Multi-Scale 2D Tracking of Articulated Objects Using Hierarchical Spring Systems

Nicole M. Artner[a,c], Adrian Ion[b,c], Walter G. Kropatsch[c]

[a] *AIT Austrian Institute of Technology, Vienna, Austria*
[b] *Institute for Numerical Simulation, University of Bonn, Germany*
[c] *PRIP, Vienna University of Technology, Austria*

Abstract

This paper presents a flexible framework to build a target-specific, part-based representation for arbitrary articulated or rigid objects. The aim is to successfully track the target object in 2D, through multiple scales and occlusions. This is realized by employing a hierarchical, iterative optimization process on the proposed representation of structure and appearance. Therefore, each rigid part of an object is described by a Hierarchical Spring System represented by an attributed graph pyramid. Hierarchical Spring Systems encode the spatial relationships of the features (attributes of the graph pyramid) describing the parts and enforce them by spring-like behavior during tracking. Articulation points connecting the parts of the object allow to transfer position information from reliable to ambiguous parts. Tracking is done in an iterative process by combining the hypotheses of simple trackers with the hypotheses extracted from the Hierarchical Spring Systems.

Keywords: 2D Tracking, Articulated Objects, Hierarchical Spring System

Email addresses: nicole.artner@prip.tuwien.ac.at (Nicole M. Artner), ion@ins.uni-bonn.de (Adrian Ion), krw@prip.tuwien.ac.at (Walter G. Kropatsch)

1. Introduction

The task of monocular tracking of articulated objects is a challenging one. Complex articulations can significantly change the appearance of the object and distant parts can perform very different motions. These aspects affect popular trackers [1] that consider the appearance of simple shapes (e.g. rectangles), as certain poses might not be very compact and cover only a small portion of the bounding box, and trackers that assume a simple global motion model for the whole part.

The most promising approaches of articulated tracking are quite complex and depend to a large extent on strong motion and subject specific priors. While they do deliver excellent results for the object class they have been designed for (e.g. humans), most of them do not generalize very well and would need extensive adaptation to work for other object classes. Recent examples of such well performing specialized methods are Lee and Elgammal [2], who introduce a model that ties together the human body configuration manifold and visual manifold in one representation, which is then used for tracking within a Bayesian framework, and Brubaker et al. [3] who present a physics-based model with a bio-mechanical characterization of lower-body dynamics, where tracking is accomplished with a form of sequential Monte Carlo inference.

In contrast, the presented approach requires only basic information on the structure of the target object and no motion prior, which makes it less object-class specific and more general. Objects are represented as features in arbitrary configurations. Tracking a whole object builds on simple, single hypothesis feature trackers, and deals with partial occlusion, scaling, and

limited non-rigid deformation. The output consists of the 2D positions and bounding box of the object parts in every frame of the video.

At the heart of the method is a representation which describes the appearance and kinematics of articulated objects. It consists of multiple object parts modeled by rectangular regions of interest and features extracted out of these regions. Kinematics are realized by connecting object parts through articulation points, which limit the movement of each part to a circle (see Fig. 3).

Multiple feature trackers, called *sub-trackers*, are used for each part: one attempting to track the whole part and the rest considering small fixed-size windows centered around detected interest points (see Fig. 1).

To deal with *occlusion* and avoid drifting of the sub-trackers we *model the parts* as a graph hierarchy with two levels: one top-level vertex for the sub-tracker tracking the whole part and multiple bottom-level vertices for the interest-point sub-trackers. The edges of the graph are weighted with the pairwise distances between the features, and act like springs pushing and pulling the vertices to reduce the deformation of the graph-structure of the parts, thus giving the name *Hierarchical Spring System* (HSS).

The final position of each feature (top and bottom level) is obtained through a mediation between the corresponding tracker, pulling towards what it considers to be the target region, and the HSS trying to enforce the initial structure (reduce deformation). The weight of each of these two factors is dynamically adjusted depending on the similarity of the region at the current position with the known appearance of the part. Thus, during occlusion (by a different looking object) the HSS has more weight allowing for badly

tracked features to be placed at known relative positions, while at times of successful tracking the very confident sub-trackers are given more weight, allowing for a certain amount of non-rigid deformation. A *global scaling* factor is maintained and used to adjust the "relaxed" (no deformation) lengths of the springs, allowing to cope with global changes in scale.

Articulated objects are modeled as multiple HSS corresponding to each part connected by vertices representing the articulation points. Articulation points have no corresponding sub-trackers and move solely under the "forces" of the adjacent parts. Thus movement of one adjacent part is transmitted to the other enforcing articulated motion.

All computation (position of sub-trackers, scaling, and articulation) is done using local confidence measures to balance between trusting the sub-trackers i.e. the visual feedback, and the object structure i.e. the prior knowledge.

1.1. Related work

First introduced by Fischler et al. in 1973 [4], pictorial structures represent an object by its parts (e.g. head, torso, arms, legs) arranged in a deformable spatial configuration. This deformable configuration is represented by spring-like connections between pairs of parts. Object recognition or tracking can be done by minimizing the energy in this deformable configuration to find the most likely configuration of the object parts in an image. Felzenszwalb et al. employed this idea in [5] 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. Ramanan et al. apply in [6] the ideas

from [5] in tracking people.

Besides Computer Vision, the proposed representation is also related to representations used in Computer Graphics called *mass-spring systems* [7]. Mass-spring systems are a physically based technique that is used to effectively model deformable objects for animations in Computer Graphics (e.g. a flag moving in the wind). An object is modeled by a collection of point masses connected by springs in a lattice structure.

Different from the mentioned approaches, we stress solutions that emerge from the underlying structure, as opposed to using structure to verify sampled hypothesis. The proposed representation not only connects parts in a deformable way like in [5], but introduces a bottom level consisting of "small" region descriptors described by a Spring System. In comparison to Pictorial structures the presented approach does not need training, because the spring-like behavior is modeled via a combination of structural and appearance offsets (provided by the sub-trackers).

Even though the bottom level of the proposed hierarchical Spring System is similar to a mass-spring system [7], there are significant differences. The presented Spring System is used to supply structural feedback for tracking algorithms, which is a totally different purpose and it does not consider any external forces (e.g. gravity). In the proposed approach a vertex does not have a mass, but the force of the spring is calculated by its confidence in the current frame.

1.2. Contributions

Our main contribution is the flexible framework for representing and tracking articulated objects of arbitrary complexity with each (rigid) part

of an object represented by a Hierarchical Spring System (HSS), connected to other parts by articulation points. Articulation points are used to transfer information between the HSS of the adjacent object parts. All decisions balance between "seeing" and "knowing" using maintained confidence measures. We pose tracking as a hierarchical optimizations process on structure and appearance.

A preliminary version of our approach has been presented in [8]. Possible applications are action recognition, human computer interfaces, motion based diagnosis and identification, etc.

1.3. Overview

This paper is organized as follows: Section 2 describes how to represent the appearance and structure of a rigid object in a HSS. It is explained how our approach combines the hypotheses of the sub-trackers and the HSS. In Section 3 the introduced concepts of Section 2 are used to model articulated objects consisting of several rigid object parts. Additionally, articulation points and the information transfer between the object parts are explained. Section 4 presents the algorithm of the tracking with the help of pseudo code. In Section 5 experiments qualitatively and quantitatively analyze the results of the presented approach. Section 6 gives a conclusion, and the Appendix introduces the employed region descriptor (Sigma Sets).

2. Representation and tracking of a rigid object

Background clutter, similar objects in the scene and occlusions are the main reasons for tracking failure, because they can be good matches to the model of the target object and thus distract the tracker.

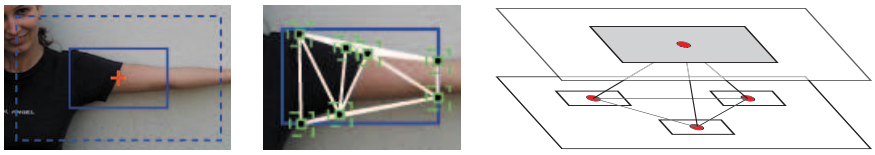
If the appearance of an object is uniform (no texture, mainly one color), it is advisable to describe and track it by one feature (e.g region descriptor). Tracking whole rigid objects or parts can deliver robust positions even during motion blur due to the large image region considered. Nevertheless, in cases of partial occlusion or scaling such a description is not able to aid the tracker in overcoming the difficult distractions by providing useful information.

On the other hand, if the target object is textured (e.g. face of a human), it is possible to extract several discriminative features out of the region covering the object and track them successfully when there are no distractions. By additionally encoding the spatial relationships of the features in the representation of the object, it is possible to deal with occlusions and estimate scaling. Unfortunately, these "small" features are more sensitive to noise and fast motion of the object (big distances between frames, motion blur).

As we cannot generally decide which representation is more suitable for an object and to get the best of both worlds, we describe and track objects using multiple features and sub-trackers, where the spatial relationships of the features are described and enforced by a *Hierarchical Spring System* (HSS).

2.1. The sub-tracker

The purpose of each sub-tracker is to attempt to track a fixed-size region independently of the other sub-trackers, based solely on the content of the image. At any frame, given as input an initial estimate of the position of a tracked region, the corresponding sub-tracker will return an offset to what it considers to be the correct position of the target region.



(a) (b) (c)

Figure 1: Example representation for a part (a) Feature for the top level sub-tracker. (b) Features for the bottom level sub-trackers. The white edges are the edges of $\mathbf{G_0}$. (c) Corresponding graph pyramid $\mathbf{P} = \{\mathbf{G_0}, \mathbf{G_1}\}$ (not all bottom level vertices and edges are shown).

2.2. The Hierarchical Spring System (HSS)

We represent the HSS of an object as a graph pyramid with two levels $\mathbf{P} = \{\mathbf{G_0}, \mathbf{G_1}\}$, where the top level $\mathbf{G_1}(\mathbf{V_1}, \mathbf{E_1})$ contains one single vertex $\mathbf{V_1} = \{\mathbf{v_p}\}$, and the bottom level graph $\mathbf{G_0}(\mathbf{V_0}, \mathbf{E_0})$ multiple vertices connected by edges. There is an one-to-one mapping between the vertices in the graph pyramid and the features with their corresponding sub-trackers. Edges are weighted with the known distance in the image plane between the features corresponding to the incident vertices. The vertex in the top level is connected with all vertices in the base level to allow communication between the two levels. Figure 1 shows an example representation for an object and the corresponding regions for the sub-trackers. (Options for inserting the edges are discussed in Section 5.3.1).

2.3. Tracking with sub-trackers and HSS

For each frame the first hypotheses of the sub-trackers are refined using an iterative alternation and combination of the offsets from the sub-trackers and the offsets from the HSS.

2.3.1. Energies in the HSS

The HSS encodes the spatial relationships of the features of the object considering their spatial distances and arrangement. Its task is to keep the structure of the features as similar as possible to the initial state in the first frame. This is realized by providing the tracker with *structural offsets* (see Sec. 2.3.4).

To calculate a structural offset for a feature it is necessary to determine the extent of the spatial deformation in the HSS. The extent of the deformation in a vertex v at time $i = 1 \dots n$ is represented and calculated by the energy ε in v:

$$\varepsilon^i(v) = \sum_{e \in \mathbf{E^{i-1}(v)}} \delta^i(v_e) \cdot (|e| - |e^1| \cdot x)^2, \tag{1}$$

where $\mathbf{E^{i-1}(v)}$ are all edges e of the levels $\mathbf{E_0}$ and $\mathbf{E_1}$ at time i incident to vertex v. $\delta^i(v_e)$ is the *confidence* (see Sec. 2.3.2) of the neighboring vertex v_e at time i connected by e, which weights the influence of v_e on $\varepsilon^i(v)$. The motivation behind the weighting with $\delta^i(v_e)$ is that occluded neighboring vertices should have a lower impact on $\varepsilon^i(v)$ than reliably tracked neighbors. $|e|$ and $|e^1|$ denote the deformed and initial edge lengths between v and v_e, and x is the current scaling factor of the object. x is used to apply a global scaling to the initial edge lengths $|e^1|$ to be able to track an object changing its distance to the camera (see Sec. 2.3.3).

2.3.2. The confidence of a vertex

The confidence is used to dynamically weight influences of vertices in different calculations and situations e.g. calculation of $\varepsilon^i(v)$ (see Sec. 2.3.1).

The confidence $\delta^i(v)$ of a vertex v at time i depends on its degree I_v (number of incident edges), its energy $\varepsilon^{i-1}(v)$ and the dissimilarity $D^{i-1}(v)$ between its feature $S^{i-1}(v)$ at time $i-1$ to its descriptor $S^1(v)$ in the initial iteration:

$$\delta^i(v) = \frac{\widetilde{I(v)} + \widetilde{\varepsilon^{i-1}}(v) + \widetilde{D^{i-1}}(v)}{3} \tag{2}$$

$\widetilde{I(v)}$, $\widetilde{\varepsilon^{i-1}}(v)$ and $\widetilde{D^{i-1}}(v)$ are normalized so that $0 \leq \delta^i(v) \leq 1$.

$$\widetilde{I(v)} = \frac{\mathbf{E}(\mathbf{v})}{\mathbf{E}} \tag{3}$$

where $\mathbf{E}(\mathbf{v})$ are the edges incident to vertex v and $\mathbf{E}$ are all edges in the HSS.

$$\widetilde{\varepsilon^{i-1}}(v) = \begin{cases} 1 - \frac{\varepsilon^{i-1}(v)}{\varepsilon^{i-1}_{\max}} & \varepsilon^{i-1} \leq \mathrm{s}^{i-1}_{\varepsilon} \\ 0 & \varepsilon^{i-1} > \mathrm{s}^{i-1}_{\varepsilon} \end{cases} \tag{4}$$

where $\varepsilon^{i-1}(v)$ is the energy in vertex v in iteration $i-1$ (see Equation 1), $\mathrm{s}^{i-1}_{\varepsilon}$ is the standard deviation of the energies in the local neighborhood (vertex v and its connected neighboring vertices), and ε^{i-1}_{max} is the maximum energy smaller or equal to $\mathrm{s}^{i-1}_{\varepsilon}$. The standard deviation $\mathrm{s}^{i-1}_{\varepsilon}$ is considered to penalize outliers and to normalize with a suitable ε^{i-1}_{max}.

$$\widetilde{D^{i-1}}(v) = \begin{cases} 1 - \frac{h(S^{i-1}(v), S^1(v))}{h^{i-1}_{\max}} & h(S^{i-1}(v), S^1(v)) \leq \mathrm{s}^{i-1}_{D} \\ 0 & h(S^{i-1}(v), S^1(v)) > \mathrm{s}^{i-1}_{D} \end{cases} \tag{5}$$

where $h(S^{i-1}(v), S^1(v))$ is the distance between the feature $S^{i-1}(v)$ in the iteration $i-1$ and $S^1(v)$ in the initial iteration. s^{i-1}_{D} is the standard deviation in the local neighborhood (vertex v and its connected neighboring vertices) and $h^{i-1}_{\max}$ is the highest distance value in the neighborhood of v, where $h^{i-1}_{\max} \leq$

s_D^{i-1}. As with $\varepsilon^{i-1}(v)$ the idea behind considering the standard deviation is to successfully deal with outliers and employ a suitable normalization factor $h_{\max}^{i-1}$.

2.3.3. Estimation of the scaling factor

To make the representation invariant to scaling, a scaling factor x^* is estimated once in each frame after the sub-trackers have provided their first hypotheses for the positions of the features.

$$x^*(v) = \sum_{e \in \mathbf{E^{i-1}(v)}} \frac{|e|}{|e^1|} \cdot \frac{\delta^i(v_e)}{\sum\limits_{v_e \in \mathbf{N(v)}} \delta^i(v_e)} \tag{6}$$

where $x^*(v)$ is the estimated scaling factor in the local neighborhood of vertex v. $\mathbf{N(v)}$ is the neighborhood of v (all vertices v_e connected to v by e). $\delta^i(v_e)$ is the confidence of the neighboring vertices in the current iteration. $x^*(v)$ is determined by a weighted sum to boost the influence of the most reliable vertices and the associated edges.

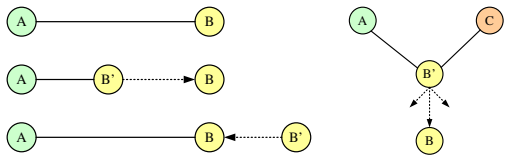
The scaling factor $x^*(v)$ of each vertex is used to calculate a scaling factor for the rigid object (part of an articulated object):

$$x^*(p) = \sum_{v \in \mathbf{V_0}} x^*(v) \cdot \frac{\delta^f(v)}{\sum\limits_{v \in \mathbf{V_0}} \delta^f(v)} \tag{7}$$

where $\mathbf{V_0}$ are all vertices v of the bottom level of the HSS.

2.3.4. Offsets of the HSS

To compute the offsets of the HSS we employ *graph relaxation*, which models the spring-like behavior of the edges with the purpose to minimize the energies in the HSS, i.e. to bring all edges $\mathbf{E}$ to have the same length ratio as in the model (e.g. initial frame).



(a) (b)

Figure 2: Graph relaxation examples. B is the initial state of the vertex and B' the deformed one. The arrows visualize the structural offset vectors $O(B')$.

A *structural offset vector* $\vec{O}(v)$ for vertex v is calculated so that it is pointing to a spatial position in which the $\varepsilon^i(v)$ is minimized:

$$\vec{O}(v) = \sum_{e \in \mathbf{E^{i-1}(v)}} \delta^i(v_e) \cdot (|e| - |e^1| \cdot x)^2 \cdot (-1) \cdot \vec{d}(e, v), \tag{8}$$

where $\vec{d}(e, v)$ is the unitary vector pointing from a neighboring vertex v_e toward v. Figure 2 shows the concept of producing structural offsets with graph relaxation.

2.3.5. Combining the hypotheses

For each feature (vertex) and in each iteration i the corresponding sub-tracker and HSS propose a "new" position with the knowledge of the position of the previous iteration $i-1$ and their offsets.

Both hypotheses are combined to determine the position $\mathbf{c}_{pos}$ of each vertex as follows:

$$\mathbf{c}_{pos} = \delta^i(v) \cdot \mathbf{t}_{pos} + (1 - \delta^i(v)) \cdot \mathbf{s}_{pos}, \tag{9}$$

where $\delta^i(v)$ is the confidence of vertex v at time i, $\mathbf{t}_{pos}$ is a vector representing the hypothesis of the sub-tracker and $\mathbf{s}_{pos}$ is the proposed position of v of the HSS.

3. Assembling parts to form articulated objects

Articulated objects are modeled as multiple object parts represented by *Hierarchical Spring Systems* (HSSs) and connected by vertices representing articulation points. To exchange information between the parts of the object, articulation points are connected to the corresponding HSSs. Articulation points have no corresponding sub-trackers and move solely under the "forces" of the adjacent parts.

3.1. The confidence of a part

The confidence of object parts $\delta^i(p)$ becomes meaningful when the target object is an articulated object consisting of several parts connected by articulation points. It is computed out of the size $I(p)$, the energy $E^{i-1}(p)$, and the dissimilarity $D^{i-1}(p)$ of the feature $S^{i-1}(p)$ in comparison to $S^1(p)$ of the initial frame.

$$\delta^i(p) = \widetilde{I(p)} + \widetilde{E^{i-1}}(p) + \widetilde{D^{i-1}}(p) \tag{10}$$

$\widetilde{I(p)}$, $\widetilde{E^{i-1}}(p)$ and $\widetilde{D^{i-1}}(p)$ are normalized to satisfy $0 \leq \delta^i(p) \leq 1$.

$$\widetilde{I(p)} = \frac{F(p)}{F} \tag{11}$$

where $F(p)$ is the number of features of part p, F is the number of all features in the object.

The sum of all local energies in object part is normalized by the number of features (vertices) in part p:

$$\widetilde{E^{i-1}}(p) = \frac{\sum\limits_{v \in p} E^{i-1}(v)}{F(p)}. \tag{12}$$

$$\widetilde{D^{i-1}}(p) = \begin{cases} 1 - \frac{h(S^{i-1}(p), S^1(p))}{h^{i-1}_{\max}} & h(S^{i-1}(p), S^1(p)) \leq \mathrm{s}^{i-1}_D \\ 0 & h(S^{i-1}(p), S^1(p)) > \mathrm{s}^{i-1}_D \end{cases} \tag{13}$$

where $h(S^{i-1}(p), S^1(p))$ is the distance between the feature $S^{i-1}(p)$ in the current iteration and $S^1(p)$ in the initial frame. s^{i-1}_D is the standard deviation of the distances for all parts in the target object and $h^{i-1}_{\max}$ is the highest distance value, where $h^{i-1}_{\max} \leq \mathrm{s}^{i-1}_D$.

3.2. Scaling of the whole object

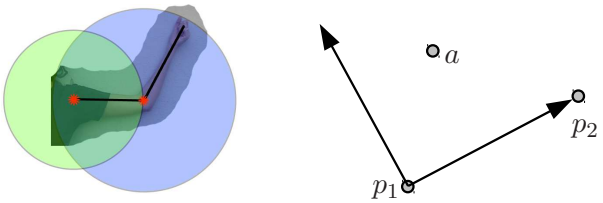
The estimation of the gobal scaling of the whole articulated object is based on the scaling factors of the object parts $x^*(p)$ (see Sec. 2.3.3), which are combined by a weighted sum:

$$x^*(O) = \sum_{p \in O} x^*(p) \cdot \frac{(\delta^f(p))}{\sum\limits_{p \in O} (\delta^f(p))}. \tag{14}$$

3.3. Articulation points: agents of the information transfer

An *articulation point* connects several rigid parts. It allows them to move independently from each other, while keeping the same distance to it. The movement of a point of a rigid part in the image plane is constrained to a circle centered at the articulation point. The radius is equal to the distance between the point of the rigid part and the articulation point. Figure 3 illustrates this concept.

If the articulation point moves it "pulls" the connected rigid part to keep the distance constrain, and vice versa. In this way position information is transfered from one rigid part to an adjacent one over the articulation point.



articulated motion local coordinate system

Figure 3: Left: distance constraints imposed by articulation points. Right: articulation point a in the local coordinate system defined by an ordered pair of points p_1, p_2.

3.3.1. Modeling articulation points

Planar articulated motion from frame f to frame $f + \delta$ can be decomposed into: an independent rotation of the rigid parts around the articulation point, followed by a common translation of the parts (and the articulation point). Given two pairs of points corresponding to two rigid parts performing articulated motion, each at frame f and $f + \delta$, the rotation $(cos(\theta), sin(\theta))$ of each part, the common translation (O_x, O_y) as well as the position of the articulated point at frame f are obtained by solving the resulting system of eight equations with eight unknowns.

During the initialization of the representation a local coordinate system of each pair of features of an object part is created (see Fig. 3). The coordinates of the articulation point in this coordinate systems are stored. Having the position of any two features is then enough to define the coordinate system and reconstruct the position of the articulation point in every frame.

3.3.2. Tracking articulation points

At any time during tracking, knowing the positions of two vertices of a part and the current scaling factor is sufficient to generate a hypothesis for the positions of all adjacent articulation points. These hypotheses are produced

with the local coordinate system defined by the two most confident features (see Sec. 2.3.2) – further on named *reference vertices* – of each part.

The hypotheses of all parts connected to an articulation point are combined with a weighted sum to calculate the current position $\mathbf{a}_{pos}$ of the articulation point a. The weight for each hypothesis depends on the confidence of the corresponding part (see Sec. 3.1).

$$\mathbf{a}_{pos} = \sum_{p \in \mathbf{P}(\mathbf{a})} y_p \cdot \frac{\delta^i(p)}{\sum\limits_{p \in \mathbf{P}(\mathbf{a})} \delta^i(p)}, \tag{15}$$

where $\mathbf{P}(\mathbf{a})$ is the set of parts connected to the articulation point a. y_p is the hypothesis determined with the local coordinate system (which considers the current scaling factor x) of part p. $\delta^i(p)$ is the confidence of part p. With this weighted sum, the influence of ambiguous parts on the position of the articulation point is low (e.g. if a part is occluded) and of reliably tracked parts high.

3.4. Information transfer

For each rigid part, the distance constraint to the articulation point is enforced by connecting all vertices from the bottom level and the vertex from the top level with the corresponding articulation point. The articulation point "transfers" position information from reliably to ambiguously tracked parts through its distance constraints (circles).

The information transfer is realized with graph relaxation by calculating a structural offset vector. Therefore, Equ. 8 is adapted as follows:

$$\vec{O}(v) = \delta^i(v) \cdot (|e| - |e^1| \cdot x)^2 \cdot (-1) \cdot \vec{d}(e, v), \tag{16}$$

where $\delta^i(v)$ is the confidence of vertex v, $|e|$ is the length of edge e connecting v with a and $|e^1|$ represents the length of the same edge in the initial frame. $\vec{d}(e, v)$ is the unitary vector pointing from a vertex v toward the articulation point a.

4. Tracking as a hierarchical optimization process - the algorithm

The algorithm to track articulated objects using HSSs is summarized in Algorithm 1.

Tracking is done in a *top to bottom* or *bottom to top* process, depending on the confidence values (see Alg. 1, Line 8). In frames when the tracking is reliable, the springs connecting the top vertex with the bottom level are used to generate additional structural offsets for the vertices in the bottom level (*top to bottom* processing). During occlusions this flow of structural feedback is inversed s.t. structural offsets are determined for the top vertex (*bottom to top* processing). The decision for *top to bottom* or *bottom to top* processing is taken by a comparison of the confidence values of the top and bottom vertices. In cases of ambiguity *bottom to top* processing is preferred (confidence value of top vertex is smaller than confidence of bottom vertex).

5. Experiments

The following experiments show the application of the presented framework on concrete tracking tasks with different complexities and difficulties.

5.1. The sub-trackers

We use the Mean shift algorithm for the sub-trackers. It is a simple, single hypothesis tracker, which on its own is not able to track complex, articulated

Algorithm 1 Algorithm for tracking articulated objects.

```
1: processFrame
       T_i threshold maximum number of iterations
2:     i ← 1                                                  ▷ iteration counter
3:     while (i < T_i) do
4:         for every rigid part do
5:             calculate confidences δ^i(v) and δ^i(p)
6:             estimate positions with sub-trackers top and bottom
7:             if i > 1 then
8:                 decide between top to bottom or bottom to top processing
9:                 do structural iteration top and bottom
10:            end if
11:            mix hypotheses for positions depending on δ^i(v)
12:            update energies in HSS
13:            if i == 1 then
14:                estimate scaling factor
15:            end if
16:        end for
17:        for every rigid part do
18:            update δ^i(v) and δ^i(p)
19:        end for
20:        calculate current position of articulation point
21:        for every rigid part do
22:            information transfer
23:            update energies in HSS
24:        end for
25:        i ← i + 1
26:    end while
27: end
```

objects successfully.

Mean shift efficiently searches for local extremal values in a probability distribution with a search window, and generates an offset vector pointing to the corresponding position. The value of the distribution at a certain point depends on the similarity between features extracted within a window centered at that point and features extracted in an initialization phase from the region to be tracked.

5.2. The region descriptors

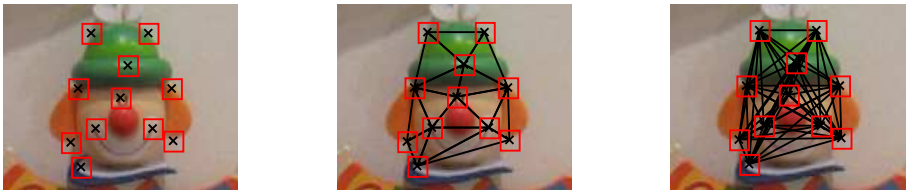
Sigma Sets are used in the experiments as the region descriptors (features) describing the appearance of the corresponding regions of interests covering the target object. Appendix A gives a brief recall of Sigma Sets.

The extraction of the features in every frame is very expensive with regard to computation time. In a frame with a resolution of 480×640 pixels the calculation of the features compensates between 60 to 70 seconds of the overall computing time of maximum 75 seconds per frame.

5.3. Initializing the Hierarchical Spring Systems

Features/Vertices. Before a HSS can be built, a target object needs to be defined and suitable features describing the object have to be selected. This can be done automatically by methods like in [9, 10, 11, 12] or semi-manually as for the experiments in this paper.

The top level is described by one region descriptor $S^1(p)$, extracted out of a region of interest (ROI) covering the whole object part (Fig. 1(a)). The bottom level consists of several smaller region descriptors, which are from the same ROI (see Fig. 1(b)). A Harris corner detector is applied on the ROI



(a) (b) (c)

Figure 4: Building a HSS. Target object: head of jumping jack. (a) Selected features: region descriptors (red boxes). (b) Inserted edges: triangulated graph. (c) Inserted edges: fully connected graph.

to find promising positions for the smaller region descriptors $S^1(v)$. Around each corner point a small ROI is built to extract a Sigma Set (e.g. 9×9 pixels).

Edges. The edges can be inserted with a Delaunay triangulation (see Figure 4(b)) or a fully connected graph can be built (see Figure 4(c)). For more details on inserting the edges refer to Section 5.3.1.

Articulation points. They can be initialized manually (as in the following experiments) or automatically by observing the articulated motion of the target object [13, 14].

5.3.1. Connectivity issues

This section deals with the impact of the connectivity of the vertices in the HSS on the quality of the structural feedback i.e. on the structural offset vector.

Given the features represented as vertices, there are different possibilities for adding the edges connecting them e.g.: a Delaunay triangulation or a fully connected graph (see Figure 4).

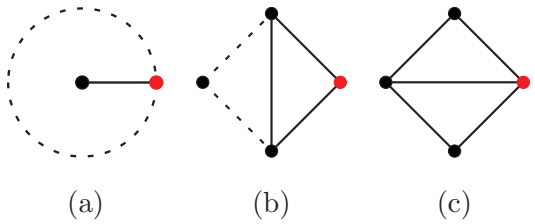


Figure 5: Ambiguity of structural offset vectors. (a) Vertex degree 1, all positions on circle are minima. (b) Vertex degree 2, two minima. (c) Vertex degree 3, one unique minimum.

If a vertex v is of degree 1 – only connected to one neighbor – the structural feedback determined by graph relaxation is ambiguous. The local energy $\varepsilon^i(v)$ in the current vertex v is minimized ($\varepsilon^i(v) = 0.0$) by moving v to any point on the circle centered on its neighbor with the radius equal to the "original" length $|e^1|$ of the edge connecting them. Therefore, there is no unique global minimum or structural offset vector for v. For a vertex v with degree 2, the ambiguity is reduced to two possible positions, both with $\varepsilon^i(v) = 0.0$. Above degree 2, there is only one position in the image, which minimizes $\varepsilon^i(v)$. Figure 5 visualizes these three cases.

In our experiments both a Delaunay triangulation and a fully connected graph are used as representation. Table 1 lists important facts of both representations.

As Tab. 1 lists, a fully connected graph may produce superior results. When determining the structural offset vector (see Equation 8) each vertex gets structural input from every other vertex in the graph. Especially in cases of occlusion, this leads to a faster propagation of "correct" position information (see Figure 6 in Section 5). The only drawback we identified for the fully connected graph is the, in our experiments insignificant, increase in processing time when calculating the structural offset vector.

Representation	Connectivity	Quality of struc. feedback	Propagation of information
Triangulation	low, some vertices have only degree 2	robust without occlusion, can be ambiguous in cases of occlusion	slow for graphs with many vertices
Fully connected	high, all vertices of degree 3 or higher	robust with and without occlusions	fast, independent on the number of vertices

Table 1: Comparison of facts of a triangulated and a fully connected graph.

5.4. Experimental setup

The videos employed for the following experiments are self-produced (800×600 pixel), from the Motion of Body (MoBo) database [15] (486×640 pixel) and from Amit et al. [16] (352×288 pixel).

The videos are selected considering the current status of the presented approach. Even though the proposed framework is able to successfully track objects through articulated motion and scaling, it can only deal with affine or perspective changes up to a certain degree. The reason for this lies in the current state of the HSS as it does not consider the 3D space when generating structural offset vectors. Therefore, videos with objects moving in the 3D space are not suitable for our experiments and will lead to significant errors in tracking.

In all experiments presented in this section, the target object is initialized manually by selecting the parts of the object and defining the positions of the articulation points. Except of the video in experiment 1, the ground truth was determined by us and is a result of manually selecting the center

positions of the object parts.

The results presented in this section (images and graphs) are best viewed in color.

5.5. Experiment 1: Occlusion

This experiment focuses on occlusions and compares the tracking results of Mean shift alone and our combined approach. The video used in this experiment is from the work of Amit et al. [16]. It shows the face of a woman being partially occluded several times.

In Figure 6 one can see the results of tracking with Mean Shift alone, with a HSS with triangulated graphs and with a HSS using fully connected graphs. As already mentioned in Section 5.3.1, the fully connected graph is superior to the triangulated graph in challenging cases of occlusion, which occur in this video sequence. The face is occluded several times by a highly-textured object (magazine) moving in different directions and occluding different parts of the face. This leads to big confusions and errors in the tracking with Mean Shift alone (see Fig. 6 (top)).

Figure 7 shows the quantitative result of this experiment. This results confirm the qualitative results. The ground truth is provided by [16]. When comparing the results of Figure 7 with the results in [16], one can see that the methods have a similar error rate. The approach of Amit et al. [16] has problems in frames 500 to 600, where as our approach performed better in this period .Both methods are challenged in frames 700 to 800, but this time the method of Amit et al. is slightly better.

Figure 6: Experiment 1: Tracking an occluded face with Mean Shift (top), with our approach in a triangulation (middle) and our approach with a fully connected graph (bottom). The images show the features of the bottom level connected by edges to illustrate the deformations and the qualitative results.

5.6. Experiment 2: Articulated motion with self-occlusion

This experiment uses a video of [15] of subject 04011 in view vr16_7. The challenges are self-occlusions and similar appearance in several object parts. (We do not show images of subject 04011 as it is not allowed by [15].)

Figure 8 shows that the presented representation significantly improves the quality of the results of tracking with Mean Shift. The left lower arm is the most challenging object part to track, but our approach is able to recover well from wrong hypotheses.

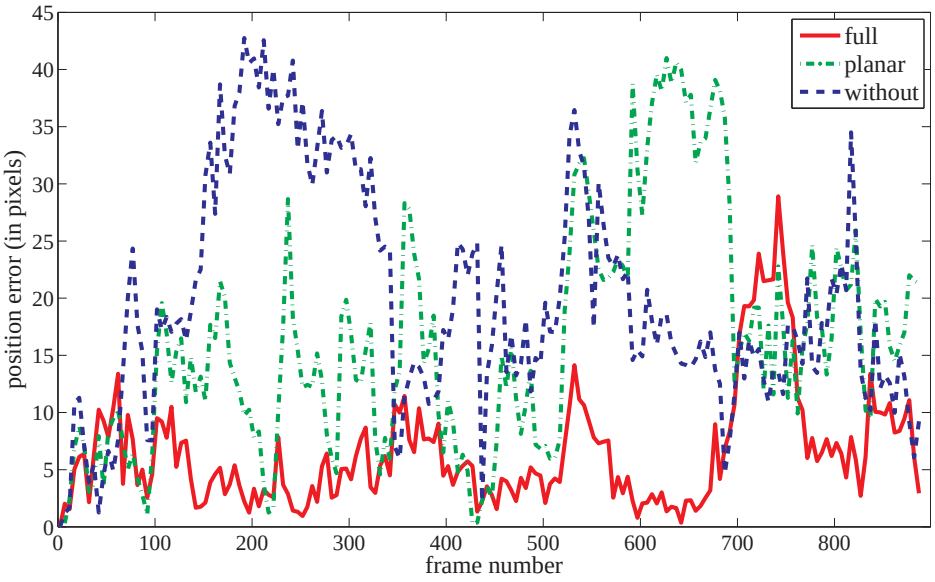


Figure 7: Experiment 1: Deviation from ground truth. (full) using HSS with a fully connected graph, (planar) using HSS with a triangulated graph, and (without) using only tracking with Mean Shift.

5.7. Experiment 3: Articulated motion under scaling

In experiment 3 the aim is to successfully track an articulated object consisting of 8 parts connected via 6 articulation points (jumping jack). The challenges are the scaling (approximately from 100 % to 130 % and to 80 %.) and the two types of motion: articulated and camera.

In Figure 9 one can see three frames of the video. Figure 10 shows the deviation from the manually labeled ground truth of tracking with Mean Shift alone, of our approach with HSSs represented by planar triangulated graphs or fully connected graphs. As expected there is no remarkable difference in the results for planar and fully connected graph.

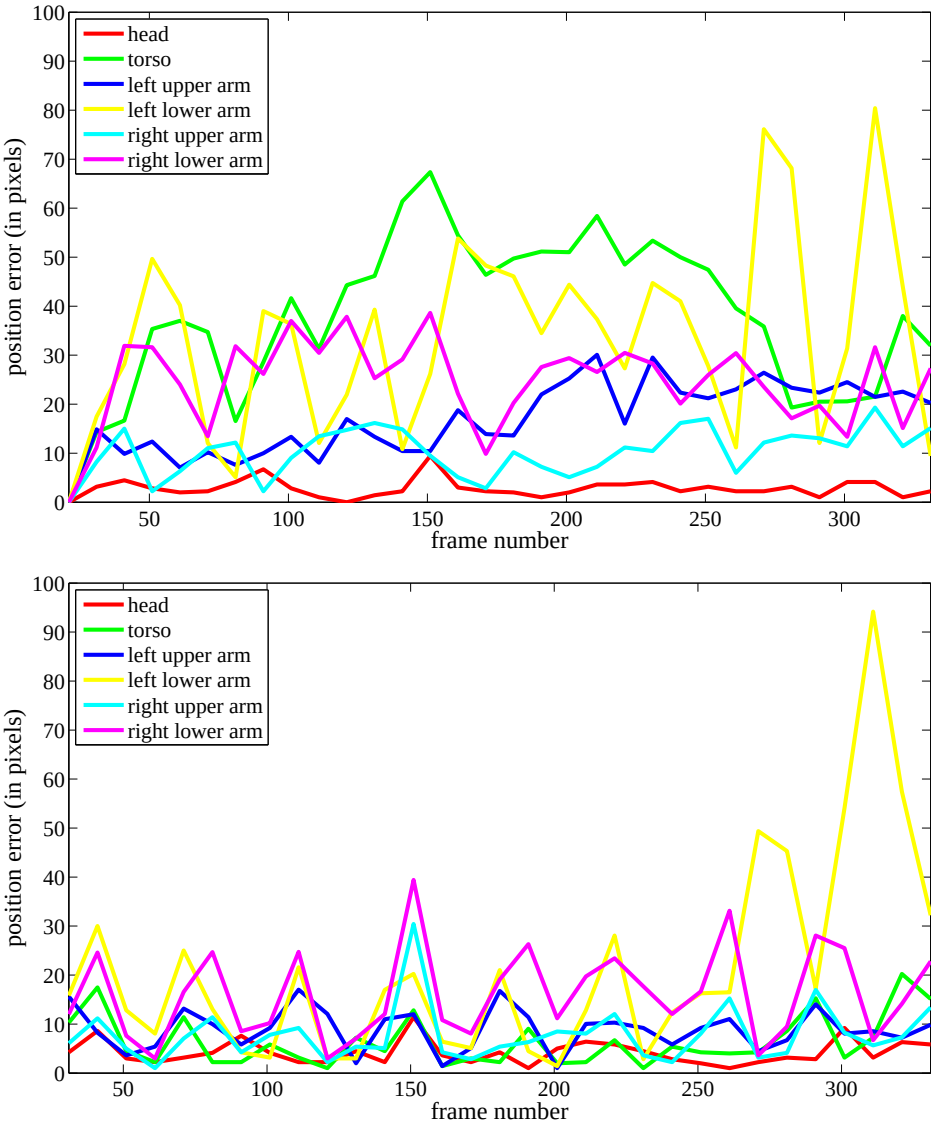


Figure 8: Experiment 2: Deviation from ground truth: (top) tracking with Mean Shift, (bottom) tracking with our approach with fully connected graphs.

5.8. Experiment 4: Fast movements

In this experiment the robustness and recovery potential of the HSS is tested. The employed video shows a woman waving a hand very fast, which

Frame 1 Frame 118 Frame 621

Figure 9: Experiment 3: Some frames of the video showing the scaling.

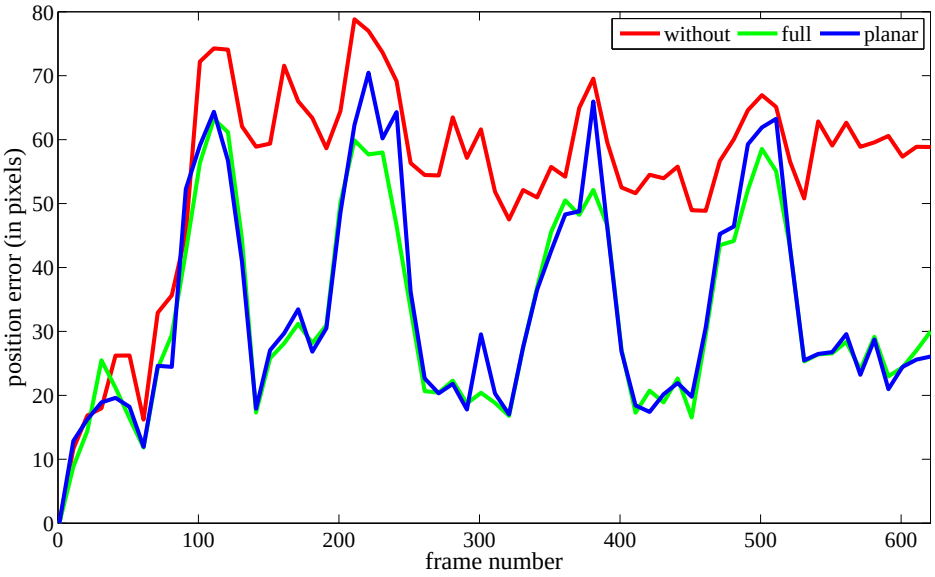
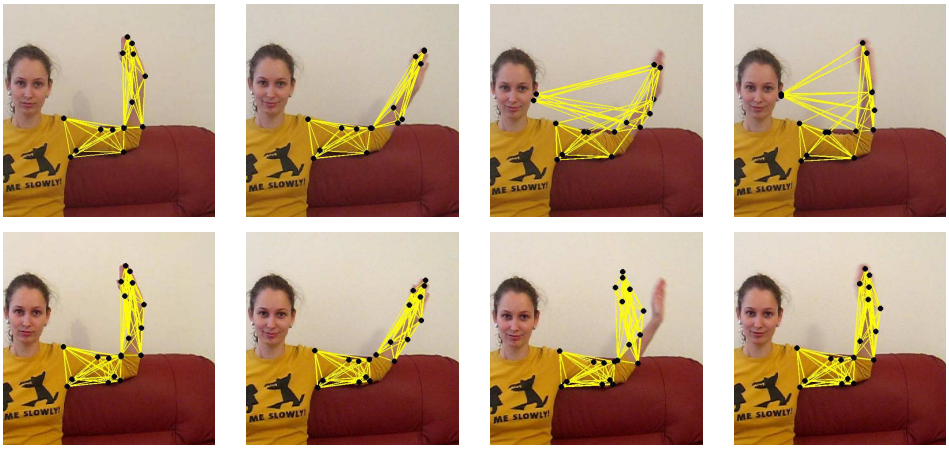


Figure 10: Experiment 3: Deviation from ground truth. The position error in pixels is a sum over the error of all object parts.

leads to heavy motion blur.

Figure 11 shows some frames of the video sequence including qualitative results for tracking with Mean Shift alone and our approach with fully connected graphs. Frames 155 and 170 show the superior results of our approach in comparison to Mean Shift on its own. Figure 12 evaluates the results in



Frame 1 Frame 68 Frame 155 Frame 170

Figure 11: Experiment 4: Tracking an articulated object through motion blur. (top) Tracking with Mean Shift and (bottom) our approach with HSS and fully connected graphs.

concrete numbers.

5.9. Experiment 5: Tracking a whole human

In experiment 5 representations with 10 object parts and 9 articulations points are built and track walking humans in 04002 and 04006 in view vr7_7 of [15]. Fig. 13 shows images of 04002 and 04006, where in (d) one can see that for some parts it is not possible to extract enough local features. In such cases also tracking is more difficult and depends mainly on the top level of the HSS. As expected tracking with our approach by combining Mean Shift and HSSs delivers the better result (see Fig. 14).

5.10. Discussion and future work

The presented experiments showed the application of the proposed framework in tracking objects of different complexity under "simple" motion, ar-

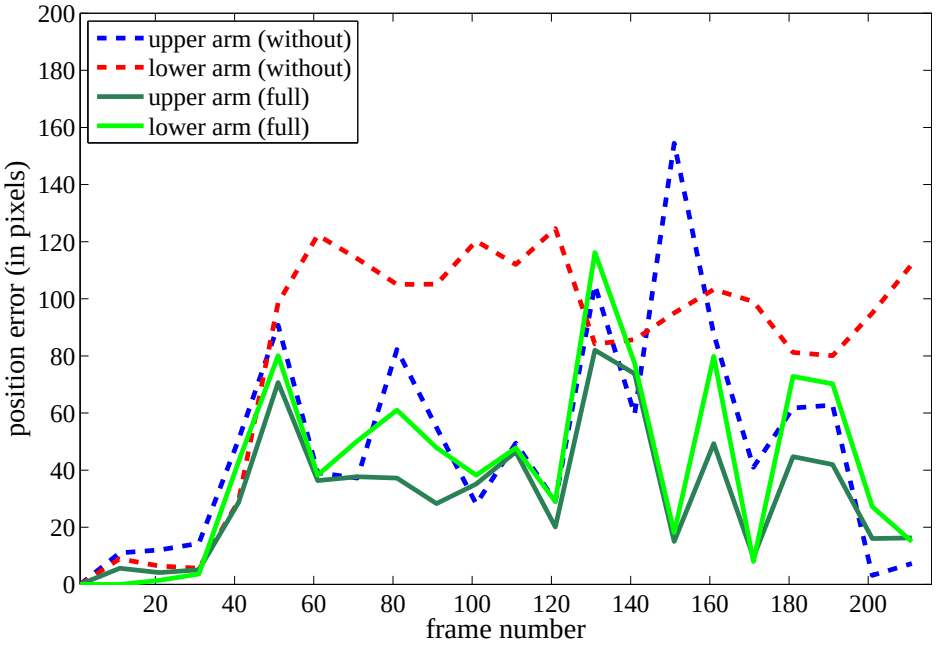
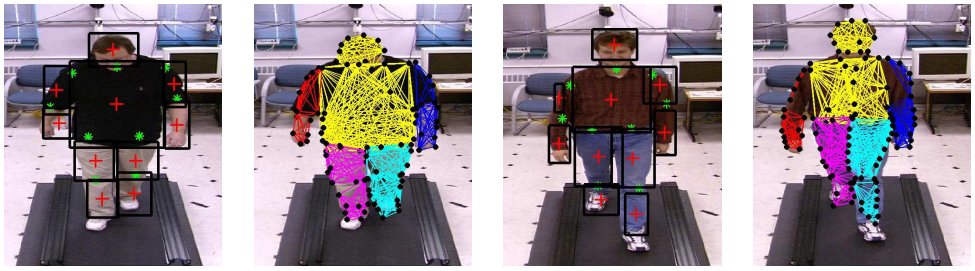


Figure 12: Experiment 4: Deviation from ground truth. (without) tracking the object parts with Mean Shift, (full) our approach with fully connected graphs.



(a) (b) (c) (d)

Figure 13: Experiment 5: (a) frame of subject 04002 with the top level of the HSSs and the articulation points, (b) subject 04002 and corresponding bottom level of HSSs, (c) frame of subject 04006 and its top level with the articulation points, and (d) showing the bottom level of the HSSs of 04006.

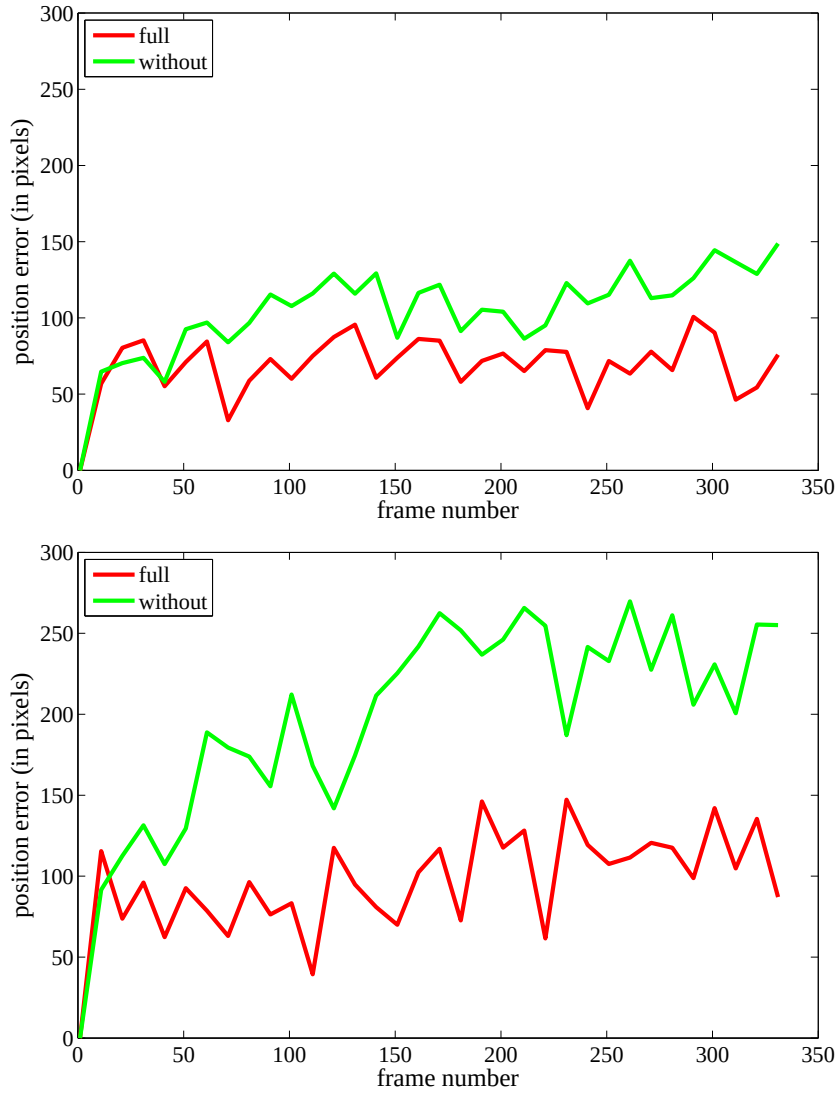


Figure 14: Experiment 5: Deviation from ground truth. (top) video with subject 04002 in view vr7_7, (bottom) subject 04006 in the same view. For both videos results with Mean Shift (without) and with our approach (full) are shown. The position error in pixels is a sum over the error of all object parts.

ticulated motion, camera motion, scaling, occlusion, and motion blur.

Even though tracking with Mean shift and Sigma Sets are employed as basic building blocks, both the tracker and the region descriptor are exchangeable. The focus of our work lies in the hierarchical representation.

The experiments in this section showed that a fully connected graph as representation for a HSS is equal or superior to a triangulated graph (especially during occlusions). Therefore, we intend to exclusively employ this representation in future. The increase in processing time is insignificant, as most of the processing time (approximately 95 %) is spent in calculating region descriptors and building distributions.

Besides its advantages during occlusion, the fully connected graph is also a good basis to start future research on updating the elements of the HSS. When an object moves in the 3D space (e.g. turning around) it happens that some regions of the object become invisible and new regions appear. Therefore, it is necessary to develop an update process for the elements of the HSS, which allows the removal of "old" vertices and the addition of "new" ones. This process requires changes in the graph representing the HSS and here a fully connected graph is easier to handle than a triangulation.

Furthermore, we plan to extend our HSS to be able to handle 3D position information. One possibility to realize this, could be to stick with Mean Shift tracking in 2D, but optimize the Spring System in 3D coordinates.

6. Conclusion

This paper presented a flexible framework to represent and track articulated objects consisting of several rigid parts connected with articulation

points. The parts of the object are described by a Hierarchical Spring System which is represented by an AG pyramid. The attributes of the pyramid are region descriptors and the edges encode the spatial relationships between the vertices/attributes. This spatial structure is enforced during tracking by the spring-like behavior of the edges in the Hierarchical Spring Systems. The "springs" allow to determine structural offsets vectors, which are combined with the offset vectors provided by the employed Mean Shift tracker. Position information can be transfered between the parts over the corresponding articulation points depending on the confidence of the parts and their features.

Appendix A: Sigma Set

Hong et al. introduced the *Sigma Set* [17], a novel second order statistics based region descriptor. The sigma set descriptor is based on the covariance matrix descriptor, which was first introduced as a region descriptor by Tuzel et al. [18]. Covariance matrices are invariant to scaling and rotation up to a certain degree (depends on the feature selection) and allow the combination of multiple features in an elegant way. Furthermore, compared to other region descriptors, region covariance is low-dimensional and can be efficiently calculated using integral images. However, there are evident disadvantages enumerated by Hong et al., which led to the development of the Sigma Set (e.g. covariance matrices do not lie on the Euclidean space, which requires time-consuming operations through Rienmannian geometry).

The covariance matrix descriptor [18] can be extracted out of a two dimensional image I of size $W \times H$. F is a feature image of size $W \times H \times d$

extracted from I, encoding a feature vector of size d at each position $F(x, y)$:

$$F(x, y) = \phi(I, x, y), \tag{17}$$

where the function ϕ can be any mapping including e.g. intensity, color, gradients and so on. A rectangular region of interest $R \subset F$ can be represented by the $d \times d$ covariance matrix

$$C(R) = \frac{1}{n-1} \sum_{k=1}^{n} (z_k - \mu)(z_k - \mu)^T, \tag{18}$$

where $\{z_k\}_{k=1\ldots n}$ are the d-dimensional feature vectors of the points in R and μ is the mean over all points.

The basic idea of Hong et al. [17] is to find a small set of points S which satisfies $C(S) = C(R)$ so that S is *equivalent* to R in terms of 2^{nd} order statistics. They employ the Cholesky decomposition to construct the Sigma Set descriptor S for a region R from the corresponding covariance matrix $C(R)$. The space complexity of Sigma Set is $(d^2 + d)/2$. For example for a color image I with a feature image $F = W \times H \times 3$ the extracted Sigma Set S has 1×6 dimensions.

Hong et al. choose the modified Hausdorff distance (MHD) to evaluate the distance h between Sigma Sets [17]:

$$h(S_A, S_B) = \frac{1}{2d} \sum_{a \in S_A} \min_{b \in S_B} (d_E(a, b)) \tag{19}$$

where S_A and S_B are two Sigma Sets and $d_E(\bullet)$ can be any distance metric defined in R^d, such as the Euclidean distance (L2 Norm).

As Sigma Set is derived from the covariance matrix uniquely, it inherits its

robustness and certain invariance against scale and rotation changes. This is essential in the presented approach to successfully associate regions in consecutive frames of a video.

In our previous work on tracking with Spring Systems [8] we employed covariance matrix descriptors. The deciding fact to chose Sigma Set as region descriptor over covariance matrix is the more efficient distance evaluation. This evaluation is obligatory in every frame and critically influencing the running time.

References

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

[2] C.-S. Lee, A. Elgammal, Coupled visual and kinematic manifold models for tracking, International Journal of Computer Vision 87 (2010) 118–139.

[3] M. Brubaker, D. Fleet, A. Hertzmann, Physics-based person tracking using the anthropomorphic walker, International Journal of Computer Vision 87 (2010) 140–155.

[4] M. A. Fischler, R. A. Elschlager, The representation and matching of pictorial structures, Transactions on Computers 22 (1973) 67–92.

[5] P. F. Felzenszwalb, Pictorial structures for object recognition, IJCV 61 (2005) 55–79.

[6] D. Ramanan, D. Forsyth, Finding and tracking people from the bottom up, in: CVPR, Vol. 2, IEEE, 2003, pp. 467–474.

[7] S. F. F. Gibson, B. Mirtich, A survey of deformable modeling in computer graphics, Tech. rep., Mitsubishi Electric Research Laboratories (1997).

[8] N. M. Artner, A. Ion, W. G. Kropatsch, Coarse-to-fine tracking of articulated objects using a hierarchical spring system, in: International Conference on Computer Analysis of Images and Patterns, Springer, Münster, Germany, 2009, pp. 1011–1018.

[9] N. M. Artner, A. Ion, W. G. Kropatsch, Rigid part decomposition in a graph pyramid, in: J. O. E. Eduardo Bayro-Corrochano (Ed.), The 14th Iberoamerican Congress on Pattern Recognition, LNCS, Springer, 2009, pp. 758–765.

[10] J. Yan, M. Pollefeys, A factorization-based approach for articulated non-rigid shape, motion and kinematic chain recovery from video, PAMI 30 (5) (2008) 865–877.

[11] T. Walther, R. P. Würtz, Unsupervised learning of human body parts from video footage, in: 2nd Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, 2009, pp. 336–343.

[12] S. Drouin, P. Hébert, M. Parizeau, Incremental discovery of object parts in video sequences, CVIU 110 (2008) 60–74.

[13] N. Artner, A. Ion, W. G. Kropatsch, Tracking objects beyond rigid motion, in: Workshop on Graph-based Representations in Pattern Recognition, Springer, 2009.

[14] N. M. Artner, A. Ion, W. G. Kropatsch, Rigid part decomposition in a graph pyramid, in: Iberoamerican Congress on Pattern Recognition, Springer, Mexico, 2009, pp. 758–765.

[15] R. Gross, J. Shi, The cmu motion of body (mobo) database, Tech. Rep. CMU-RI-TR-01-18, Robotics Institute, Pittsburgh, PA (June 2001).

[16] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: CVPR, 2006, pp. 798–805.

[17] X. Hong, H. Chang, S. Shan, X. Chen, W. Gao, Sigma set: A small second order statistical region descriptor, in: Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1802–1809.

[18] O. Tuzel, F. Porikli, P. Meer, Region covariance: A fast descriptor for detection and classification, in: ECCV, Springer, 2006, pp. 589–600.

***Author Biography**

MSc Nicole M. Artner received the “Bachelor” form the University of Applied Science of Hagenberg, Austria in 2006, and the “Mater of Applied Science” from the University of Applied Science of Hagenberg, Austria in 2008. She started her PhD in 2008 and her research interests lie in the field of Computer Vision and include hierarchical, structural, and part-based representations and tracking of articulated objects.

Dr. Adrian Ion received the "Inginer Diplomat" from the "Politehnica" University of Timisoara, Romania in 2001, and the Dr. techn. in computer science from the Vienna University of Technology in 2009. His research interests include graph based representations and algorithms, irregular pyramids, and the analysis of 2D and 3D shapes. Currently, Dr. Ion is a post doctoral researcher at the University of Bonn, Germany.

Prof. Walter G. Kropatsch has joined the PRIP group at the Vienna University of Technology in 1990. Prof. Kropatsch has received his PhD in computer science from the Technical University of Graz, in 1982, and his habilitation (Venia Docendi) from the University of Innsbruck, in 1991. He has done sustained research in the fields of irregular pyramids, graphs, and their application in various computer vision areas.