Stochastic Meta-Descent for Tracking Articulated Structures

[1]Matthieu Bray [1]Esther Koller-Meier [2]Nicol N. Schraudolph [1]Luc Van Gool
Swiss Federal Institute of Technology (ETH)
Computer Vision Laboratory
Gloriastrasse 35, 8052 Zürich, Switzerland
[1]{bray,ebmeier,vangool}@vision.ee.ethz.ch, , [2]n@schraudolph.org

Abstract

Recently, an optimization approach for fast visual tracking of articulated structures based on Stochastic Meta-Descent (SMD) [2] has been presented. SMD is a gradient descent with local step size adaptation that combines rapid convergence with excellent scalability. Stochastic sampling helps to avoid local minima in the optimization process. We have extended the SMD algorithm with new features for fast and accurate tracking by adapting the different step sizes between as well as within video frames and by introducing a robust likelihood function which incorporates both depths and surface orientations. A realistic deformable hand model reinforces the accuracy of our tracker. The advantages of the resulting tracker over state-of-the-art methods are corroborated through experiments.

1. Introduction

'Condensation' by Isard *et al.* [3] is a robust and powerful stochastic sampling approach, but it is not suited for tracking articulated structures: dense sampling becomes infeasible for higher-dimensional state spaces. One has either to lower the dimensionality or to devise schemes that succeed with fewer samples. An action-specific dynamic model allows Sidenbladh *et al.* [13] to reduce the number of state parameters, though they still need many samples. Wu *et al.* [15] represent articulations in a lower-dimensional space by a set of linear manifolds, then apply a particle filter with importance sampling. Their algorithm performs optimally when the palm is orthogonal to the camera. Alternatively, one can devise methods that work successfully with a reduced number of samples. Deutscher *et al.* [4] propose a sparser particle filter based on a simulated annealing algorithm able to track an articulated model in a high dimensional space. Sminchisescu *et al.* [14] combine global sampling with local optimization by gradient descent. However, this approach is rather slow. Recently, we have introduced 'Stochastic Meta-Descent' (SMD) [2] into computer vision for 3D hand tracking and have demonstrated its efficiency for optimization in high-dimensional spaces. A cost function is minimized by a stochastic gradient descent, with individual step sizes for each dimension that are adapted by a meta-level gradient descent. SMD can naturally incorporate constraints (e.g. anatomical hard constraints) which other optimization techniques find difficult or costly to deal with. Stochasticity in the evaluation of the cost function increases the chance of getting out of local minima. Rehg *et al.* [12] also used gradient descent to minimize the residual between an observed image and overlapping templates describing an articulated object. By comparison, we use 3D rather than 2D video, deal with 5 rather than 2 fingers, and employ a more sophisticated gradient descent scheme, a requirement also felt by Rehg *et al.* [12] (pp. 616). We extend SMD by adapting the different step sizes between as well as within video frames and by proposing a robust likelihood function which includes both depths and surface orientations. We demonstrate it on 3D hand tracking.

The remainder of the paper is organized as follows: Section 2 introduces our 3D hand model and defines the cost function to minimize. Section 3 describes our extended SMD scheme. In Section 4, the incorporation of constraints is discussed and Section 5 explains the step size adaptation. Sections 6 and 7 present the results and conclusions, respectively.

2. Model & Cost Function

2.1. The Hand Model

We use a deformable hand model (see Figure 1) able to realistically reproduce actual hand shapes. A polygonal skin representation is coupled to an underlying skeleton. A more detailed description can be found in [8]. Constraints of the human hand reduce the model to 30 degrees of freedom (DOFs) as shown in Figure 1. The anatomy of the hand imposes limits on the joint angles. We apply such constraints

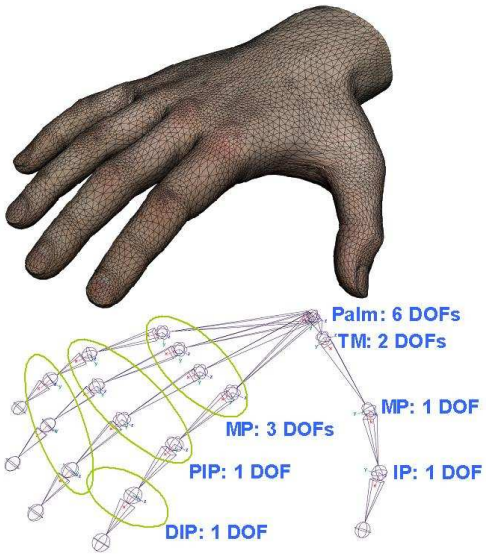


Figure 1. The hand model as polygonal surface (top). The hand model and its degrees of freedom (bottom), where DIP indicates the 'distal interphalangeal' joints, PIP the 'proximal interphalangeal' joints, MP the 'metacarpophalangeal' joints, IP the 'interphalangeal' and TM the 'trapeziometacarpal' joints.

as determined by Lin *et al.* [9] for our tracking purpose. Additional dependencies between the 'proximal interphalangeal' (PIP) and the 'distal interphalangeal' (DIP) angles (DIP = 2/3 PIP) further reduce our model to 26 DOFs.

2.2. Hand Model as a Function

The varying state $\boldsymbol{p}$ of the hand model consists of the translation and rotation of the palm, and the joint angles of the phalanges. This is encoded as a function $\mathcal{F}$ that, for a given $\boldsymbol{p}$ and intrinsic model parameters $\mathcal{M}$, maps a point $\hat{\boldsymbol{x}}_m$ in 3D hand model coordinates to its (predicted) 3D position $\hat{\boldsymbol{x}}_c$ in camera coordinates:

$$\hat{\boldsymbol{x}}_c \;=\; \mathcal{F}(\hat{\boldsymbol{x}}_m, \boldsymbol{p}, \mathcal{M}). \qquad (1)$$

For simplicity, a pseudo-orthographic projection is assumed.

2.3. Cost Function

Tracking proceeds by matching the hand model against depth maps generated by a structured light sensor[1]. Several systems providing such data are now available [5, 7]. We use skin color detection to mask out the background.

1 ShapeSnatcher from Eyetronics (http://www.eyetronics.com/)

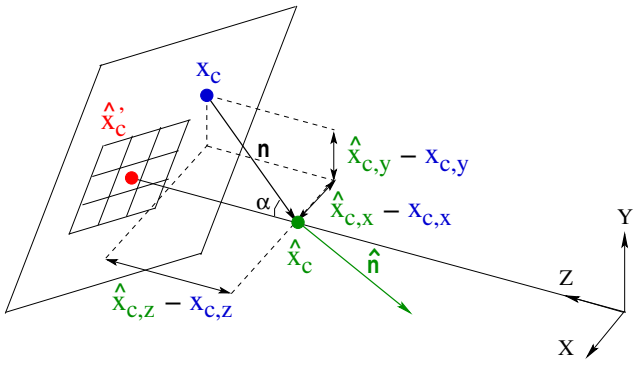


Figure 2. The calculation of the error function. An approximation to the closest point on the surface is calculated by projecting the model point $\hat{x}_c$ onto the tangential plane at $\hat{\boldsymbol{x}}_c^{'} = (\hat{\boldsymbol{x}}_{c,x}, \hat{\boldsymbol{x}}_{c,y}, \mathcal{Z}(\hat{\boldsymbol{x}}_{c,x}, \hat{\boldsymbol{x}}_{c,y}))$. This yields the approximation x_c. $\mathcal{Z}$ is the depth map provided by the 3D sensor in camera coordinates. Furthermore, the difference between the observed normal n and the predicted normal $\hat{n}$ is taken into account.

For a tracked point $\hat{\boldsymbol{x}}_m$ on the hand model, we seek to minimize the distance between its predicted 3D position $\hat{\boldsymbol{x}}_c$ and observed 3D position $\boldsymbol{x}_c$. As we have no information about this corresponding, observed point, we rather minimize – in a way similar to some Iterative Closest Point (ICP) [1] implementations – the distance of $\hat{\boldsymbol{x}}_c$ to the tangent plane at the point on the observed 3D surface with the same image projection, i.e. $\hat{\boldsymbol{x}}_c^{'} = (\hat{\boldsymbol{x}}_{c,x}, \hat{\boldsymbol{x}}_{c,y}, \mathcal{Z}(\hat{\boldsymbol{x}}_{c,x}, \hat{\boldsymbol{x}}_{c,y}))$. This pseudo-orthographic projection is illustrated in Figure 2. Interpolation fills in small holes. Moreover, the difference between the observed normal $\boldsymbol{n}$ and the predicted normal $\hat{\boldsymbol{n}}$ at these points is considered.

If the closest point on this plane is $\boldsymbol{x}_c$, the tracker minimizes the following sum-squared cost function over a sample set S_i:

$$E \;=\; \sum_{S_i} \frac{1}{2} \left(||\hat{\boldsymbol{x}}_c - \boldsymbol{x}_c||^2 + k\,||\hat{\boldsymbol{n}} - \boldsymbol{n}||^2 \right), \qquad (2)$$

where $||\cdot||$ denotes the L_2-norm. A value of $k = 3$ was found to provide a robust cost function and was chosen for reasons of simplicity. Results in Section 6 show that incorporating the normals increases the robustness of the SMD tracker. E is evaluated by considering only a rather limited set of points, 45 in our current implementation, to speed up the tracking.

2.4. Stochastic Sampling

The discrete nature of the sampling process and the noise in the 3D measurements introduce local minima in the optimization function. By randomly changing the set of points where E is evaluated at each iteration step – referred to as 'stochastic sampling' – these spurious minima will also change, lowering the chance for the tracker to get stuck.

3. The SMD Algorithm

Nonlinear optimization problems are often solved by second-order gradient techniques such as the Levenberg-Marquardt algorithm [6, 10] or truncated quasi-Newton methods like the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [11]. These techniques update the model parameters in large steps, each of which is relatively expensive to compute. This makes it hard to enforce constraints on the parameters: the farther a single step takes us outside the feasible region, the more difficult it becomes to return to it. Interior point methods use barrier functions to confine the search to the feasible region which are computationally too expensive for our purposes. Conjugate gradient (CG) techniques [11] are computationally cheaper — linear cost per iteration — but have the drawback that each iteration strongly depends on the preceding ones. Thus CG must be restarted whenever it strays from the feasible region, at a large loss in performance. CG also does not tolerate the noise inherent in our approximation of the gradient by sampling, and converges poorly on highly nonlinear problems.

SMD's strength lies in 3 aspects [2]. Firstly, the use of stochasticity lets the method escape from local minima more easily. Secondly, the use of separate, adaptive step sizes per dimension makes it more efficient in the case of ill-conditioned systems whose energy landscapes show long, narrow valleys. Thirdly, it updates parameters while taking account of the past history of step sizes, and thus is able to capture long-range effects missed by other algorithms. This dampens erratic variations and increases the method's efficiency.

Here we give a concise overview of SMD. More detailed explanations can be found in [2]. If $\boldsymbol{g}_i$ is the gradient (of $E \circ \mathcal{F}$) at iteration step i, i.e.

$$\boldsymbol{g}_i \equiv \frac{\partial E}{\partial \boldsymbol{p}_i} \;=\; \sum_{S_i} \boldsymbol{J}_{\mathcal{F}}^T \boldsymbol{J}_E^T \quad , \tag{3}$$

the parameter vector $\boldsymbol{p}_i$ is updated via

$$\boldsymbol{p}_{i+1} = \boldsymbol{p}_i \;-\; \boldsymbol{a}_i \cdot \boldsymbol{g}_i \,, \tag{4}$$

where $\cdot$ denotes the Hadamard (i.e., component-wise) product, $\boldsymbol{J}_{\mathcal{F}}$ the Jacobian of the function $\mathcal{F}$, $\boldsymbol{J}_E$ the Jacobian of the function E and T the matrix transpose. The vector $\boldsymbol{a}$ of local step sizes is in effect a diagonal conditioner for the gradient system. S_i represents the sample set, which is stochastically changed for each iteration step i.

The local step size vector $\boldsymbol{a}$ is adapted by a scalar meta-step size μ:

$$\boldsymbol{a}_i = \boldsymbol{a}_{i-1} \cdot \max(\tfrac{1}{2}\,,\, 1 \,+\, \mu\, \boldsymbol{v}_i \cdot \boldsymbol{g}_i), \tag{5}$$

where $\boldsymbol{v}$ is an exponential average of the effect of *all* past step sizes on the new parameter values:

$$\boldsymbol{v}_{i+1} \;=\; \lambda\, \boldsymbol{v}_i \;+\; \boldsymbol{a}_i \cdot (\boldsymbol{g}_i - \lambda\, \boldsymbol{H}_i \boldsymbol{v}_i)\,, \tag{6}$$

$$\boldsymbol{H}_i \boldsymbol{v}_i \;\approx\; \sum_{S_i} \boldsymbol{J}_{\mathcal{F}}^T \boldsymbol{H}_E\, \boldsymbol{J}_{\mathcal{F}} \boldsymbol{v}_i\,, \tag{7}$$

where $\boldsymbol{H}_i$ denotes the Hessian (i.e., matrix of second derivatives), or a stochastic approximation thereof, at iteration step i. $\boldsymbol{H}_E$ denotes the Hessian of the cost function E. The factor $0 \leq \lambda \leq 1$ governs the time scale over which long-term dependencies are taken into account.

4. Incorporation of Constraints

The introduction of the hand model constraints into the optimization is most beneficial in the high-dimensional space that we have to explore. We enforce them by means of a function that after each update (4) maps the parameters back into the feasible region:

$$\boldsymbol{p}_{i+1}^c \;=\; \mathrm{constrain}(\boldsymbol{p}_{i+1})\,. \tag{8}$$

Since SMD uses the gradient not only to update the parameter vector $\boldsymbol{p}$, but also to adjust $\boldsymbol{a}$ and $\boldsymbol{v}$, we must somehow make it aware of the constraints on $\boldsymbol{p}$. We do this by calculating a hypothetical 'constrained' gradient $\boldsymbol{g}^c$ which, applied in an unconstrained setting, would cause the same parameter change that we observe after application of the constraints. In other words, we require that

$$\boldsymbol{p}_{i+1}^c = \boldsymbol{p}_i^c \;-\; \boldsymbol{a}_i \cdot \boldsymbol{g}_i^c \Rightarrow \quad \boldsymbol{g}_i^c = \frac{\boldsymbol{p}_i^c - \boldsymbol{p}_{i+1}^c}{\boldsymbol{a}_i}. \tag{9}$$

By using this constrained gradient instead of the usual one in Eq. (6), we can get SMD's step size adaptation machinery to work well for constrained optimization.

5. Step size Adaptation

In a tracking context, SMD has to minimize the cost function several times for subsequent frames. Resetting all the parameters to their initial values would be undesirable. Another extension to SMD is to let the system benefit from experience gathered during the previous frame. As the first step of intra-frame SMD gives information about whether

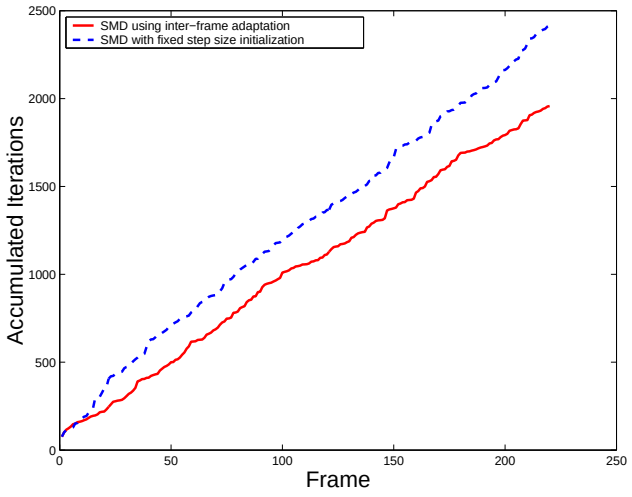


Figure 3. Comparison of the accumulated number of iterations when tracking with or without inter-frame step size adaptation. The solid line is with inter-frame initialisation, the dashed line when step sizes are simply reset at the beginning of each frame.

the different step sizes in $\boldsymbol{a}_0$ are too small or too large and suggests improved step sizes $\boldsymbol{a}_1$ to start intra-frame optimization within that frame, we take these improved values as appropriate initial step sizes $\boldsymbol{a}_0$ for the intra-frame optimization in the next frame. With a good approximation of the initial step sizes, convergence is reached with fewer iterations. Figure 3 shows the advantageous effect of using this information. It compares the accumulated number of iterations with (solid line) and without (dashed line) this initialisation based on the previous frame. Without, the step sizes are reset at each frame to the same values. The inter-frame adaptation accelerates the tracking by **19%**.

This extended version of SMD can be summarised as follows:

- LOOP $t := 1$ TO last frame
  - $\boldsymbol{v}_0 := \boldsymbol{0}$; $\boldsymbol{a}_0 := \boldsymbol{a}^*$; $\boldsymbol{p}_0 := InitialState$
  - LOOP $i := 1$ TO convergence
    1. pick sample points S_i;
    2. calculate $\boldsymbol{g}_i$ (3) and $\boldsymbol{a}_i$ (5);
    - IF $i = 1$ THEN $\boldsymbol{a}^* := \boldsymbol{a}_1$;
    3. calculate $\boldsymbol{p}_{i+1}$ (4),
    4. calculate $\boldsymbol{p}^c_{i+1}$ (8),
    5. calculate $\boldsymbol{g}^{\boldsymbol{c}}_{\boldsymbol{i}}$ (9), $\boldsymbol{H}_i\boldsymbol{v}_i$ (7), $\boldsymbol{v}_{i+1}$ (6).

The initial pose $\boldsymbol{p}_0$ of the hand is determined by manual alignment in the first frame.

6. Results

The SMD tracker's performance is illustrated on 3D hand data, obtained with a structured light sensor yielding depth maps of 720×576 pixels at 12.5 frames per second. The processing was carried out on a Sunfire 1GHz PC. The first experiment presented in Figure 4 highlights the importance of stochasticity. Figure 4 shows the evolution of the SMD tracker, while initialized fairly far from its target, projected on the translation in X and rotation in Z of the palm. As shown, in 7 iterations, the tracker recovers the target. One can notice the landscape of the function changing over time due to the stochastic sampling. The global extremum is naturally always nearly the same, however, the local minimas are changing constantly. The SMD tracker can therefore avoid spurious minimas, increasing the probability to match correctly the target. In comparison, Figure 5 presents for the same sequence, the results obtained by gradient descent and a deterministic sampling approach. The gradient descent method needs 16 iterations to reach an acceptable match but as one can observe, the hand model is shifted from the target providing a less good match than SMD. The deterministic sampling approach, as expected, fails after few iterations. If more points were used to sample the hand model, the deterministic sampling would be less sensitive to local minimas but this would increase the processing time.

Figure 6 illustrates the good performance of the SMD tracker when compared to conventional optimization schemes such as Gradient Descent (GD) [11] or Powell [11], as well as to the state-of-the-art Annealed Particle Filter (APF) [4]. Furthermore, the second row of Figure 6 shows the SMD tracker using a cost function E defined by the sample positions only, i.e. without surface orientation. All methods were running on exactly the same 3D input data and were optimizing the same cost function. The parameters in all methods were chosen carefully in order to optimize their results. For Powell we used the implementations of the VXL package[2]. GD was applied by using SMD and setting μ and λ to zero. For APF we had to use our own implementation. This may not have been maximally optimised for speed, but neither is our current implementation of SMD. The first two rows in Figure 6 show that the incorporation of surface orientations in E increases the robustness of the SMD tracker. With only the position used in E, the tracker gets trapped in a local minimum more easily (in this example the thumb gets stuck to the palm). In this experiment – and several additional, similar experiments – SMD performed best, not only in terms of accuracy but also in terms of computation time. The speeds given in sec/frame for the experiment

2 http://vxl.sourceforge.net/

in Figure 6 are presented in Table 1:

method	SMD	GD	Powell	APF
time [sec/frame]	3	3	232	114

Table 1. Computation time comparison of the Figure 6.

The GD approach is as fast as SMD, but falls in a local minima and is unable to recover. Powell's method is rather slow as it does not use gradient information and it loses the correct solution early on. Besides the SMD algorithm, APF provides the most convincing results, because it is more apt to find the global minimum. But, as shown in Figure 6, in the last frame, the index finger is unable to bend correctly and loses the target. SMD with positional and orientational differences in the cost function seems to offer the best compromise between speed and accuracy.

Figure 7 gives an example with serious self-occlusion, where the SMD tracker does a good job in figuring out the overall rotation that the hand makes. Figure 8 shows examples from an experiment where the word 'FLY' was formed in American Sign Language (ASL). As a matter of fact, the hand pattern formed in Figure 6 corresponds to the letter 'A'. Although SMD shows a rather good performance, we have observed some recurring problems in our overall set of experiments. The most prominent one is that sometimes the thumb tip was still not tracked well due to the paucity of the depth map in that area. The thumb is more prone to error in this regard, because of its wider range of motions and its outspoken independence of the other fingers.

7. Conclusions and Future Work

We have presented a novel SMD tracker which is based on a rapid stochastic gradient descent approach with adaptive step sizes. The inclusion of constraints that can be imposed on the many DOFs of the hand model could be achieved quite elegantly through constrained gradients. We have extended the basic SMD scheme for the purpose of tracking in several ways. As tracking is in fact a series of optimisations for subsequent frames, it was ensured that these would not start from scratch, but would gain robustness and speed by taking over a good initialization for the step sizes from the previous frame. Then, it has been shown that including the surface orientation information in the cost function increases its robustness.

Our experiments show that the SMD tracker performs robustly and efficiently on rather complex 3D hand sequences. Performance came out to be better than that of several, alternative methods running on the same data. The propounded tracker is sufficiently generic to be adapted to different types of input. If the cost function is adapted in an appropriate way, it can use 2D instead of 3D features as well. We are currently extending the scheme to take contour information into account, for instance. Another ongoing improvement is aimed at further increases in robustness. While the stochastic sampling approach helps to overcome local minima, it is not guaranteed that the global optimum will be found. Combining several SMD trackers as 'smart particles' in a Condensation framework is one of the avenues that we are currently exploring.

References

[1] P.J. Besl and H.D. McKay, "A method for registration of 3D shapes," *PAMI*, Vol. 14, pp. 239-256, 1992.

[2] M. Bray, E. Koller-Meier, P. Müller, L. Van Gool and N.N. Schraudolph, "3D Hand Tracking by Rapid Stochastic Gradient Descent using a Skinning Model," *1st European Conference on Visual Media Production (CVMP)*, pp. 59-68, 2004.

[3] M. Isard, A. Blake, "Condensation – conditional density propagation for visual tracking," *IJCV*, Vol. 29, pp. 5-28, 1998.

[4] J. Deutscher, A. Blake, I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," *CVPR*, pp.126-133, 2000.

[5] T.P. Koninckx, A. Griesser and L. Van Gool, "Real-Time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light," *3DIM*, pp. 293-300, 2003.

[6] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *Quarterly Journal of Applied Mathematics, II(2)*, pp. 164-168, 1944.

[7] S. Rusinkiewicz , O. Hall-Holt and M. Levoy, "Real-time 3D model acquisition," *ACM SIGGRAPH*, pp. 438-446, 2002.

[8] J.P. Lewis, M. Cordner and N. Fong, "Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation," *ACM SIGGRAPH*, 2000.

[9] J. Lin, Y. Wu and T.S. Huang, "Modeling Human Hand Constraints," *ARL Federated Laboratory 5th Annual Symposium*, pp. 105-110, 2001.

[10] D.W. Marquardt, "An Algorithm for Least-Squares Estimation of Non-Linear Parameters," *Journal of the Society of Industrial and Applied Mathematics, 11(2)*, pp. 431-441, 1963.

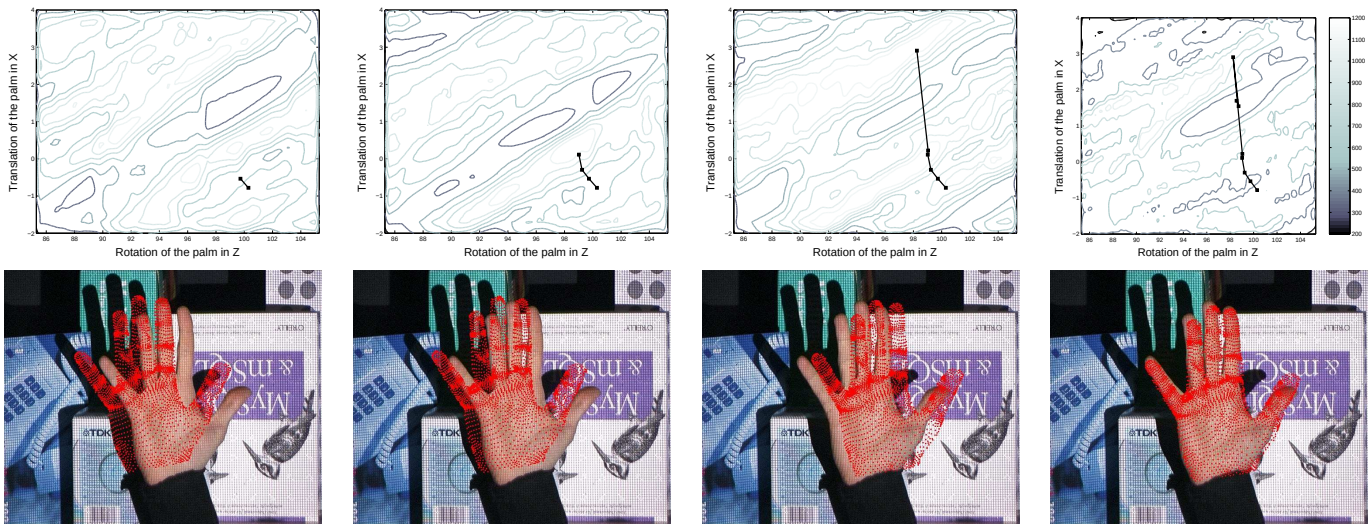
[11] W. H. Press, B. P. Flannery, S. A. Teukolsky, W.T. Vetterling, "Numerical recipes in C," *Cambridge University Press*, 1988.

[12] Rehg J.M. and Kanade T., "Model-Based Tracking of Self-Occluding Articulated Objects," *ICCV*, pp. 612-617, 1995.

[13] H. Sidenbladh, M. J. Black and D. J. Fleet, "Stochastic Tracking of 3D Human figures using 2D Image Motion," *In ECCV*, pp. 702-718, 2000.

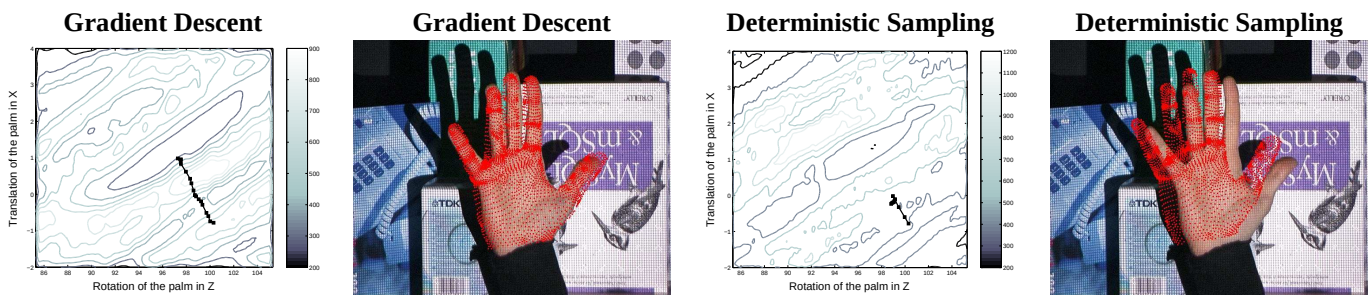
[14] C.Sminchisescu and B.Triggs, "Covariance Scaled Sampling for Monocular 3D Body Tracking," *CVPR*, pp. 447-454, 2001.

[15] Y. Wu, J. Lin and T. S. Huang, "Capturing Natural hand Articulation," *In ICCV*, pp. 426-432, 2001.



Initial state and first iteration | after 3 iterations | after 5 iterations | after 7 iterations

Figure 4. Evolution of the SMD algorithm while tracking. The top row shows this evolution projected on the translation in X and rotation in Z of the palm and the bottom rows presents the resulting sequence.



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

Figure 5. Evolution of the gradient descent algorithm ((a) and (b)) and a deterministic sampling ((c) and (d)) while tracking the same sequence as in Figure 4. Figure (b) is the final result of the gradient descent after 16 iterations: the hand model is slightly shifted from the target. Figure (d) is the final result of the deterministic sampling after 9 iterations: this approach clearly diverges from the correct solution.

SMD including surface orientation: 3 seconds per frame

SMD excluding surface orientation: 3 seconds per frame

Gradient Descent method: 3 seconds per frame

Powell method: 232 seconds per frame

Annealed Particle Filter method: 114 seconds per frame

Figure 6. Comparison between SMD and alternative optimization algorithms. From top to bottom: the results of the SMD using a cost function E evaluating the distances and the differences in surface orientations, SMD using a cost function E only evaluating the distances, Gradient Descent, Powell and APF . The model is visualised as the vertices of the skin polygonal representation. The pictures have been cropped for better visibility, but see Fig. 7 for examples of similar, complete frames.

frame 70 frame 90 frame 103 frame 103

Figure 7. Hand tracking with self-occlusions. Far right: 3D model for frame 103 (top view).

F letter:

L letter:

Y letter:

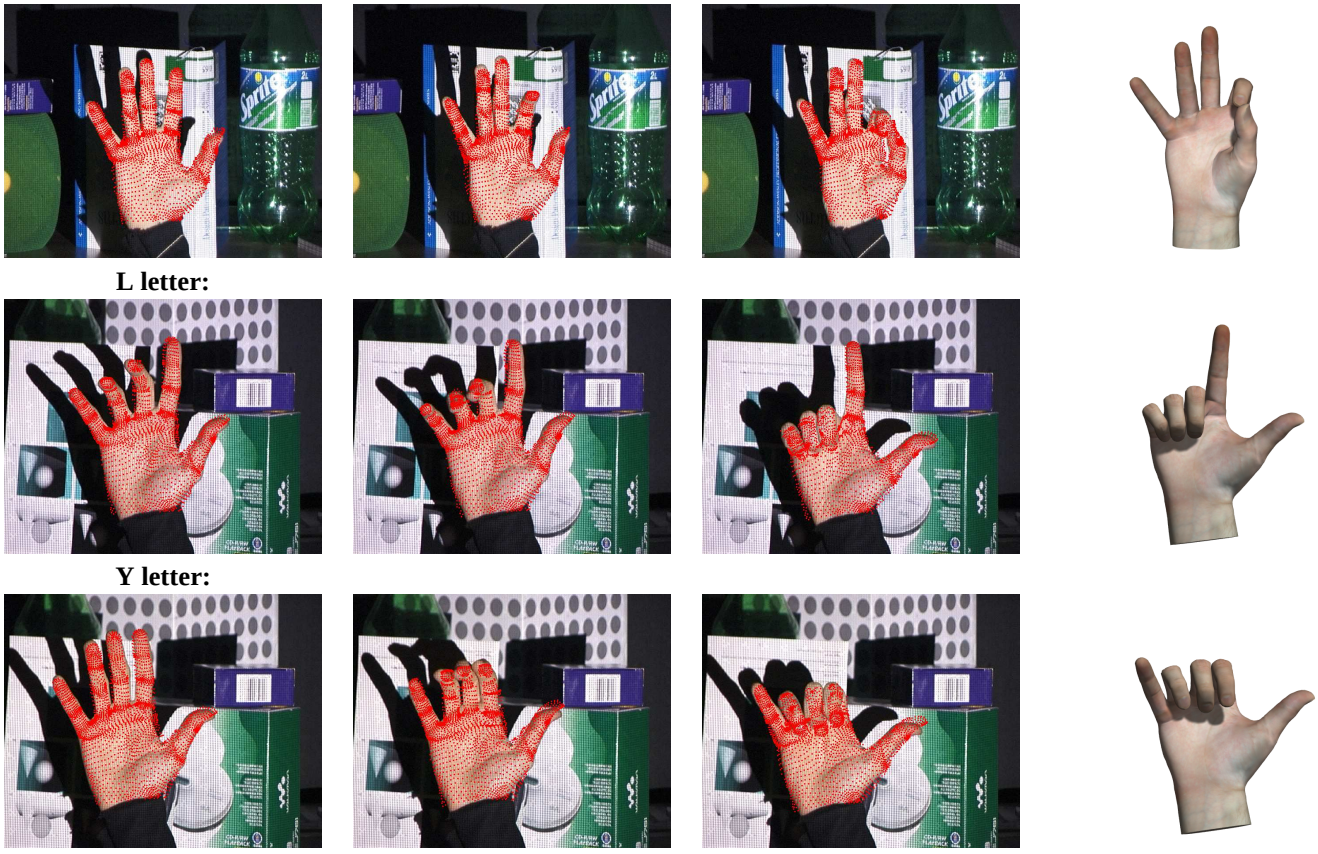


Figure 8. Spelling the word 'FLY' in American Sign Language with SMD: top row represents the 'F' letter, the second row the 'L' letter and the last row the 'Y' letter. The last column is the 3D reconstruction from the third one.