Tracking a Hand Manipulating an Object

Henning Hamer[1] Konrad Schindler[2] Esther Koller-Meier[1] Luc Van Gool[1,3]

[1]Computer Vision Laboratory
ETH Zurich
{hhamer,ebmeier,vangool}@vision.ee.ethz.ch

[2]Computer Science Department
TU Darmstadt
schindler@cs.tu-darmstadt.de

[3]ESAT-PSI/VISICS
KU Leuven
luc.vangool@esat.kuleuven.be

Abstract

We present a method for tracking a hand while it is interacting with an object. This setting is arguably the one where hand-tracking has most practical relevance, but poses significant additional challenges: strong occlusions by the object as well as self-occlusions are the norm, and classical anatomical constraints need to be softened due to the external forces between hand and object. To achieve robustness to partial occlusions, we use an individual local tracker for each segment of the articulated structure. The segments are connected in a pairwise Markov random field, which enforces the anatomical hand structure through soft constraints on the joints between adjacent segments. The most likely hand configuration is found with belief propagation. Both range and color data are used as input. Experiments are presented for synthetic data with ground truth and for real data of people manipulating objects.

1. Introduction

Visual hand tracking has several important applications, including intuitive human-computer interaction, human behavior and emotion analysis, safety and process integrity control on the workfloor, rehabilitation, and motion capture. Not surprisingly, much research has already gone into computer algorithms for hand tracking. Yet, the majority of contributions have only considered free hands, whereas in many applications the hands will actually be manipulating objects. In this paper, we present for the first time a system, which can track the articulated 3D pose of a hand, while the hand interacts with an object (such as depicted in Fig. 1).

The presence of objects has a significant impact on the complexity and generality of the task. First, the manipulated objects will frequently occlude parts of the hand, and hand poses occurring during the process of grabbing or holding will aggravate the problem of self-occlusion (*e.g.* in Fig. 1 large parts of four fingers are partially or even fully occluded). Second, the hand structure itself is less constrained in the presence of objects: parameter ranges have to be widened and some simplifying assumptions derived from human anatomy no longer hold. When in contact with an object, forces are exerted on the hand, resulting in poses which cannot be achieved with the bare hand (*e.g.* bending fingers backwards when pressing against a rigid surface, breaching the "2/3-rule" between the joints of a finger when pushing a button, etc.). Tracking hands under these less favorable conditions is the topic of this paper. To the best of our knowledge, visual hand tracking in the presence of objects is uncharted terrain.

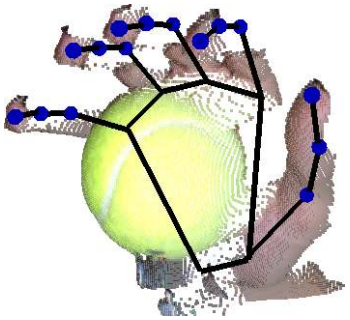


Figure 1. The goal of the present work: recovering the articulated 3D structure of the hand during object manipulation.

Object manipulation is an inherently 3-dimensional phenomenon, whereas 3D pose estimation in monocular video is seriously under-constrained. We therefore base our estimations not only on image color, but also on 2.5D depth maps. In our case, the depth maps are obtained with a real-time structured light system [19], but in the near future such data will in all likelihood be available at negligible cost, due to the rapid progress of time-of-flight sensors [14, 3].

Our approach has been inspired by an established trend in object recognition and detection. Occlusion is a frequent and not reliably solved problem in these applications. Models are split into local parts, and each part separately contributes evidence about the complete model. In this way robustness to partial occlusion is achieved and the estimation relies only on observable parts, *e.g.* [11, 10]. The underlying global configuration can then be used to infer information regarding the occluded parts. In much the same way,

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

we intentionally refrain from employing a single high dimensional model, but use local 6 DOF trackers for individual hand segments. We then exploit anatomic constraints between adjacent segments to enforce the hand structure. The constraints are represented by a first order Markov random field (MRF). Each of the local trackers corresponds to one rigid hand segment, and independently recovers a *pdf* over the segment pose from local evidence. Then, a valid hand structure is enforced by belief propagation [15] on the hand graph. An explicit occlusion model makes sure that missing local observations do not corrupt the estimation.

2. Related Work

While some state-of-the-art hand trackers rely on detection mechanisms [16, 1], many others estimate hand configurations based on an articulated, connected model [17, 18, 21, 4]. A common problem in articulated tracking is the "curse of dimensionality": depending on the exact model, the human hand has ≈ 26 degrees of freedom. Accordingly, an exact representation of the posterior distribution over model configurations is intractable, and non-parametric approximations of the whole high-dimensional search space are difficult to maintain. Sampling-based approaches such as particle filters cannot cope without measures to drastically cut down the search space or reduce the dimensionality of the problem [12, 20]. A recent summary of state-of-the-art methods can be found in [7].

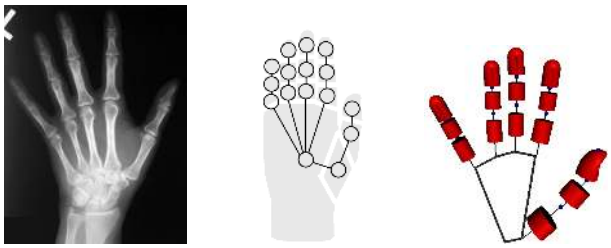
Given the additional problem with strong occlusions, which routinely occurs in our object manipulation scenario, we follow a different strategy, and divide the articulated structure into local segments which are tracked individually.

Our work has been inspired by [17], which introduced the idea of belief propagation on a graph consisting of local hand parts. Although the approach has been extended in [18] to incorporate self-occlusion between hand-parts, their method targets only bare hands. In our work, we consider not only self-occlusion but also occlusion by an object, which is handled explicitly with an occlusion model.

In [17, 18], sampling is performed along the kinematic chain. We refrain from this for two reasons: firstly, the observation of the palm is important in such an approach, but the palm is often occluded during object handling; secondly, such a sampling imposes hard constrains on the joints, since samples even slightly violating anatomical constraints are never drawn. However, some anatomical constraints no longer hold strictly when in contact with an object.

Instead of sampling along the kinematic chain, we focus on the independence of the local trackers and sample from local proposal functions. To cover the state space appropriately we proceed hierarchically. To avoid impossible configuration, we impose soft constraints, by penalizing a sample's deviation from a valid hand shape.

For better accuracy, especially in depth, we model the hand as a collection of surface patches, rather then only silhouette edges, which allows for a richer representation. The observation likelihood of these patches is measured using a modified 3D chamfer distance.



(a) Human hand (b) Hand graph (c) Hand model

Figure 2. (a) An X-ray picture of a human hand shows the 27 bones. Image courtesy of M. L. Richardson[1]. (b) The hand graph. (c) The complete hand model consisting of a skeleton and ruled surfaces for the skin.

None of the previous approaches are concerned with articulated hand tracking in the presence of objects. Object manipulation is targeted in the literature only in terms of action and object interpretation [9, 13] without exact pose estimation, so no detailed, high-dimensional model of the hand is provided. [9] for example focuses on the recognition of the general kind of manipulation and the manipulated object. Hand gestures are classified on a per-frame basis using 2D image features and learned sample sequences. [13] considers the relationship between objects, like an attachment or a contact between them, with the goal to explain the given scene. Hands are not treated separately and the objects are recognized and tracked from given 2D templates.

A taxonomy of human hand poses with regard to the grasping of objects has been provided in [5]. In [8], manipulative hand gestures are visually recognized using a state transition diagram that encapsulates the task knowledge. The feature extraction is based on thresholding the hue value, so that the person has to wear special gloves, and the gestures are simulated, without a real object involved. [6] recognizes grasps referring to the grasp taxonomy defined in [5]. Real objects are handled, but do not impair the hand observation, because a data glove, rather than visual input, provides the hand pose.

3. Hand Model

The human hand consists of 27 bones (see Fig. 2(a)). Eight of these are located within the wrist, and four make up the palm. We ignore both the bones within the wrist and those in the palm (following [20]), and represent the palm as a single segment in our model. Thus, we consider $3 \times 5 = 15$ phalanges for the five fingers, plus the palm which is defined as the remaining hand above the wrist. As

[1] University of Washington, http://uwmsk.org/RadAnat

the wrist itself is irrelevant in terms of object grasping [5], we do not include it in the model. The degrees of freedom between hand segments are constrained by revolute joints: hinge joints between finger phalanges allow only for bending within a certain range (*flex*, 1 DOF); saddle joints connecting the fingers to the palm additionally allow for spreading the fingers (*flex* and *abduction*, 2DOF). Neither hinge nor saddle joints permit a *twist* around the bone axis.

3.1. Local Hand Segments

In our model, each hand segment (either phalanx or palm) has its own six dimensional state space, with three dimensions corresponding to the position of the segment and the other three to its orientation (96 DOFs for the whole hand). The state of a segment is represented by a local coordinate system aligned with the segment. To complete the model, we associate every phalanx with a mesh approximating the skin. The mesh has the form of a cylinder for the middle and base segments of the fingers. In case of the tips, it corresponds to a cylinder with a spherical cap (Fig. 2(c)).

3.2. Anatomical Constraints

Since each segment has its own pose, constraints are required to ensure that neighboring segments stay connected at the joints, and that their respective orientations result in a valid hand configuration.

Note that in the chosen parameterization the constraints obey the (first order) Markov property (*i.e*. they apply only to adjacent segments), and that the hand graph is a tree with the root at the palm and leaves at the fingertips as depicted in Fig. 2(b). The constraints can therefore be efficiently optimized with belief propagation.

We use soft constraints: to make sure hand segments stay (nearly) connected, we employ *proximity constraints*, meaning that we penalize configurations of neighboring segments proportionally to the distance between their endpoints. To ensure valid joint angles, we use *angle constraints*. As already argued, the traditional anatomical limits for the free hand are no longer valid in contact with objects, so enlarged angle ranges are used. Details about the penalty function and how it exploits the constraints are given in Sec. 4.3.

4. Tracking Method

Every segment of the hand model has its own local tracker. In each computation step, the local tracker draws a number of samples from a local proposal function. The sample space is 6D – three parameters for the position of the segment, and three rotation angles. Fig. 4(d) shows exemplary hand segment samples. We sample each parameter uniformly, within a different range to account for the kinematics of the human hand (*e.g*. it is easy to confirm that we can bend our fingers faster than we can spread them). For each sample, the likelihood is computed by comparing it locally to the observation (Sec. 4.1), taking into account occlusion information (Sec. 4.2). Then, belief propagation is applied to combine the evidence of the local trackers (Sec. 4.3). The resulting weights, together with the corresponding samples of a hand segment, are a discrete representation of the posterior *pdf* over the segment's pose. The posterior *pdf* is then transformed to the next time step with a dynamic prediction to yield the new proposal function.

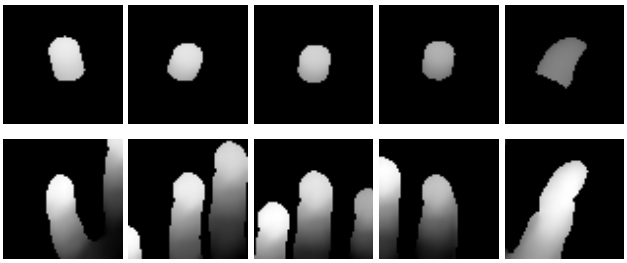


Figure 3. Top row: examples of model patches – one rendered sample for each finger tip (from left to right: little, ring, middle, index, thumb). Bottom row: the corresponding data patches. Often, parts of more than one finger are contained. Black areas represent background (unknown/infinite depth).

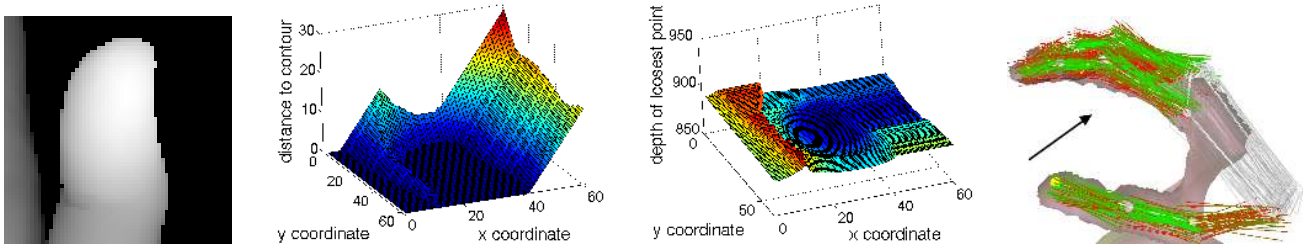
4.1. Observation Model

As input, our system uses range data with a resolution of 640×480 depth points recorded at 25 Hz and conventional two-dimensional color images, see [19]. The mean error in depth of the range data is approximately 2 mm. Color information is exploited in a preprocessing step to locate the hand and to detect object occlusion (see Sec. 4.2) via skin color segmentation.

The local observation of a phalanx consists of a rectangular patch of range data D (the "data patch") around the predicted position. We do not consider the observation of the palm, which deforms less rigidly than the other segments, and is often occluded during object handling.

To compare a pose sample for a given phalanx to the data, its local surface mesh is rendered into a depth image, using the known camera calibration. This projection yields a depth image M (the "model patch"), which is in pointwise correspondence with the range data for that segment, so that the two patches are directly comparable. Fig. 3 provides examples of model and data patches. The patches $\{M_i\}$ for the entire set of samples can be computed efficiently by rendering all samples as one big texture on the GPU.

To evaluate the likelihood of a sample, we compare its rendered depth image M to the corresponding data patch D with a simple distance measure $d_{\mathbf{x}}$, computed over all pixels $\mathbf{x}_M$ of the hand surface in the model patch. If a pixel $\mathbf{x}_M$ belongs to the surface in both the model *and* the data patch,



(a) Depth data (b) Distance Transform (c) Depth of Closest Observation (d) Observation Likelihood

Figure 4. Extended model-to-data distance. (a) Depth observation of the thumb tip. (b) Distance transform showing for each pixel the (x, y)-distance to the nearest skin point in (a). (c) Extended distance transform showing for each location the depth of the nearest skin point. (d) Hand segment samples. Color encodes relative observation likelihood: green is highest, red is lowest. The palm has no model, hence uniform observation likelihood. The arrow indicates the viewing direction of the camera.

we directly use the depth difference, $d_{\mathbf{x}} = z_M - z_D$. If the hand pixel in the model at location $\mathbf{x}_M$ does not belong to the hand according to the data, then we use a generalized distance to the nearest hand pixel $\bar{\mathbf{x}}_D$:

$$d_{\mathbf{x}} = \sqrt{(x_M - \bar{x}_D)^2 + (y_M - \bar{y}_D)^2 + (z_M - \bar{z}_D)^2}\ . \tag{1}$$

The distance between $\mathbf{x}_M$ and $\bar{\mathbf{x}}_D$ can be computed efficiently using an extended 3D distance transform. The extended distance is illustrated in Figs. 4(a)–4(c).

In comparing the data with the model, we do not consider the situation where there is a hand pixel in D, but not in M (unexplained observation): it cannot be decided locally, whether the data is observed by another hand segment, since the local tracker has no information from other parts of the hand. The likelihood of a sample M_i is hence defined as

$$L(D|M_i) = \frac{1}{Z}\, e^{-\left(\frac{\widehat{d}}{\sigma_{obs}}\right)^2}\ , \tag{2}$$

where σ_{obs} is a user parameter which specifies the accuracy of the range data. Z is a normalization factor, which assures probability distributions integrate to 1, and will from now on be omitted for brevity. $\widehat{d}$ is the mean value over all T considered distances, $\widehat{d} = \frac{1}{T}\sum_{\mathbf{x}_M \in M_i} d_{\mathbf{x}}$.

We prefer to use the average error for the sample, rather than computing an individual likelihood $L(D|\mathbf{x}_M)$ for each pixel and multiplying them together. This choice is motivated by the nature of range scanners: the point density of such systems depends on the surface orientation. Furthermore, there are occasional missing depth values, and these tend to be clustered, forming holes in the observed surface. These two properties may cause a heavy bias in error measures that depend on the number of observed pixels.

4.2. Occlusion Model

Given the large amount of occlusion during object manipulation, an explicit occlusion model is required to achieve robustness.

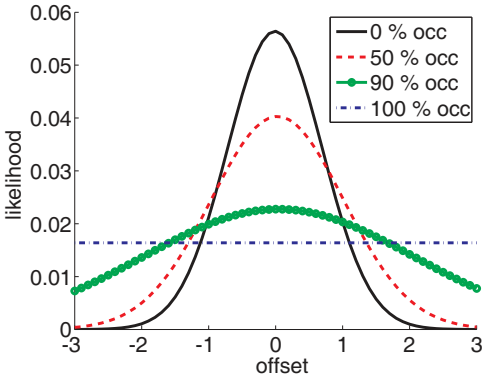


Figure 5. Illustration of the occlusion model. If the model is moved away from its correct position (offset 0), its likelihood decreases. However, with increasing occlusion the difference in likelihood becomes smaller, as less and less evidence supports it.

After obtaining the observation patch of a hand segment, we label self-occlusion within the patch. Accurate detection of self-occluded pixels requires the global hand configuration. Locally, the only way to detect self-occlusions is to find regions where the data is substantially closer to the camera than predicted by the model. We do this by applying a distance threshold of 10 mm (approximately the diameter of a finger). Object points (depth observations inconsistent with the skin color model) are also regarded as occluders, if they are closer to the camera than the skin mesh of a sample.

When computing the mean depth error $\widehat{d}$ as explained in Sec. 4.1, we do not count the occluded model pixels as part of the model, which is equivalent to assigning the average error of the visible pixels to the occluded ones. Such a definition does not penalize hand samples for moving into occlusion, but attracts them to the data, as soon as they move out of the occlusion. This behavior has proved to be desirable in our experiments.

As explained above, we do not take into account the number of observed points on a hand segment, because it

would bias the estimation from range data. However, in the presence of occlusion the amount of visible surface *does* matter: if a hand segment is completely occluded, the *pdf* from the observation model should be uniform, since there is no information available about its pose. Moreover, if it is *partially* occluded, there are observations about the pose of the segment, but they should not carry the same weight as for an unoccluded segment, both due to the smaller number of observations and to the narrower field of view. We introduce a smooth dependence on the amount of occlusion by introducing an additional factor α in the exponent of Equation 2 as follows:

$$L(D|M_i) = e^{-\alpha \cdot \left(\frac{\hat{d}}{\sigma_{obs}}\right)^2} , \quad (3)$$

where α is the fraction of unoccluded pixels, estimated from the predicted state. Intuitively, this definition produces peakier distributions for unoccluded segments, and gradually flatter distributions as the degree of occlusion increases. In the extreme case of total occlusion, the exponent in (3) vanishes. All samples are assigned equal likelihood, and the pose is entirely determined by the structural constraints. Fig. 5 graphically illustrates this definition.

4.3. Enforcing Constraints

As already discussed, the (soft) constraints modelling the structure of the hand can be divided into two categories, those acting on the position of neighboring segments (*proximity constraints*), and those acting on their orientation (*angle constraints*). The constraint network is a tree obeying the first-order Markov property, hence constraint enforcement by belief propagation will yield a globally optimal configuration. When sending a message from node i to node j, a $S \times S$ constraint matrix (S being the amount of samples at each node) is computed for all possible combinations of samples of the two nodes. The matrix entries are the probabilities of observing the respective sample pair u_i, u_j, and are defined as

$$\psi(u_i, u_j) = p_{prox}(u_i, u_j) \cdot p_{ang}(u_i, u_j) , \quad (4)$$

with p_{prox}, p_{ang} the two types of constraints.

Proximity constraints make sure the hand segments stay connected. We define the proximity error ϵ_{prox}—the degree to which adjacent segments violate the constraint—as the Euclidean distance between the corresponding endpoints,

$$p_{prox} = e^{-\left(\frac{\epsilon_{prox}}{\sigma_{prox}}\right)^2} . \quad (5)$$

The parameter σ_{prox} specifies the importance of the observed error, and also the relative weight of this error against those of the angle constraint errors to be defined next. We set σ_{prox} to 5 mm.

Angle constraints are defined in a similar way. In analogy to the segments of the hand model, each sample has a local coordinate system. Consider the angles (*flexion*, *abduction*, *twist*) rotating the local coordinate system of a sample of node i into the local coordinate system of a sample of node j. We compare these angles to anatomically valid angles for the connecting joint and compute error values ϵ_{flex}, ϵ_{abd} and ϵ_{twist} such that

$$p_{ang} = e^{-\left(\frac{\epsilon_{flex}}{\sigma_{flex}}\right)^2} \cdot e^{-\left(\frac{\epsilon_{abd}}{\sigma_{abd}}\right)^2} \cdot e^{-\left(\frac{\epsilon_{twist}}{\sigma_{twist}}\right)^2} . \quad (6)$$

Again, σ_{flex}, σ_{abd} and σ_{twist} encode the relative importance of the different constraints, and their importance compared to the other observed errors. In our case, $\sigma_{flex} = \sigma_{abd} = \sigma_{twist} = 10$ degrees.

Once all samples have been generated and their likelihoods with respect to the observations have been computed, belief propagation is used to propagate the local probabilities through the graphical model by marginalization. Each node contains a normalized S-dimensional observation vector ϕ representing a discretized *pdf*. Node i, which has a number of neighbors $N(i)$, sends a message to each node $j \in N(i)$ when i has received messages from all other neighbors $N(i) \setminus j$. The message from i to j may be interpreted as the opinion of i regarding the state of j. With the likelihoods defined above, the message from i to j for a specific sample u_j has the form

$$m_{i \to j}(u_j) = \sum_i \phi(u_i) \cdot \psi(u_i, u_j) \prod_{k \in N(i) \setminus j} m_{k \to i}(u_i) . \quad (7)$$

The product combines the incoming messages for each state. The sum is the marginalization over all possible states of i. Finally, the belief regarding the state of a node i, taking into account the information passed by the neighbors, is computed by

$$b_i(u_i) = \phi(u_i) \prod_{j \in N(i)} m_{j \to i}(u_i) . \quad (8)$$

For a more detailed description of belief propagation, see for example [22].

An obvious constraint, which we have not used so far, is that fingers cannot intersect. The reason is that in our experiments, we have never observed such problems. This testifies to the robustness of the proposed technique, but it also prevents the graph model from including loops (see Fig. 2(b)). This is important as loop-free belief propagation delivers exact marginal probabilities – all available information is distributed throughout the entire graph – and is guaranteed to yield the optimum. Moreover, computational efficiency is better than with loopy belief propagation, which would be the fall-back strategy in case loops need to be included [17]. Hence, including non-intersection of fingers is feasible without major alterations to the system.

4.4. Hierarchical Computation

The local trackers discretize their state space by sampling. The computational cost of evaluating the compati-

bility functions within the belief propagation (see Sec. 4.3) scales quadratically with the amount of samples drawn by the local trackers. To guarantee a sufficiently fine discretization of the state space and an acceptable computation time, we proceed hierarchically. For each frame of the input data, pose sampling, observation evaluation and belief propagation (*i.e.* one complete computation step) are performed several times. At first we sample in a large region of the state space in order to cover the required portion of the space. The S' samples with the highest weights are selected (in our implementation $S' = 5$), and uniform kernels are placed at their positions, as new local proposal functions for the next step. We have experimentally confirmed that the number of modes in the state space is usually ≤ 3, so that no important samples are lost by the intermediate hard decisions.

We use 10 hierarchy levels. When the last level is completed and the transition to the next time step occurs, it has proved beneficial in practice to include a simple dynamic model in the proposal function. In our implementation we use ICP [2] to predict large global hand motion, and a linear (constant velocity) prediction for the motion of individual hand segments. With the prediction step, sampling can focus on deviations from the dynamic model.

5. Results

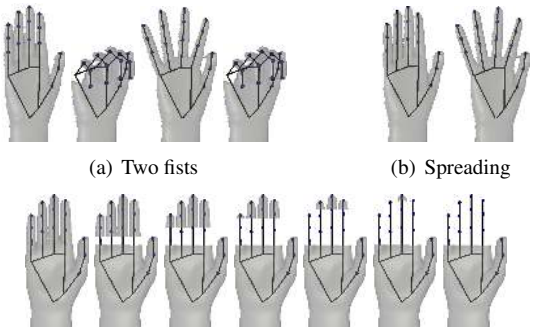
We have conducted experiments with both synthetic and real data. The synthetic data serves for quantitative evaluation, while the real data confirms that the proposed method is applicable to the input delivered by actual range cameras. Computation time of our C++/Cg implementation is ≈ 6.2 sec/frame, with a 3GHz CPU and a GeForce 8800 Ultra.

5.1. Artificial Data

To quantitatively measure the performance of our approach, we generated sequences of artificial 2.5D observations: a 3D scan of a real hand was bound to a 26 DOF forward-kinematic hand model and animated. The animation sequences were then rendered as in a virtual range camera, to obtain 2.5D videos. Two experiments are presented in this category.

Strong self-occlusion. The artificial hand has been tracked over a period of 160 frames, taken at normal video rate. The hand forms a fist twice, producing extreme self-occlusion, once starting with joined fingers and once after spreading them. The sequence is illustrated in Fig. 6(a). As an error measure, we define the error of a phalanx as the mean distance of its two endpoints from those of the ground truth given by the kinematic hand model, and the *frame error* as the mean over all segments in that frame. Frame errors are in the range [0.24... 2.77] mm (mean 1.04, median 0.92). For comparison, the distance from the base of the palm to the tip of the extended middle finger is 230 mm.

Occlusion by an object. To verify the robustness of our method in the presence of an occluding object, we have introduced artificial occluders into a sequence of 45 frames. The hand first spreads and then returns to its initial pose (see Fig. 6(b)). Fig. 6(c) demonstrates the seven tested degrees of occlusion, ranging from no occlusion to full occlusion. The error over all hand segments in the different occlusion scenarios is plotted in Fig. 7. Up to occlusion level four there is almost no increase of the error. At higher levels fingers are fully occluded so their state has to be hallucinated, based only on the anatomic constraints. The system can hardly be blamed for the larger errors in such situations.



(a) Two fists (b) Spreading

(c) Degrees of occlusion

Figure 6. (a) Self-occlusion. A fist is formed twice, starting once with joined and once with spread fingers. (b) Object occlusion. Fingers are spread, then joined again. (c) Different levels of occlusion, ranging from 0% to 100% occlusion of the fingers.

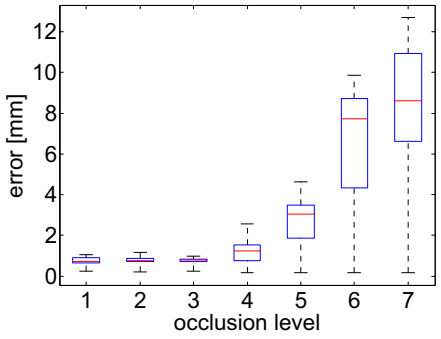


Figure 7. Seven occlusion levels. Occlusion ranges from 0% to 100% occlusion of the fingers. For each level, the median error (red), the lower/upper quartile (box) and the entire error range (whiskers) are displayed.

5.2. Real Data

To assess the validity of our approach for real object manipulations observed by an actual 2.5D camera, we recorded manipulations of 3 different everyday objects with our real-time structured light system. All three sequences contain

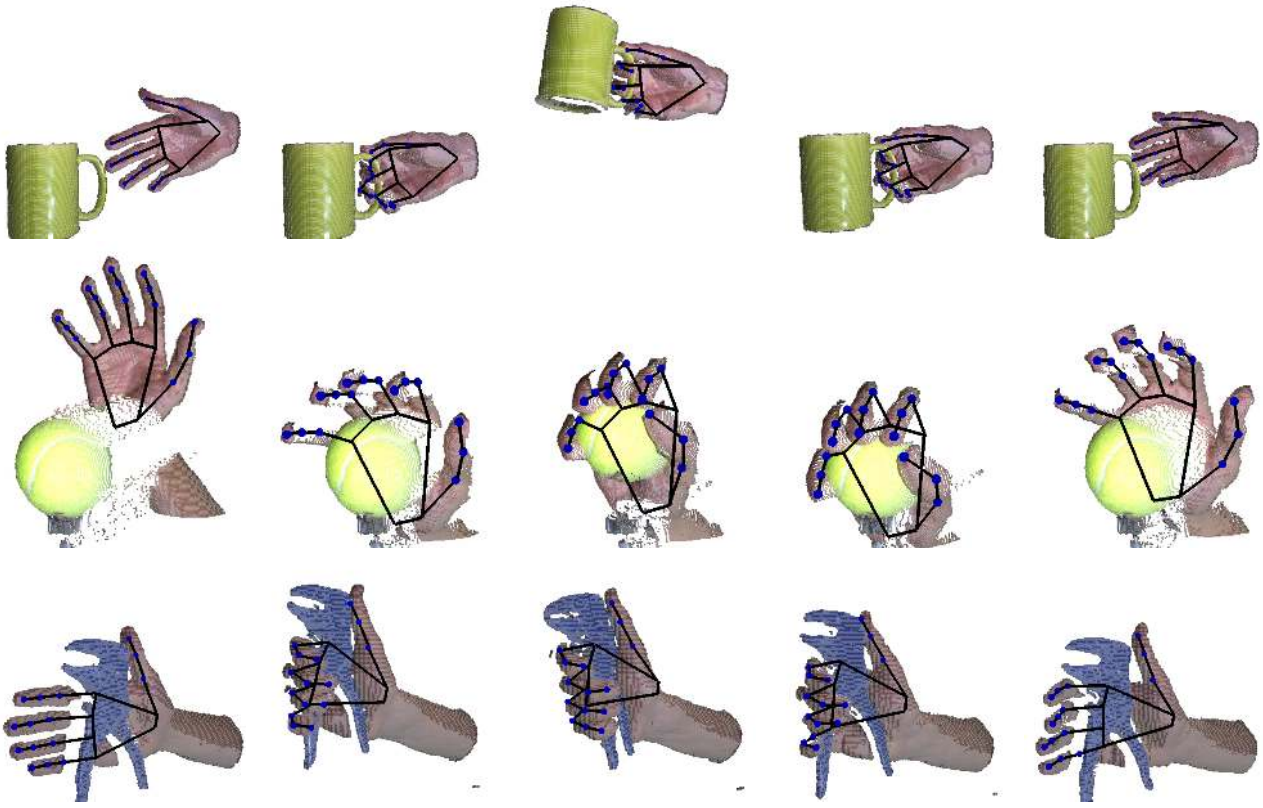


Figure 8. Snapshots of three real data sequences, each containing a different every day object. All three sequences were tracked successfully. In the first two sequences, a cup and a tennis ball are being picked up, lifted, and then lowered back down. In the third sequence, the pliers are not only lifted, but also pressed together. Blue dots represent distal endpoints of the phalanges. There may be a small gap between segments due to our soft constraints. The mean gap of all three sequences is 1.0042 mm (deviation:0.6049 mm).

severe occlusions. In order to represent different aspects of object handling, each of the sequences demonstrates a different grasp type with respect to the grasp taxonomy defined in [5]. Snapshots are provided in Fig. 8. The complete sequences are available as supplemental material.

In all experiments, the initial state of the hand was determined by manual initialization in the first frame. Initialization, while not the topic of this work, could be automated by using a standardized pose, like the "T-pose" in commercial motion capture systems. It may even be possible to initialize the hand pose on the fly, while the hand is not in contact with an object.

The first sequence shows a hand which approaches a mug, grasps it by the handle (*precision grasp*), lifts it up, then places it back on the table and releases the grip. Of particular interest are the moments at which parts of the index and middle fingers are occluded and disoccluded as they grasp the handle. The finger tips smoothly move into the occlusion, since occluded model pixels do not decrease the likelihood of a pose (*c.f*. Sec. 4.2). While occluded, the fingertips continue to move with the rest of the hand, as their local prediction and the proximity constraints at the joints push them forward. As soon as the skin is observed again inside the handle, the model is pulled towards the new observation by the extended distance function $d_{\mathbf{x}}$ (*c.f*. Sec. 4.1), because of the increased penalty for samples far away from the observed skin pixels.

The second sequence shows a hand manipulating a tennis ball. The ball is gripped from behind with a *power grasp*, then lifted up, lowered, and released. The most critical point is shown in the second snapshot. The palm and lower phalanges of the fingers are largely occluded by the ball, and the middle phalanges of the fingers are completely occluded by the finger tips and are about to reappear above them. These segments now have to reattach to the skin. The sequence is tracked successfully, but presents the limit of our current tracker. Fig. 9 shows that the reattachment of the ring finger lags behind several frames, because the amount of visible skin is initially very small. The finger thus continues in an anatomically valid, but inaccurate position, until enough evidence is available for it to recover.

The third sequence is the most complex one. The handled object is a pair of pliers, which is not only lifted with a *hook grasp*, but also pressed together. Note how the hand

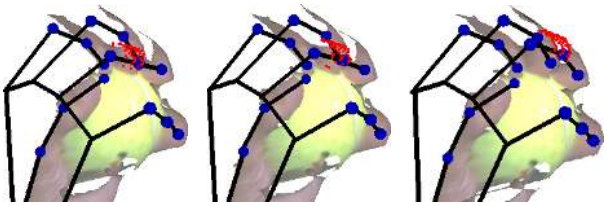


Figure 9. Failure and recovery of a local tracker. The structured light system looks at the scene from the right. The rendered side-view shows how the ring finger initially fails to reattach to the back of the finger after reappearing from behind the tip, but as soon as enough evidence is available, the local tracker recovers.

constraints ensure correct tracking of the fingers in spite of long occlusions and scarce ambiguous evidence (*e.g.* the little finger in the fourth image).

6. Conclusion

We have presented a method for articulated tracking in 2.5D range data, which can track a hand while interacting with objects. The key contributions of the method are to track a configuration of local parts coupled by soft constraints, rather than the complete hand model, and to explicitly model occlusion by both hand parts and objects. Valid hand configurations are enforced by means of a MRF model connecting the segments. Similar to part-based models in other computer vision applications, the use of local parts greatly increases robustness to occlusions, while the soft constraints imposed by the MRF add flexibility in situations in which classical kinematic constraints are too strict due to the influence of handled objects.

Our system offers the information a classical data glove provides. Therefore, previous work based on the output of data gloves can be applied on top of our method—*e.g.* different grasp types like precision and power grasps can be recognized [6] and associated with the grasped object.

We emphasize that our method does not depend on knowledge about the manipulated objects. However, when the object geometry is known (*e.g.* from CAD-models or range scans), then this delivers valuable additional constraints: hand segments cannot penetrate the object. In the future we plan to integrate this information into our model. We also plan to estimate contact points between hand segments and the object, which are important in applications such as the control of robotic hands and the automatic creation of physically plausible animations.

Acknowledgements The authors gratefully acknowledge support through the EC Integrated Project 3D-Coform. We also thank Thibaut Weise for sharing source code.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *CVPR*, 2003.

[2] P. Besl and N. McKay. A method for registration of 3-D shapes. *PAMI*, 14(2):239–256, 1992.

[3] B. Büttgen, T. Oggier, M. Lehmann, R. Kaufmann, and F. Lustenberger. CCD/CMOS lock-in pixel for range imaging: Challenges, limitations and state-of-the-art. Technical report, CSEM, 2005.

[4] T.-J. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *CVPR*, 1999.

[5] M. Cutkosky and P. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. In *ICRA'86*.

[6] S. Ekvall and D. Kragic. Grasp recognition for programming by demonstration tasks. In *ICRA*, 2005.

[7] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *CVIU*, 108(1-2):52–73, 2007.

[8] K.-H. Jo, Y. Kuno, and Y. Shirai. Manipulative hand gesture recognition using task knowledge for human computer interaction. In *FG*, 1998.

[9] H. Kjellström, J. Romero, D. M. Mercado, and D. Kragic. Simultaneous visual recognition of manipulation actions and manipulated objects. In *ECCV*, 2008.

[10] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop*, 2004.

[11] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[12] J. Maccormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV*, 2000.

[13] R. Mann, A. Jepson, and J. M. Siskind. The computational perception of scene dynamics. 65(2):113–128, 1997.

[14] T. Oggier, B. Büttgen, and F. Lustenberger. Swissranger sr3000 and first experiences based on miniaturized 3d-tof cameras. Technical report, CSEM, 2005.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 1988.

[16] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *PAMI*, 28(9):1372–1384, 2006.

[17] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. In *CVPR*, 2004.

[18] E. B. Sudderth, Michael, W. T. Freeman, and A. S. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *NIPS*, 2004.

[19] T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In *CVPR*, 2007.

[20] Y. Wu and T. S. Huang. Hand modeling, analysis and recognition. *IEEE Signal Proc Mag*, (3):51–60, 2001.

[21] Y. Wu, J. Y. Lin, and T. S. Huang. Capturing natural hand articulation. In *ICCV*, 2001.

[22] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, 2003.