Computer Vision Winter Workshop 2009, Adrian Ion and Walter G. Kropatsch (eds.)
Eibiswald, Austria, February 4–6
Publisher: PRIP, Vienna University of Technology, Austria

Tracking Articulated Objects Using Structure*

Nicole M. Artner[1], Adrian Ion[2], and Walter Kropatsch[2]

[1] Austrian Research Centers GmbH - ARC, Smart Systems Division, Vienna, Austria
nicole.artner@arcs.ac.at

[2] PRIP, Vienna University of Technology, Austria
ion@prip.tuwien.ac.at, krw@prip.tuwien.ac.at

Abstract *This paper extends our previous work in rigid object tracking using a spring system to track articulated objects. The idea is to represent each part of an articulated object with a spring system and connect them via articulation points. Articulation points are detected automatically by observing the motion of the object and integrated into the spring systems of the rigid parts. Experiments with real and synthetic video sequences show promising results and improvements over the rigid, previous approach. In a detailed discussion the parameters of the approach are analyzed and future plans are mentioned.*

1 Introduction

In the context of this paper, the *properties* of a scene can be divided as follows.

***Intrinsic*:** They refer to the nature or the constitution of the object and are considered wholly independent of any external factor or other object. For example: size, texture, rigid or non-rigid, number and position of articulation points, motion model, ...

***Extrinsic*:** These properties concern the environment (context) of the object. For example properties describing the influence of the environment on the object (i.e. background and occlusions) and the relationships between object and environment (i.e. other objects).

Depending on the source for the properties they can be separated in two categories.

Perception based: Values of properties are directly extracted from the image (e.g. edge segments, color regions, corner points, histograms, ...).

Knowledge based: Information extracted from previous experiences. It does not have to be present in the pool of perception based features at that moment. For example: structure of an object (rigid, articulated) and the space-time continuum (occluded object did not disappear).

Tracking articulated objects undergoing non-rigid motion and occlusions is a challenging task in computer vision. To solve this task one has to consider combining information and knowledge about the target object and its environment. This means covering both intrinsic and extrinsic properties determined by perception and knowledge.

In [3], initial steps toward solving this tracking task have been made. A spring system was employed to coordinate the tracking of multiple features of a rigid object (recall in Section 3). The spatial relationships between the features encoded in the spring system increased the robustness with respect to similar neighboring objects and considerable amount of occlusions. The approach uses intrinsic object properties and information from perception.

In [10], the focus was on automatically deriving the model of an articulated object and its environment (background, other objects), to cover intrinsic and extrinsic properties. An important finding of this approach is that structurally relevant features, like articulation points, are not always visually salient (trackable). Thus, searching for articulation points only in the set of visual features is not feasible.

This paper is a first attempt in the plan to combine all possible permutations of intrinsic and extrinsic properties retrieved via perception and knowledge. There is a vast amount of work in using statistical approaches to solve this or a similar problem (Section 2). In our approach we try to stress solutions that emerge from the underlying structure, instead of using structure to verify hypothesis obtained by other means. This paper is an extension of the approach in [3], to track objects consisting of several rigid parts connected by articulation points. Every part is represented by a spring system encoding the spatial relationships of the features describing it. The articulation points of the object are found through observation of the behavior/motion of the object parts over time. They are integrated into the spring systems of the connected rigid parts and their energy minimization process and impose additional distance constraints.

This paper is organized as follows. In Section 2 a short overview of the related work is given. Section 3 recalls the work done in [3] to track rigid objects. In Section 4 the concept of articulation and its integration is explained. Section 5 describes improvements of the approach in [3], in addition to the articulation. Experiments are in Section 6, and Section 7 discusses the parameters of our approach. Conclusion and future plans are in Section 8.

*Partially supported by the Austrian Science Fund under grants P18716-N13 and S9103-N13.

2 Related work

Related work in the narrow field of approaches employing spring systems is [8, 7, 14, 12]. The idea to describe the relationships of the parts of an object in a deformable configuration - spring system - has already been proposed in 1973 by Fischler et al. [8]. Felzenszwalb et al. employed this idea in [7] to do part-based object recognition for faces and articulated objects (humans). Their approach is a statistical framework minimizing the energy of the spring system learned from training examples using maximum likelihood estimation. The energy depends on how well the parts match the image data and how well the relative locations fit into the deformable model. Ramanan et al. apply in [14] the ideas from [7] in tracking people. They model the human body with colored and textured rectangles, and look in each frame for likely configurations of the body parts. Mauthner et al. present in [12] an approach using a two-level hierarchy of particle filters for tracking objects described by spatially related parts in a mass spring system.

In comparison to the related work above we do not directly employ a statistical framework or look for hypotheses for the target object in the whole image. This approach solves the recognition and association problem by locally minimizing the energy of the spring system.

Related work in a broader field also includes the work done in tracking and analysing the movements of articulated objects - in the most cases humans. There is a vast amount of work in this field as can be seen in the surveys [9, 13, 1, 2]. It would go beyond the scope of this paper mentioning all of this work. Interesting to know is that early works even date back to the seventies, where Badler and Smoliar [4] discuss different approaches to represent the information concerning and related to the movement of the human body. Our approach is similar to the *Eshkol-Wachmann* notation.

3 Recall: Tracking rigid objects with a spring system

In [3], deterministic tracking of multiple features of an object is combined with a graph representation that encodes the structural dependencies between the features. A spring system is used to model the dynamic behavior of the structural dependencies.

To identify suitable features of a rigid object, the Maximally Stable Extremal Regions (MSER) detector [11] is applied to a region of interest. MSER extracts regions which are well-defined (high color uniformity). An attributed graph (AG), representing the structural dependencies, is created by associating a vertex to each region. The corresponding color histograms of the underlying regions are the attributes of the vertices. With a Delaunay triangulation the edges between the vertices are inserted and the spatial relationships between the regions are defined.

This approach uses the mode seeking property of the Mean Shift algorithm to associate the vertices of the AG over adjacent frames. Therefore, a Mean Shift tracker is initialized on each vertex of the graph and the color histograms of the vertices become the target models $\hat{q}$ for the tracker. The implementation of the tracking with Mean Shift follows the ideas in [6, 5].

During object tracking the color histograms of the AG and "spring-like" edge energies of the structure are used to carry out gradient ascent energy minimization on the joint distribution surface (color similarity and structure).

3.1 Graph relaxation

The graph relaxation step introduces a mechanism which - upon drift in the tracking results - imposes structural constraints on the mode seeking process of Mean Shift. It is the realization of the spring system in this approach. As the tracked objects are rigid, the objective of the relaxation is to maintain the tracked structure as similar as possible to the initial structure. Graph relaxation is used to minimize the dissimilarity between the initial structure of the object and the tracked structure. This is an energy minimization problem on the total energy E_t of the structure.

Because the initial structure is considered as the "true" object structure, the total energy of the structure in the initial state is 0. Due to the spatial tracking errors of the Mean Shift tracker, during the tracking process E_t usually changes. The structural energy E_t is computed using the concept of "spring-like" edges between vertices

$$E_t = \sum_e k \cdot |e' - e|^2, \tag{1}$$

where e' and e denote the deformed and undeformed edge lengths and k is the elasticity constant of the edge (spring). The variations of the edge lengths and their directions are used to determine a structural offset component for each vertex. The offset vector is the direction where a given vertex should move such that its edges restore their initial length and the energy of the structure is minimized. This structural energy minimization offset vector $\vec{O}$ is calculated for each vertex v as follows:

$$\vec{O}(v) = \sum_{e \in E(v)} k \cdot (|e'| - |e|)^2 \cdot (-\vec{d}(e, v)), \tag{2}$$

where $E(v)$ are all the edges e incident to vertex v, k is the elasticity constant of the edges in the structure and $\vec{d}(e, v)$ is the unitary vector in the direction of edge e that points toward v.

In Figure 1(a) three possible states of an edge are shown. The first state is the initial one. In the next state the edge is contracted, so that the offset vector $\vec{O}$ will force vertex B' to move, enlarging the edge length back to its initial length. The third and last case shows an edge that is too long, so $\vec{O}$ will tend to contract it. Figure 1(b) shows how the sum of the offset vectors of each edge would move vertex B' to its structurally correct position B.

3.2 Combining Mean Shift and graph relaxation

Graph relaxation is embedded into the iterative process of tracking with Mean Shift. For every frame, Mean Shift and structural iterations are performed until a maximum number of iterations is reached ϵ_i, or the graph structure attains equilibrium i.e. its total energy is beneath a certain threshold ϵ_e. To compute the position of each region (vertex), Mean Shift offset and structure-induced offset are combined using

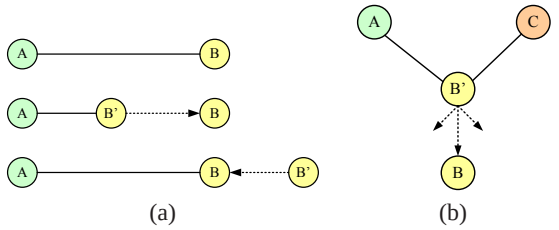


Figure 1: Edge relaxation examples. B is the initial state of the vertex and B' the deformed one.

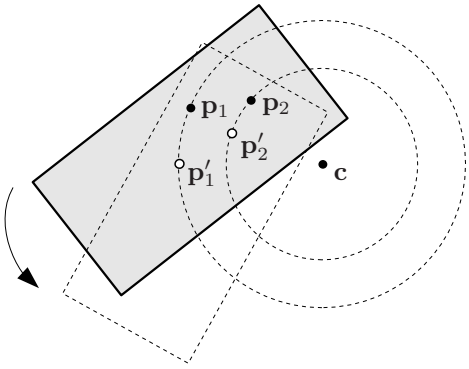


Figure 2: Rotation of rigid part around articulation point $\mathbf{c}$. $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}'_1$, $\mathbf{p}'_2$ are the vertices at time t respectively $t+\delta$.

a mixing coefficient g called *gain* ($g = 5$). The ordering of the region selection during the iterations is randomized.

As the experiments in [3] demonstrated, the joint use of Mean Shift and structural constraints significantly improves tracking in the presence of occlusions or in cases when multiple similarly colored nearby objects are tracked on patterned background. The calculation of the 3D color histograms for the Mean Shift iteration represents the biggest part of the computational costs. Because of this, one could say that the complexity of the algorithm scales linearly with the number of regions.

4 Articulation: Bending the structure

Articulated motion is a piecewise rigid motion, where the rigid parts conform to the rigid motion constraints but the over all motion is not rigid [2]. An *articulation point* connects two rigid parts. The parts can move independent to each other, but their distance to the articulation point remains the same (see Figure 2). This paper considers articulation in the image plane (1 degree of freedom).

4.1 Imposing articulation

As before (Section 3), rigid parts are tracked using a competition between the tracker (Mean Shift) and the graph structure (spring system). Two vertices of each rigid part are connected with the common articulation point[2]. These two *reference vertices* constrain the distance to the articulation point, of all other vertices in the same part.

Each rigid part is optimized iteratively including all articulation points it is connected to, but independently from all other (rigid) parts. Transfer of "information" between the parts of the articulated objects is achieved through the articulation points, as they are present in the computation of all incident parts.

Important features of the structure of an object do not necessarily correspond to easily trackable visual features, e.g. articulation points can be occluded, or can be hard to track and localize (e.g. textured regions for trackers of homogeneous blobs, and vice-versa). Articulation points are thus not associated to a tracked region (as opposed to tracked features of the rigid parts).

The position of the articulation point can be given or compute as explained in Section 4.2. To derive the position of the articulation point in each frame of the video, the following procedure is applied. First, in the frame in which the position of the articulation point is detected, a local coordinate system is created for each rigid part by using the reference vertices. In Figure 3 this concept is shown, where $\mathbf{p}_1, \mathbf{p}_2, \mathbf{c}, X, Y$ are the tracked vertices, articulation point (rotation center) and coordinate system at time t; $\mathbf{p}'_1, \mathbf{p}'_2, \mathbf{c}', X', Y'$ at time $t+\delta$; $\mathbf{o}$ is the offset (translation), and θ is the rotation angle. The position of the articulation point, in that coordinate system, is computed and associated to the part.

At any time, having the tracked reference vertices enables determining the local coordinate system and the position of the articulation point relative to that coordinate system.

Algorithm 1 Algorithm for tracking articulated objects.

```
1:  PROCESSFRAME
        ε_e threshold total energy of structure
        ε_i threshold maximum number of iterations
2:      i ← 1                                          ▷ iteration counter
3:      while (i < ε_i and E_t > ε_e) do
4:          for every rigid part do
5:              define region order depending on B
6:              for every region do
7:                  do Mean Shift iteration
8:                  do structural iteration
9:                  calculate mixing gain g
10:                 mix offsets depending on g and set new position
11:             end for
12:         end for
13:         calculate current position of articulation point
14:         for every rigid part do
15:             define region order depending on B
16:             for every region do
17:                 do Mean Shift iteration
18:                 do structural iteration including articulation point
19:                 calculate mixing gain g
20:                 mix offsets depending on g and set new position
21:             end for
22:         end for
23:         i ← i + 1
24:         E_t ← determine total energy of spring system
25:     end while
26: end
```

For each frame the steps in Algorithm 1 are executed. First there is an independent optimization of each rigid part (see Section 3). This is followed by estimating the position of the articulation point using the reference vertices of each part. Then the hypothesis of the parts for the position of the

[2] One could consider connecting all vertices of a part, but this would unnecessarily increase the complexity of the optimization process.

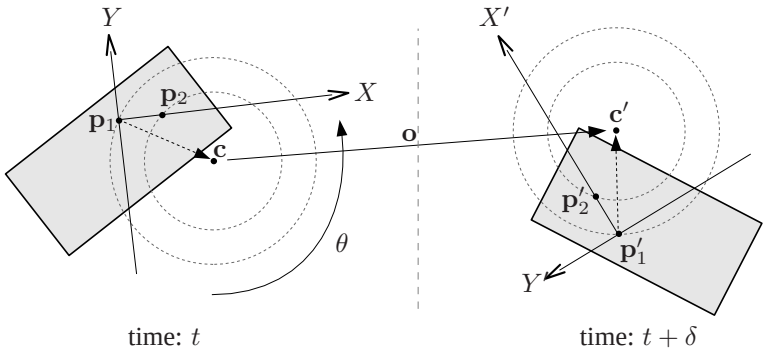


Figure 3: Encoding and deriving of articulation point in the local coordinate system, during two time steps: t and $t+\delta$.

articulation point are combined using the gain a:

$$a_i = \frac{Z_i}{\sum_{k=1}^{m} Z_k} \quad Z_i = \sum_{j=1}^{v_i} B_{ij} \tag{3}$$

where Z_i is the sum of all Bhattacharyya coefficients (see Equation 6) of parts i with v_i regions/vertices, and a_i is the gain for part i weighting its influence on the position of the articulation point. a_i depends on the correspondence of its color regions with the target models. Afterward each part is optimized including the articulation point at the determined position.

4.2 Determining the articulation point

The following approach is used to compute the position of the articulation point. The determined position is then mapped to the local coordinate systems and used for tracking as mentioned above.

For discrete time steps (e.g. frames of a video) the motion of an articulated object can be modeled by considering rotation and translation separately:

$$\mathbf{p}' = \text{translate}(\text{rotate}(\mathbf{p}, \mathbf{c}, \theta), \mathbf{o}),$$

where $\mathbf{p}$ is the vertex at time t and $\mathbf{p}'$ is the same vertex at time $t+\delta$. $\mathbf{p}'$ is obtained by first rotating $\mathbf{p}$ around $\mathbf{c}$ with angle θ and then translating it with offset $\mathbf{o}$. More formally,

$$\mathbf{p}' = (R * (\mathbf{p} - \mathbf{c}) + \mathbf{c}) + \mathbf{o}, \tag{4}$$

where R is the 2D rotation matrix with angle θ given by:

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Equation 4 can also be formulated using homogeneous coordinates replacing $R, \mathbf{c}, \mathbf{o}$ with a single matrix.

To compute the position of $\mathbf{c}$ at time t it is enough to know the position of two rigid parts A and B, each represented by two reference vertices, at times t and $t+\delta$: $\mathbf{p}_i, \mathbf{p}'_i$, $0 < i \leqslant 4$, where $\mathbf{p}_i$ is the position of a vertex at time t and $\mathbf{p}'_i$ is the position at time $t+\delta$. The vertices of part A are identified by $i \in \{1,2\}$ and of B by $i \in \{3,4\}$. The previous relations produce the following system of equations:

$$\begin{cases} \mathbf{p}'_1 = (R_A * (\mathbf{p}_1 - \mathbf{c}) + \mathbf{c}) + \mathbf{o} \\ \mathbf{p}'_2 = (R_A * (\mathbf{p}_2 - \mathbf{c}) + \mathbf{c}) + \mathbf{o} \\ \mathbf{p}'_3 = (R_B * (\mathbf{p}_3 - \mathbf{c}) + \mathbf{c}) + \mathbf{o} \\ \mathbf{p}'_4 = (R_B * (\mathbf{p}_4 - \mathbf{c}) + \mathbf{c}) + \mathbf{o} \end{cases},$$

where R_A and R_B are the 2D rotation matrices of the parts A respectively B, with angles θ_A, respectively θ_B. Solving the system gives $\mathbf{c}, \mathbf{o}, \sin(\theta_A), \cos(\theta_A), \sin(\theta_B), \cos(\theta_B)$. The position of the articulation point $\mathbf{c}$ is computed in the first frames and used further on as mentioned above (see Section 4.1).

5 Improvements on spring system

In the course of the current approach the spring system of [3] has been improved with respect to the ordering of the regions in the iterative process and the calculation of the gain for mixing the offsets of Mean Shift and structure.

The ordering of the regions during the iterations depends now on the correspondence between the candidate model $\hat{p}$ of the current frame and the target model $\hat{q}$ from the initialization. Both models are 3D color histograms in the RGB color space. They are normalized by the constant C such that

$$\sum_{u=1}^{m} \hat{q}_u = 1, \tag{5}$$

where m is the number of bins. The similarity between the models can be determined by the *Bhattacharyya* coefficient [6]:

$$B = \sum_{u=1}^{m} \sqrt{\hat{p}_u \cdot \hat{q}_u}. \tag{6}$$

With this similarity measurement the regions are ordered descending so that the regions where the candidate model is very similar to the target model are processed first. This ordering, in comparison to the randomized ordering, has the advantages that regions which are represented very well come first, and regions which are occluded are processed at the end.

The second improvement of the spring system is the calculation of the gain. In [3], the gain was set to 0.5 so both Mean Shift and the spring system had the same influence on the resulting positions. This gain is "fair" but not the best choice. It would be more reasonable to calculate the gain separately for each region depending on its candidate model $\hat{p}$ and the resulting Bhattacharyya coefficient B. In the improved version of the spring system the gain is calculated as follows:

$$g = 0.5 - (B - 0.5). \tag{7}$$

Initial structure | Frame 70 | Frame 120 | Frame 180

Figure 4: Improvements on spring system for rigid parts/objects. Top: old approach [3], bottom: improved approach (Section 5).

g weights the offset of the spring system and $1-g$ the offset of Mean Shift.

Experiment: Figure 4 shows an experiment with a real video sequence. It is a challenging sequence because the pattern on the t-shirt is undergoing partly non-rigid motion, meaning it is not only translated and rotated, but also squeezed and expanded (crinkles of t-shirt). As can be seen in the first row of Figure 4, the "old" version of the spring system has problems tracking the target object, because the ordering of the regions and the gain do not adapt to the situation. The second row shows that the improved version of the spring system is more robust.

6 Experiments

The following experiments use synthetic sequences to accurately analyze the behavior of this approach when tracking articulated objects. A advantage of using synthetic sequences is the knowledge of ground truth and the controlled environment. In all sequences the size of the search windows of Mean Shift is equal to the bounding boxes of the regions detected by MSER. The synthetic pattern contains 7 regions and is 50×100 pixels and occlusion is 100×100 pixels. The experiments differ in the parameters used to translate and rotate the patterns and the occlusion (see Table 1).

Experiment 1: In this experiment the patterns are translated 6 pixels in every frame (see Table 1 and Figure 5). Due to this big movement and the full occlusion of the left pattern in frame 8, separately tracking the two patterns fails. Mean Shift is not able to correctly associate the regions after the occlusion, because the search windows are not at good starting positions and too small. Our new approach using the estimated articulation point is able to successfully track the regions through this sequence. The distance constraint imposed by the articulation point is the reason why even though there are big to full occlusions, the positions of the occluded regions can be reconstructed by the spring system without visible features. The graphs in Figure 6 show the spatial deviation of each region over time from ground truth.

Experiment 2: This experiment differs from experiment 1 by a higher rotation between the frames (see Figure 7). The approach without the articulation point has the same problems as in experiment 1. Our approach is able to successfully track and re-assign the regions after occlusion, but this time the spatial deviation is higher (see Figure 8) and the spring system needs more time to minimize its energy. The reason for this is the higher rotation offset between the frames. The articulation point only brings in distance constraints and no motion model or other knowledge about the behavior of the object parts.

Experiment 3: In experiment 3 the rotation offset is 6° per frame and the duration of the occlusion is higher (the translation offset was reduced to assure that Mean Shift is able to track the parts if there is no occlusion). This very high rotation offset leads to the result that our approach with articulation also fails (see Figure 9). The cause is already mentioned in the experiment above: there is only distance and no motion information.

7 Discussion

This section discusses the main parameters of our approach.

Tracker: The choice of Mean Shift for the tracker is based on its iterative nature, and its ability to provide the position of the target object based on a search started from a given position. These two properties fit very well into this approach as the spring system optimization is also iterative, and it is possible to re-initiate Mean Shift at any given state of a vertex in the spring system. Another tracker with the same properties could also be used.

Structure: The current approach extends the rigid structure to handle articulation. This only imposes a distance constraint and does not consider any information related to the motion of the parts. We propose to approximate the relaxation (energy minimization) process on the graph of the whole object with local processes on each of the rigid parts, including the incident articulation points. Instead of keeping the articulation points fixed in the local optimization one could consider giving them a movement liberty inverse proportional to the object parts on the other side. Also the criteria for the selection of the two reference vertices of the parts, connected with the articulation point has to be considered:

Table 1: Parameters of Experiments. The offsets define the amount of translation and rotation in every frame of the synthetic sequences. Translation offsets are in pixels and angle offsets are in degree.

experiment	x-offset	ϕ-offset	occlusion x-offset	occlusion y-offset
1	6	2	20	20
2	6	4	20	20
3	4	6	10	10

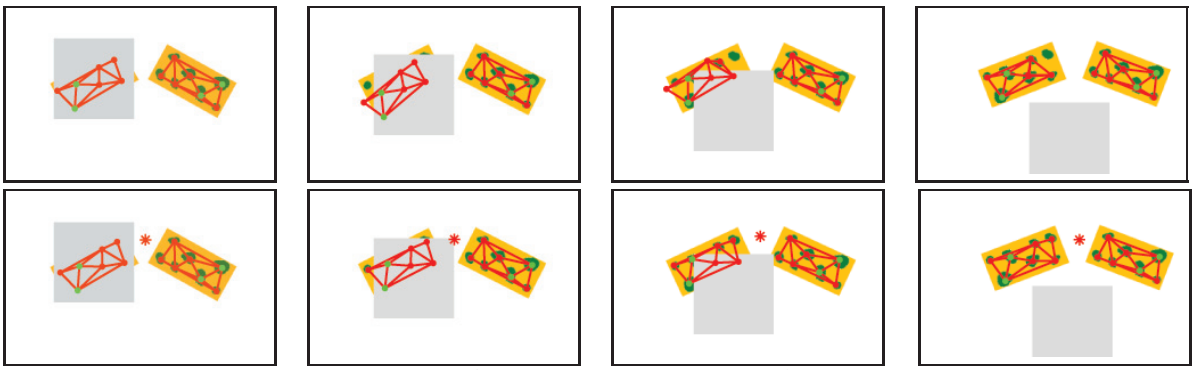
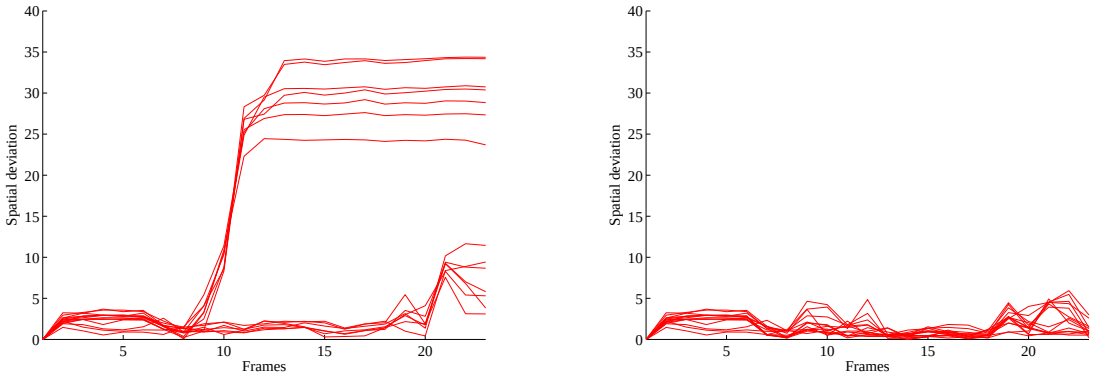


Figure 5: Articulation experiment 1. Top row: frames without articulation point, bottom row: with articulation point. The green vertices in the graphs represent the reference vertices.



(a) (b)

Figure 6: Spatial deviation in pixels without (a) and with (b) articulation point for experiment 1.

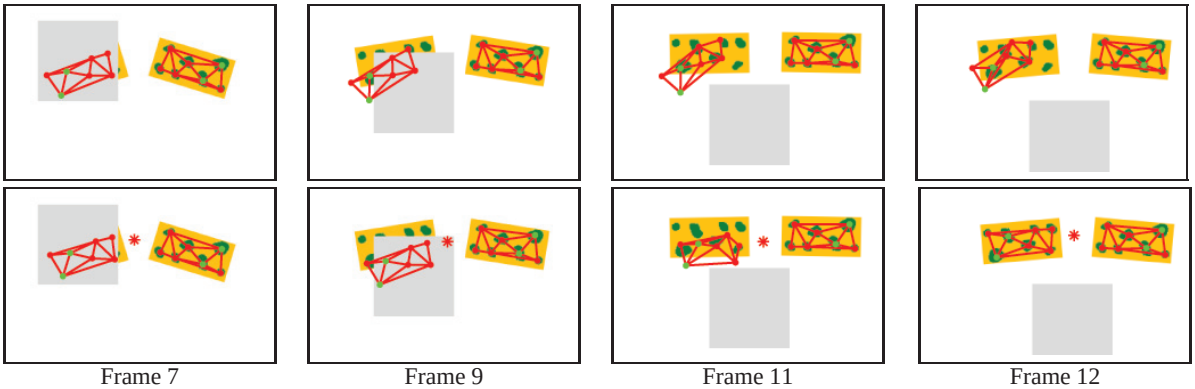
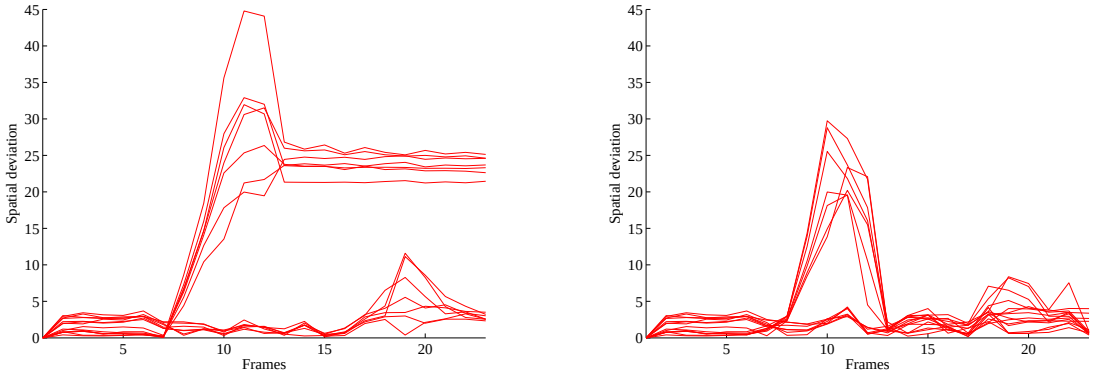
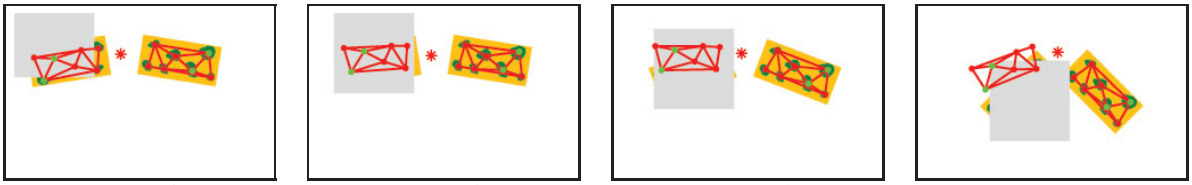


Figure 7: Articulation experiment 2. Top row: frames without articulation point, bottom row: with articulation point.



(a) (b)

Figure 8: Spatial deviation in pixels for experiment 2 without (a) and with (b) articulation point.



Frame 6 Frame 8 Frame 10 Frame 14

Figure 9: Articulation experiment 3, with articulation point.

e.g. taking vertices with highest connectivity and best visual support. Depending on the structure of the parts using only two vertices might not be enough to quickly propagate the information from the articulation point to every vertex of the part.

As shown in figure 9 the structure of the objects (structure in motion) without any information about its behavior (structure of motion) is not enough to solve complicated cases with whole parts being occluded. One can consider adding motion models to the parts, or consider additional information, to handle those cases.

Context: During an occlusion it is not possible to know for sure how the occluded object part is behaving. It could continue its movement, increase or decrease its speed or even stop the movement completely. Instead of estimating the positions of the invisible features we plan to use higher level knowledge like spatio-temporal continuity to observe the occluded part reappearing around the borders of the visible occluding object.

8 Conclusion

This paper presents a structural approach for tracking articulated objects. The spatial relationships between the parts of the object are encoded into a spring system. The position of the articulation points is derived by observing the articulated object. Integrating the articulation point into the optimization process of the spring system leads to improved tracking results in videos with big transformations and occlusions. In the future we plan to make our approach invariant to scaling and perspective changes of the target object and to cope with disappearing, appearing and reappearing features. The purpose is to add more knowledge about the environment of the object and its relations.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, march 1999.

[2] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Articulated and elastic non-rigid motion: A review. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 2–14, 1994.

[3] N. Artner, S. B. López Mármol, C. Beleznai, and W. G. Kropatsch. Kernel-based tracking using spatial structure. In *32nd Workshop of the Austrian Association for Pattern Recognition*, pages 103–114. Austrian Computer Society, May 2008.

[4] Norman I. Badler and Stephen W. Smoliar. Digital representations of human movement. *ACM Comput. Surv.*, 11(1):19–38, 1979.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.

[7] Pedro F. Felzenszwalb. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2005.

[8] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *Transactions on Computers*, 22:67–92, January 1973.

[9] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image*

Understanding, 73(1):82–980, January 1999.

[10] Salvador B. López Mármol, Nicole M. Artner, Adrian Ion, Walter G. Kropatsch, and Csaba Beleznai. Video object segmentation using graphs. In *13th Iberoamerican Congress on Pattern Recognition, CIARP 2008*, pages 733 –740. Springer, September 2008.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, September 2004.

[12] Thomas Mauthner, Michael Donoser, and Horst Bischof. Robust tracking of spatial related components. In *International Conference of Pattern Recognition*. IEEE, December 2008.

[13] Thomas B. Moeslund, Adrian Hilton, and Volker Krger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2–3):90–126, 2006.

[14] D. Ramanan and D.A. Forsyth. Finding and tracking people from the bottom up. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2:II–467–II–474 vol.2, June 2003.