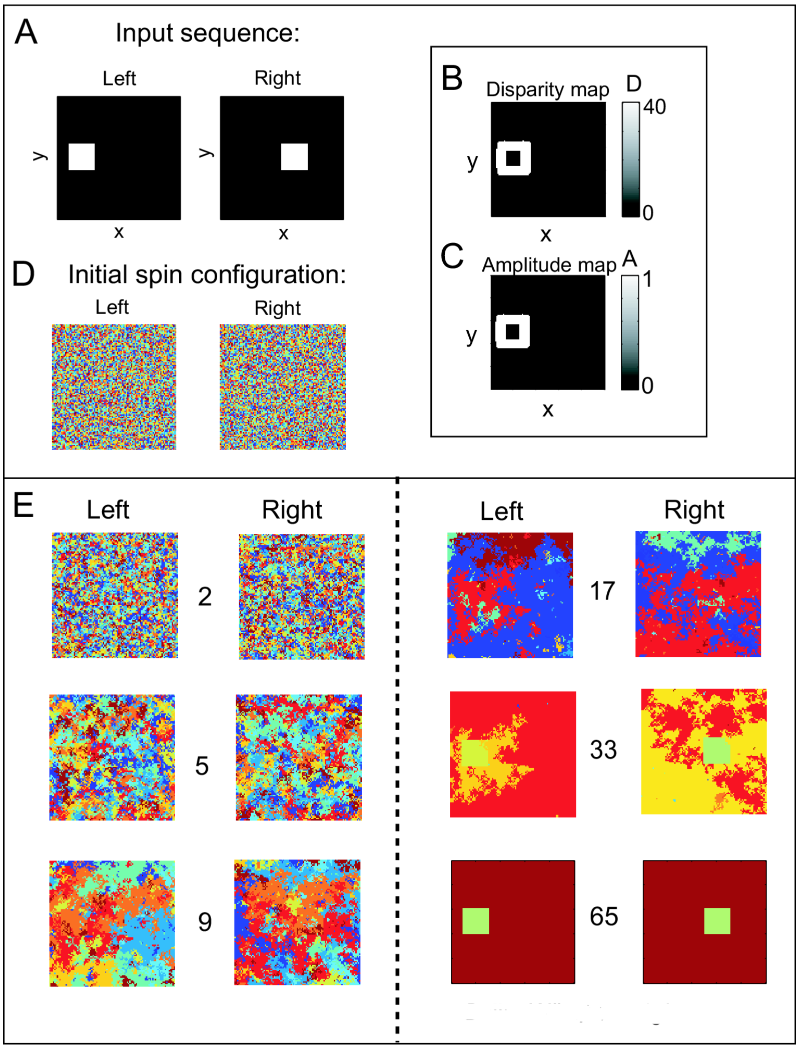

Figure 2: Solid-square stereo pair. **A** The input image consists of a square which is shifted by a disparity value of 40 pixels from one frame to the other, labeled left and right. **B** By applying a stereo algorithm, the disparity map D of the stereo pair could be computed, returning reliable disparity values at the edges. **C** The corresponding amplitude map A reflecting the confidence in the computed stereo values is shown. **D** The spin states are initialized randomly to values between 1 and 10. **E** The spin states after $2, 5, 9, 17, 33,$ and 65 iterations are shown. Progressive agglomeration of clusters of aligned spins can be observed until a stable configuration is reached.

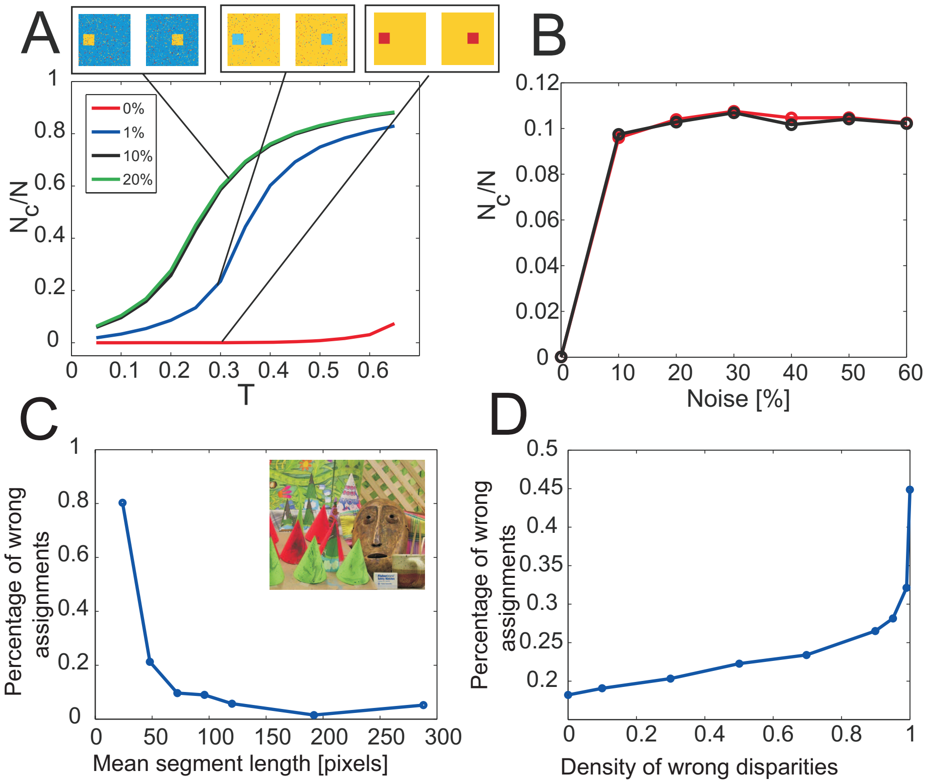

Figure 3: Sensitivity analysis. **A** The ratio of the total number of clusters N_c divided by the total number of pixels N is plotted as a function of the parameter T for different realizations of Gaussian noise, having standard deviations from the absolute gray-value difference of object and background of 0%, 1%, 10%, and 20%. The segmentation results are shown for $T = 0.3$ and different noise levels. **B** The ratio N_c/N is plotted as function of the noise level using the ground-truth disparity map (red line) and the disparity computed with a phase-based stereo algorithm [20] (black line). **C** Percentage of wrongly assigned image points as a function of the mean segment length for the Cones stereo pair (left image see inset). **D** Total percentage of wrongly assigned image points as a function of the density of erroneous disparities.

is achieved for $N_c = 2$, corresponding to $N_c/N = 10^{-4}$. For a noise level of 0%, the performance of the algorithm is only weakly sensitive to changes in temperture (red line). However, when adding noise to the images, the algorithm becomes more sensitive to changes in temperature (blue line), but fast saturates for increasing noise levels (black and green line). For each noise level, the segmentation results are depicted for $T = 0.3$. To establish the 3D neighborhood of image pixels here, the ground-truth disparity map of the image pair was used. However, usually, when adding noise, the quality of the disparity map decreases. Consequently, we also investigated the performance of the algorithm when computing the disparity map with a phase-based stereo algorithm. In Fig. 3B, the ratio of the number of clusters and the total number of image pixels is depicted as a function of the noise level at temperature $T = 0.1$ (black line). The ratios when using the ground truth disparity is plotted for comparison (black line). In this example, the performance is independent of the quality of the disparity map.

We further investigate the performance of the algorithm with respect to establishing correspondences on the example of the Cones stereo pair (URL: vision.middlebury.edu/stereo/). The left frame of the Cones stereo pair is shown as an inset of Fig. 3C. The percentage of wrongly assigned image points was computed independently for every segment, and the average percentage of wrongly assigned image points was plotted as a function of the mean length of the segments (Fig. 3C). A segment of length l contains l^2 image points. The plot demonstrates that the performance of the algorithm is higher for large segments than for small segments, confirming our expectation that color segmentation works best for large uniform image regions. In textured areas, corresponding to very small segment sizes, the performance of the algorithm decreases rapidly.

We also investigated the influence of errors in the precomputed disparity on the performance of the algorithm by replacing disparity values of the ground-truth map randomly by erroneous values ranging from 0 to n, where n is the width of the image. In Fig. 3D the total percentage of wrongly assigned image points (taken from all segments) is plotted as a function of the density of erroneous disparity values. As expected, the performance decreases with increasing error in the disparity map. In summary: one finds that the errors are in general small and the error curves flat for larger segments corresponding to non-textured regions. It is evident that all gray (color) difference based segmentation algorithms in general do not capture textured regions and the increasing errors for small segments reflect this situation. On the other hand, it is very assuring that those segments, which follow from larger consistent gray (color) value similarities, are indeed only little affected by errors in the (stereo-)correspondence map.

Real stereo pair

This stereo pair shows two views of a scene of cluttered objects, i.e., paper boxes, a trash can, and a white Styrofoam object (Fig. 4A, left and right panel). Each image is of size 180×380 pixels^2. This stereo pair is demanding because of the amount of occlusion, the light reflexions, shadows, and the large disparities, which lead to perspective distortions, posing a problem to approaches based on segment matching. The stereo algorithm returns reliable disparity values at the edges (Fig. 4B-C). Otherwise, the amplitude is zero (Fig. 4C). However, when performing clustering with $T = 0.2$, the algorithm is still able to segment most of the boxes into their composite surfaces (Fig. 4D-E). Some of the surfaces are partly shattered though, due to light reflexions and shadows, breaking the uniformity of the surfaces. Both the spin states after 150 and 176 iterations are shown to allow easier identification of correlated

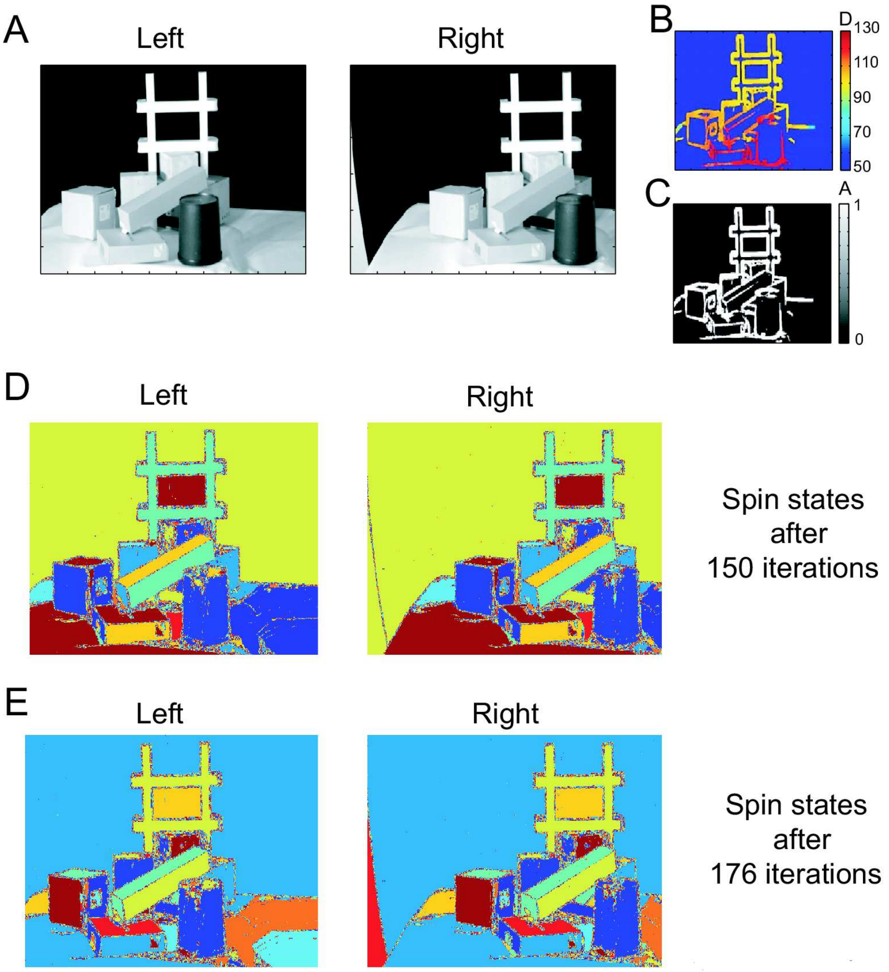

Figure 4: Cluttered-objects stereo pair. **A** Stereo image pair showing a cluttered scene containing a variety of objects. **B-C** The dense stereo algorithm returns mainly disparity information the edges of the objects. **D-E** The spin states computed by the clustering algorithm are shown after both 150 and 176 iterations for easier visual identification of the segments.

spins through visual inspection.

Real motion sequence

So far we had been validating our method using synthetic and real stereo pairs. Now we demonstrate that spatiotemporal synchronization of spins enables segments to be tracked through the frames of real movies.

We apply the core algorithm to three frames of a motion sequence showing a woman walking from the right to the left. The frames are of sizes 118×158 pixels^2 (Fig. 5A). To compute optic flow, any standard algorithm can be used [10], e.g. a differential technique by Lucas and Kanade [22]. Here we used a method proposed in [23]. The performance of the segmentation is only weakly sensitive to the quality of the optic-flow estimation. The optic-flow fields, coding the mapping from the frame t_0 to frame t_1, and from frame t_1 to frame t_2, are depicted in Fig. 5B. The spin states after 100 iterations are shown in Fig. 5C. The algorithm successfully segmented the legs, the arms, a part of the head, and parts of the background, which thus can be tracked from frame to frame. For the highly textured area in the background, no stable 3D clusters could form since the gray-value similarity of neighboring pixels is too low. However, texture could be treated by performing segmentation based on texture similarity instead of color similarity.

When analyzing long motion sequences, it is inefficient to apply the algorithm to all frames at once because the computational costs increase with the number of pixels. Hence, we split the sequences in pairs of two frames at a time, where the last frame of the previous sequence is identical with the first frame of the next sequence. Then, we initialize the spin states of each sequence with the final spin states of the previous sequence. The spin states for the first sequence containing frame t_0 and t_1 after 100 iterations are shown in Fig. 5D. Then, the algorithm is applied to the next pair, containing frame t_1 and t_2, where the spin states of both frame have been initialized to the final spin states of frame t_1 of the previous sequence. The spin states after 13 iterations are shown in Fig. 5E, demonstrating that the number of iterations required to achieve a satisfying segmentation result is greatly reduced by this technique. The number of clusters for the first sequence and the second sequence are displayed as a function of the iteration number in Fig. 5F, dashed and solid line, respectively. The number of clusters for the second sequence is plotted as a function of the iteration number at a different scale (Fig. 5G). Initially, the number of clusters decreases slightly and then approaches a stable state. In a motion sequence, the number of clusters is expected not to change much from one frame to the next. Mainly the boundaries of the clusters reorganize during the first iterations.

The segments of adjacent image pairs are connected as follows. Two segments belonging to the segmentation of frame t_1 of pair $\{t_0, t_1\}$ and frame t_1 of pair $\{t_1, t_2\}$, respectively, are assigned the same label if they occupy the same region in image frame t_1. This way we can track the segment through the whole sequence.

3.2. Segment tracking with feedback stabilization

We add feedback control (see Section II.C) with parameters $\alpha = 200$ pixels, $\beta = 0.8$, $\tau_1 = 50$ pixels, $\tau_2 = 0.9$, and $\tau_3 = 0.6$ to the core algorithm with temperature $T = 0.05$ and apply the algorithm to long

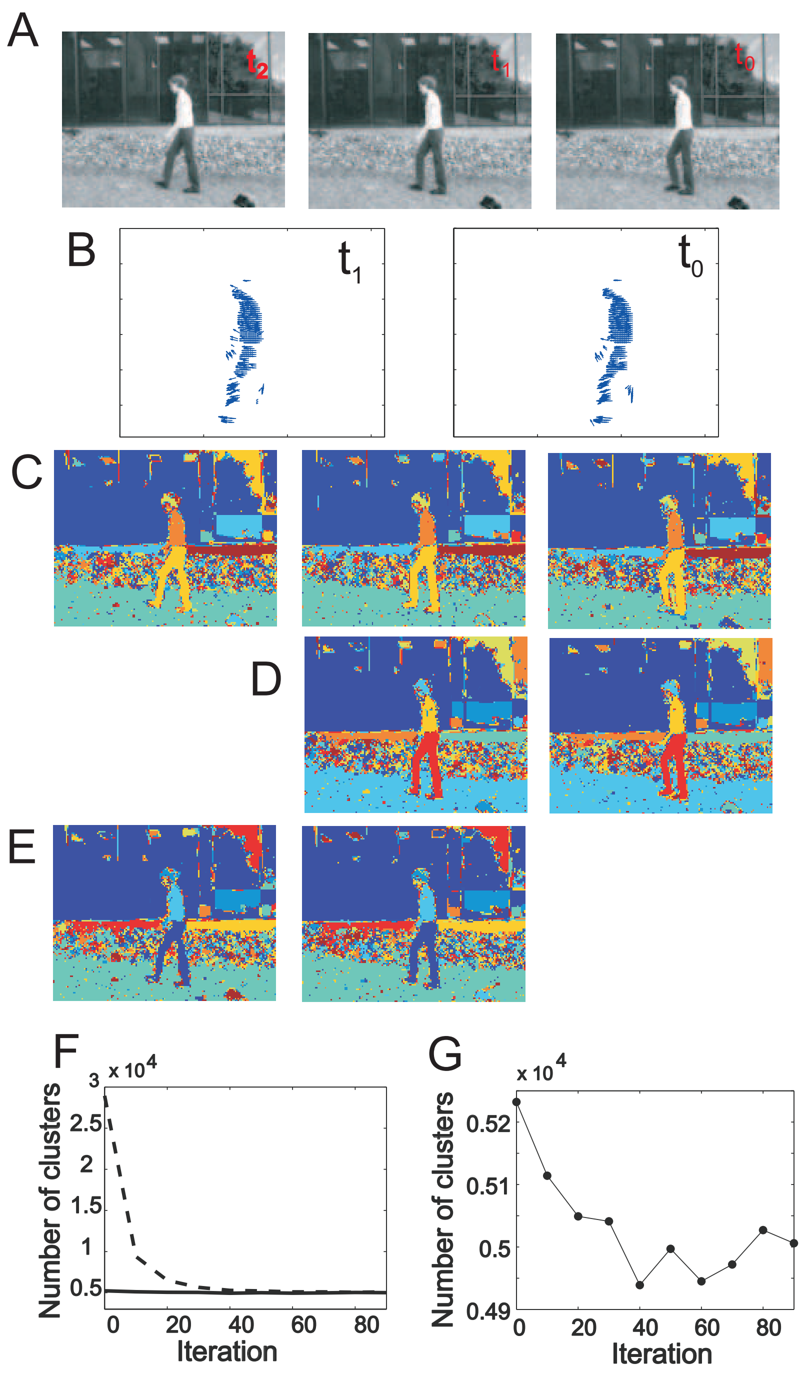

Figure 5: **A** Three frames of a walking sequence, labeled t_0, t_1, and t_2, respectively. **B** Optic-flow vectors coding the mapping from frame t_0 to t_1, and t_1 to t_2. **C** Spin states after 100 iterations. **D** Spin states for a sequence consisting of frames, t_0 and t_1. **E** Spin states for a sequence consisting of t_1 and t_2. **F** The number of clusters as a function of the iteration number for the first sequence containing frame t_0 and t_1 (dashed line) and for the second sequence containing frame t_1 and t_2 (solid line). **G** Enlarged plot of the second sequence.

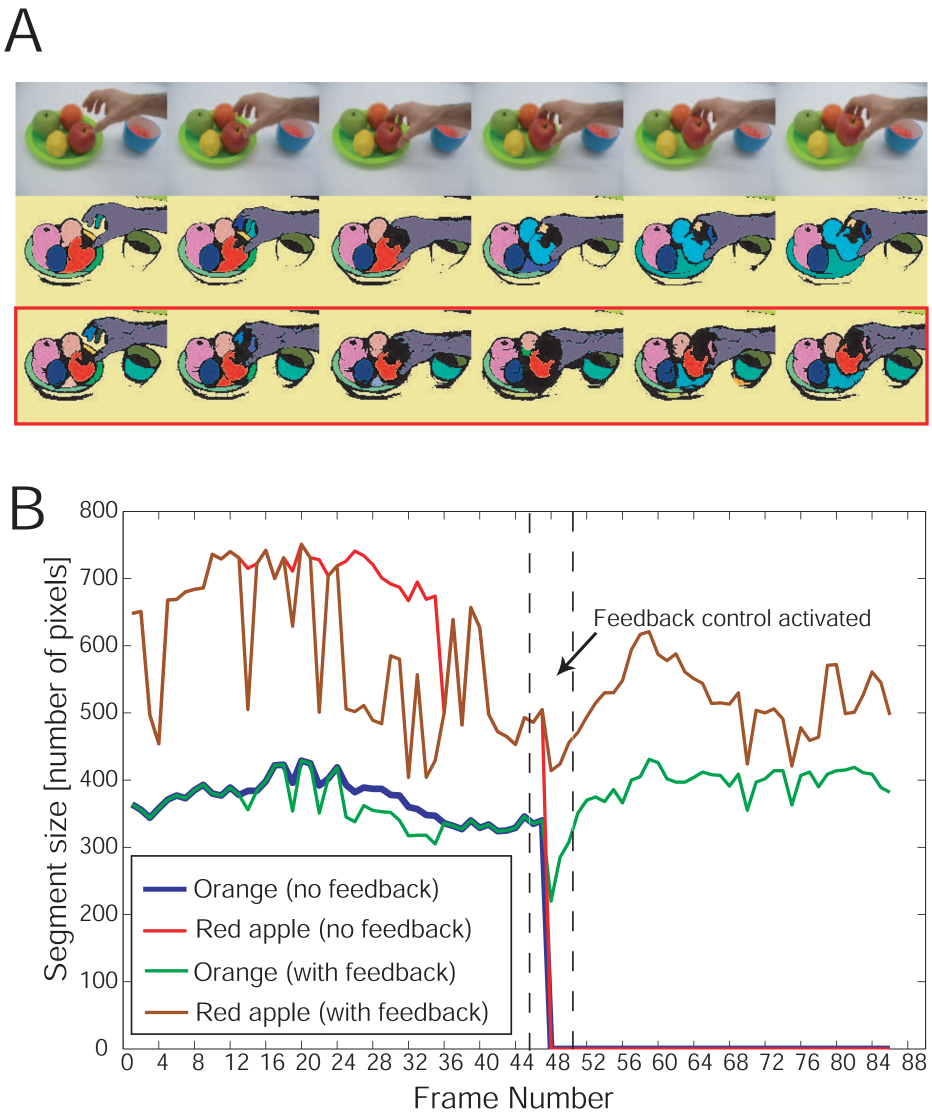

Figure 6: Feedback control for segmentation stabilization. **A** A few frames of a movie showing a hand taking a red apple from a plate are shown together with the results of the core algorithm without and with feedback control (upper, middle, and lower panel, respectively). **B** The segment size is plotted as a function of the frame number for the segments representing the red apple and the orange without and with feedback control, depicted as red, blue, brown and green lines, respectively. At frame number 45 the segment sizes of the red apple and the orange drop unexpectedly to zero (red and blue lines), and the feedback control is activated, increasing the temperature T until the original segments are recovered (brown and green lines).

image sequences. The first movie shows a hand taking a red apple from a plate with several fruits. A few frames of the movie are depicted in the upper panel of Fig. 6A. If the core algorithm is applied at constant temperature without feedback control, the red segment and the light pink segment, representing the respective parts of the red apple and the orange, are lost at frame number 45 due a segmentation instability: The red segment and the light pink segment merge and form a new segment, colored in light blue (see Fig. 6A, middle panel). If feedback control is included, this segmentation instability is detected and the original segments can be recovered by increasing the temperature in steps of $\triangle T = 0.15$. As a consequence, the segments can be continuously tracked, as shown in Fig. 6A (lower panel). The segments representing the cup could be recovered using the same mechanism.

The work of the feedback controller is further illustrated in Fig. 6B, where the segment size is plotted as a function of the frame number for the segments representing the red apple and the orange without and with feedback control, depicted as red, blue, brown and green lines, respectively. At frame number 45 the segment sizes of the red apple and the orange drop unexpectedly to zero (red and blue lines), thus indicating a segmentation instability (see Section II.C). As a consequence, the feedback controller is activated and the temperature of the core algorithm is increased until the original segments are recovered (brown and green lines). The results for the whole movie are shown in Fig. 7A.

We further applied the algorithm to another movie, showing the filling of a cup with sugar (Fig. 7B). The movie is challenging because it contains light reflexions and changing shadows. However, the algorithm is capable of tracking the main segments of the movie, i.e. the two cups and the hand.

We obtained similar segment-tracking results for other real movies, e.g. Moving Object, Making Sandwich, Opening a Book. These results can be found at http://www.nld.ds.mpg.de/ eren/Movies/Movies.html.

4. Discussion

We presented an algorithm for model-free segment tracking based on a novel, conjoint framework, combining local correspondences and image segmentation to synchronize the segmentation of adjacent images. The algorithm provides a partitioning of the image sequence in segments, such that points in a segment are more similar to each other than to points in another segment, and such that corresponding image points belong to the same segment. We tested the method on various synthetic and real image sequences, and showed stable and reliable results overall, thus fulfilling the most important requirement of segmentation algorithms. The method leads to the formation of stable region correspondences despite largely incomplete disparity or optic-flow maps. Similar algorithms for the extraction of region correspondences could potentially be constructed using other image segmentation algorithms, i.e. methods based on agglomerative clustering [24,25]. We decided to use physics-based model for its conceptual simplicity which allowed us to integrate local correspondence information in a straightforward way. It further has the advantage that the interacting parts are inherently converging to the equilibrium state and thus are not being trapped in local extrema (detailed balance). As a consequence, the result is independent of the initial conditions, allowing us to apply the algorithm to long image sequences via spin-states transfer. This allows for consistent segment tracking across many frames without additional assumptions, which is most of the time not immediately possible with other methods. In addition, no assumptions about the underlying data are required, e.g. the number of segments, leading to a model-free segmentation. This has the consequence that a single pixel of distinct gray value (compared to its

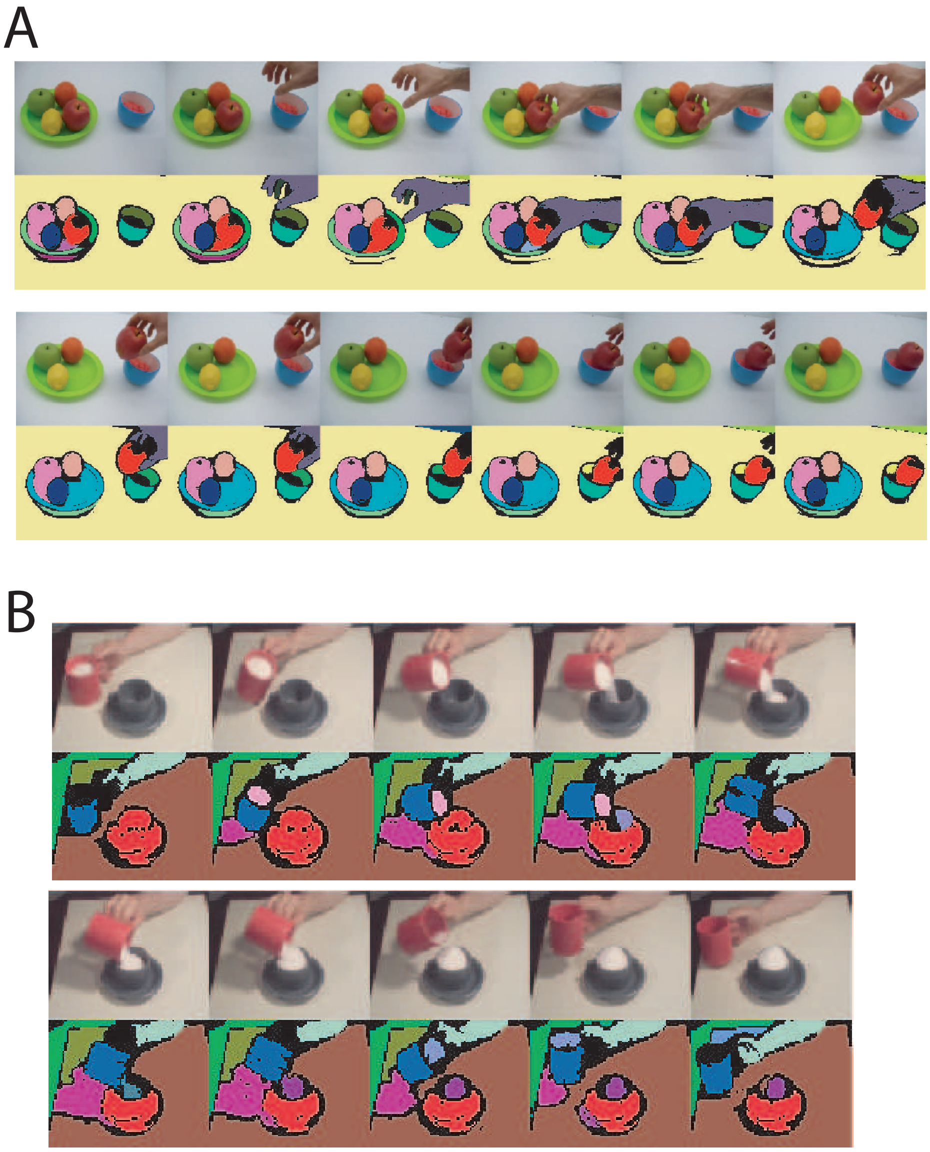

Figure 7: Segment tracking for real movies. **A** The algorithm (with feedback control) is applied to a movie showing a hand taking an apple from a plate (upper panel). The corresponding segment-tracking results are depicted below. **B** The results of the algorithm (with feedback control) for a movie showing the filling of a cup are shown. Segment-tracking results for other real movies can be found at http://www.nld.ds.mpg.de/ eren/Movies/Movies.html

neighbors) might define a single segment. In algorithms, which partition the image into a fixed and usually small number of segments, this phenomenon does not occur. This, however, is a problem as in all realistic situations one never knows how many segments exist and self-adjustment of the total number of segments is, thus, usually desired as compared to a pre-defined maximal number.

We further introduced a feedback controller which allows to detect segmentation instabilities, i.e. merging and splitting of segments. The feedback controller adjusts the control parameter of the core algorithm in order to recover the original segments. This allows to keep track of segment even in long movies.

Segment tracking has been performed previously in the context of video segmentation [2,3,4,5,6,7,8,9]. Our method differs from these approaches in the choice of the segmentation algorithm, the way linking is achieved, and the addition of a feedback controller which detects segmentation instabilities. Superparamagnetic clustering allows a model-free unsupervised segmentation of the image sequences, including a self-adjustment of the total number of segments. Linking is introduced trough local correspondence information which synchronizes the spin-relaxation process of adjacent images. This approach has the advantage that the partitions of adjacent images are less prune to partitioning instabilities. Further, our method does not require corresponding regions to fulfill any segment similarity criterium. Finally, feedback controll allows segmentation instabilites occuring in long sequences to be removed by assuming that "good" segments change their size in a continuous "predictable" manner.

There have been a few other approaches combining image segmentation with correspondence information. The work by Toshev *et al.* [26] uses a joint-image graph containing edges representing intra-image similarities and inter-image feature matches to compute matching regions. Joint segmentation has also been employed by Rother *et al.* [27] using histogram matching.

Vision problems have been formulated in terms of energy minimization in many ways before. The major challenge of these approaches lies in the computation of the global minimum, which is often difficult in particular for interesting energy functions. Various techniques have been proposed, such as variational methods [28], graph cuts [29,30,31,32,33,34,35,36], dynamic programming [37], simulated annealing [38], or relaxation labeling [39,40,41]. Superparamagnetic clustering has been shown to equilibrate to a global minimum for the Potts model used in this work [13]. The work of [42], which computes disparities by minimizing energy functions through an inference algorithm defined on graph partitions, shows some similarities to the core algorithm proposed in this paper, even though it has been applied to a different problem, i.e. stereo matching. However, in the algorithm of Barbu and Zhu (2005) the number of segments is a parameter to the algorithm. Unlike in superparamagnetic clustering, it is assumed that there is a natural set of labels (disparities), and a data penalty function, which makes some pixel-label assignments more likely than others. These assumptions will lead to a violation of detailed balance and spin-transfer is not possible in this framework.

The algorithm has potential applications in model-free moving object detection and tracking by merging coherently moving segments (Gestalt law of common fate). The method is further applicable to action-recognition tasks, where certain characteristic action patterns are inferred from the spatiotemporal relationships of segments. First results for this problem are reported in [19]. In the future, texture cues may be incorporated into the algorithm to allow tracking of segments defined by texture.

The algorithm is not yet working in real time. However, we are currently working on a parallel implementation of the algorithm on GPUs to achieve real-time segment tracking for robot applications. Our goal is the development of a vision-front end for real-time video segment tracking on top of which other algorithms, i.e. moving-object detection/tracking and action recognition, can be applied.

Acknowledgment

We thank Sinan Kalkan for valuable discussion. The work has received support from BMBF funded BCCN Göttingen, the EU Project DRIVSCO under Contract No. 016276-2, and the EU Project PACO-PLUS under Contract No. 027657.

References

1. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: a survey. *ACM Comput. Surv.* **2006**, *38*, 1–45.
2. Choi, J.G.; Lee, S.W.; Kim, S.D. Spatio-temporal video segmentation using a joint similarity measure. *IEEE Trans. Circuits Syst. Video Technol.* **1997**, *7*, 279–286.
3. Salembier, P.; Marques, F. Region-based representations of image and video: Segmentation tools for multimedia services. *IEEE Trans. Circuits Syst. Video Technol.* **1999**, *9*, 1147–1169.
4. Tuncel, E.; Onural, L. Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modeling. *IEEE Trans. Circuits Syst. Video Technol.* **2000**, *10*, 776–781.
5. Deng, Y.; Manjunath, B.S. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Pattern Anal. Machine Intell.* **2001**, *23*, 800–810.
6. Patras, L.; Hendriks, E.A.; Lagendijk, R.L. Video segmentation by MAP labeling of watershed segments. *IEEE Trans. Pattern Anal. Machine Intell.* **2001**, *23*, 326–332.
7. Yokoyama, Y.; Miyamoto, Y.; Ohta, M. Very low bit rate video coding using arbitrarily shaped region-based motion compensation. *IEEE Trans. Circuits Syst. Video Technol.* **1995**, *5*, 500–507.
8. Wang, D. Unsupervised video segmentation based on waterheds and temporal tracking. *IEEE Trans. Circuits Syst. Video Technol.* **1998**, *8*, 539–546.
9. Mezaris, V.; Kompatsiaris, I.; Strintzis, M.G. Video object segmentation using bayes-based temporal tracking and trajectory-based region merging. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 782–795.
10. Barron, J.L.; Fleet, D.J.; Beauchemin, S.; Burkitt, T. Performance of optical flow techniques. *Int. J. Comput. Vis.* **1994**, pp. 43–77.
11. Potts, R.B. Some generalized order-disorder transformations. *Proc. Cambridge Philos. Soc.* **1952**, *48*, 106–109.
12. Swendsen, R.; Wang, S. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters* **1987**, *76*, 86–88.
13. Opara, R.; Wörgötter, F. A fast and robust cluster update algorithm for image segmentation in spin-lattice models without annealing – visual latencies revisited. *Neural Computation* **1998**, *10*, 1547–1566.
14. von Ferber, C.; Wörgötter, F. Cluster update algorithm and recognition. *Physical Review E* **2000**, *62*, 1461–1664.

15. Geman, D.; Geman, S.; Graffigne, C.; Dong, P. Boundary detection by constrained optimization. *IEEE Trans. Pattern Analysis Machine Intelligence* **1990**, *12*, 609–628.
16. Vorbrüggen, J.C. *Zwei Modelle zur datengetriebenen Segmentierung visueller Daten*. Frankfurt am Main, Harri Deutsch, Thun, 1976.
17. Eckes, C.; Vorbrüggen, J.C. Combining data-driven and model-based cues for segmentation of video sequences. *WCNN World Conference on Neural Networks, San Diego* **1996**.
18. Blatt, M.; Wiseman, S.; Domany, E. Superparametric clustering of data. *Physical Review Letters* **1996**, *76*.
19. Aksoy, E.E.; Wörgötter, F.; Dellen, B. Recognizing Object-Action Relations from Semantic Scene Graphs. *9th IEEE-RAS International Conference on Humanoid Robots (Humanoids'09)(submitted)* **2009**.
20. Sabatini, S.P.; Gastaldi, G.; Solari, F.; Diaz, J.; Ros, E.; Pauwels, K.; Hulle, K.M.M.V.; Pugeault, N.; Krüger, N. Compact and accurate early vision processing in the harmonic space. *Int. Conf. on Computer Vision Theory and Applications (VISAPP), Barcelona* **2007**.
21. Metropolis, N.; Rosenbluth, A.W.; M. N. Rosenbluth, A.H.T.; Teller, E. Equations of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087–1091.
22. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. *Proc. DARPA IU Workshop* **1981**, pp. 121–130.
23. Dellen, B.; Wörgötter, F. A local algorithm for the computation of optic flow via constructive interference of global Fourier components. *British Machine Vision Conference* **2008**.
24. Ward, J.H. Hierarchical grouping to optimize and objective function. *J. Am. Statistical Assoc.* **1963**, *58*, 236–244.
25. Fränti, P.; Virmajoki, O.; Hautamaäki, V. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1875–1881.
26. Toshev, A.; Shi, J.; Daniilidis, K. Image matching via saliency region correspondences. *IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis* **2007**.
27. Rother, C.; Minka, T.; Blake, A.; Kolmogorov, V. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society: Washington, DC, USA, 2006, pp. 993–1000.
28. Horn, B.K.P.; Schunk, B. Determining optical flow. *Artificial Intelligence* **1981**, *17*, 185–203.
29. Greig, D.; Porteous, B.; Seheult, A. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society B* **1989**, *31*, 271–279.
30. Ferrari, P.; Frigessi, A.; de Sa, P. Fast approximate maximum a posteriori restoration of multicolour images. *Journal of the Royal Statistical Society B* **1995**, *57*, 485–500.
31. Roy, S.; Cox, I. A maximum-flow formulation of the n-camera stereo correspondence problem. 1998.
32. Ishikawa, H.; Geiger, D. Occlusions, discontinuities, and epipoloar lines in stereo. *European Conference on Computer Vision* **1998**, pp. 232–248.
33. Boykov, Y.; Veksler, O.; Zabih, R. Markov random fields with efficient approximations. *IEEE Conference on Computer Vision and Pattern Recognition* **1998**, pp. 648–655.

34. Veksler, O. Image Segmentation by nested cuts. *IEEE Conference on Computer Vision and Pattern Recognition* **2000**, *1*, 339–344.
35. Wu, Z.; Leahy, R. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1993**, *15*, 1101–1113.
36. Gdalyahu, Y.; Weinshall, D.; Werman, M. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2001**, *23*.
37. Amini, A.; Weymouth, T.; Jain, R. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1990**, *12*, 855–867.
38. Geman, D.; Geman, S. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1984**, *6*, 721–741.
39. Chou, P.; Brown, C. The theory and practice of Bayesian image labeling. *International Journal of Computer Vision* **1990**, *4*, 185–210.
40. Rosenfeld, A.; Hummel, R.A.; Zucker, S.W. Scene labeling by relaxations operations. *IEEE Transactions on Systems, Man, and Cybernetics* **1976**, *6*, 420–433.
41. Szeliski, R.S. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision* **1990**, *5*, 271–302.
42. Barbu, A.; Zhu, S.C. Generalizing Swendson-Wang to sampling arbitrary posterior probabilities. *Trans. Pattern Anal. Mach. Intell.* **2005**, pp. 1239–1253.

© August 6, 2009 by the authors; submitted to *Sensors* for open access under the terms and conditions of the Creative Commons Attribution license http://creativecommons.org/licenses/by/3.0/.