3D Hand Tracking in a Stochastic Approximation Setting

Desmond Chik[1,2], Jochen Trumpf[1,2], and Nicol N. Schraudolph[2,1]

[1] Research School of Information Sciences and Engineering,
Australian National University, Canberra ACT 0200, Australia
[2] Statistical Machine Learning, NICTA, Locked Bag 8001,
Canberra ACT 2601, Australia
desmond.chik@rsise.anu.edu.au, jochen.trumpf@anu.edu.au,
nic.schraudolph@nicta.com.au

Abstract. This paper introduces a hand tracking system with a theoretical proof of convergence. The tracking system follows a model-based approach and uses image-based cues, namely silhouettes and colour constancy. We show that, with the exception of a small set of parameter configurations, the cost function of our tracker has a well-behaved unique minimum. The convergence proof for the tracker relies on the convergence theory in stochastic approximation. We demonstrate that our tracker meets the sufficient conditions for stochastic approximation to hold locally. Experimental results on synthetic images generated from real hand motions show the feasibility of this approach.

1 Introduction

Pose estimation for articulated structures such as the human body and hand is a growing field that has real-world applications. Conceivable uses for such technology range from surveillance, over HCI, to motion capture. There are many image-based 3D tracking approaches to date, often tailored for specific applications, see the surveys [1, 2].

There have been notable works on tracking articulated bodies using monocular images, including [3, 4]. However, depth ambiguities from single images do mean that pose recovery is limited.

Multiple camera views are needed for applications that require a more precise estimation of the body pose. A stereo pair of cameras is enough for depth recovery, but having more cameras reduces ambiguities arising from self-occlusion. Some approaches in multi-view tracking explicitly extract 3D information. For example [5] uses volume reconstruction for a voxel-based fitting process. Silhouette fitting is a popular technique employed by many, *e.g.* [6]. Additional cues are often used to complement silhouette fitting to make the tracking more robust. For example in [7], the reconstruction of a 3D motion field is used to help with the tracking. In [8], motion and spatial cues from images are used to recover displacement in parameter space. In [9], edges, optical flow and shading information from the hand model are used for tracking.

The evaluation of tracking performance is typically based on ground truth sequences. These might be pre-rendered sequences or data retrieved from an alternative source, such as a commercial motion capture system. Whether a tracker can inherently converge to the ideal pose has largely been empirically verified, *e.g.* [10]. A tracker is said to converge to the optimal parameters if the predicted parameters are close enough to the real ground truth values for a particular set of test sequences. Works on the theoretical convergence of a tracker have been lacking in this area. This is understandable as most tracking systems are complex enough to make this task difficult.

The contribution of this paper is to provide a theoretical framework for proving tracker convergence. We present a 3D hand tracking system that has a theoretical proof for convergence. The tracking system is built to be parallelizable and employs stochastic approximation techniques. We will show that the tracking system locally meets the conditions required for stochastic approximation to work, and by that virtue, the results on stochastic convergence from stochastic approximation theory follow.

The paper is organised as follows; Section 2 describes the tracker. Section 3 shows the existence of a unique global minimum for most cases. Section 4 examines the relevance of stochastic approximation for our tracker and introduces the sufficient conditions for stochastic approximation. Proof that these conditions are met locally is also examined. Tracking results are presented in section 5.

2 Tracking System

A stereo pair of images of the hand is acquired by a pair of calibrated cameras. Using these images, our model-based tracking system estimates the 3D pose of the hand in a stochastic approximation framework. Points are sampled from the surface of a fully articulated hand model and projected onto model image planes. By looking at corresponding pixel coordinates in the real images, a cost function based on the hand silhouette and the colour constancy assumption is evaluated. Errors from the cost evaluation are backpropagated as gradients to the parameter space of the hand model. The gradients are then used to minimise the cost function. Figure 1 shows the tracking process.

This paper concentrates on showing that this tracker setup is compatible with the stochastic approximation approach.

2.1 Hand Model

The tracker uses a fully articulated hand model (see figure 2) having 16 joints, totalling 26 degrees of freedom (DOF). There are 6 DOFs at the palm joint, defining the global rotation and translation of the hand. Each digit has 4 DOFs to encapsulate its articulated movement. Rotations at the joints are parameterised with Euler angles. The skin is modelled by a dense mesh (*e.g.* acquired from the 3D scanning of a real hand) and is bound to the underlying skeleton via linear skin blending. Linear skin blending allows sample points taken near the joint regions to deform in a more realistic manner when the joint is bent [11].

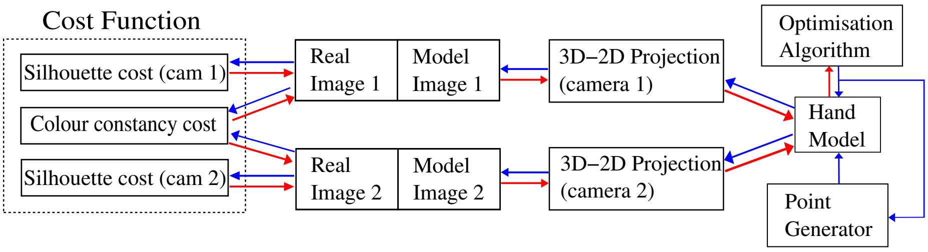

Fig. 1. Flow diagram showing the components making the tracking system.

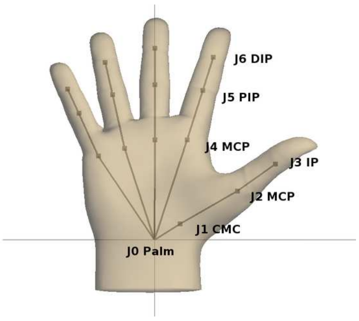

Fig. 2. Deformable hand model used by our tracker.

2.2 3D to 2D Projection Pipeline

This part projects the ith sample point p_i on the hand to a model image plane. Let A_j be the rigid transformation that takes a point in the world coordinates and transforms it to the jth camera coordinates. Let K_j be the calibration matrix of the jth camera. Then $s_{i,j}$, the projection of the ith sample point on the jth image plane is given as

$$s_{i,j} = K_j A_j p_i. \tag{1}$$

2.3 Cost Function

Assigned to each pixel coordinate $s_{i,j}$ is a YUV value $I(s_{i,j}) \in \mathbb{R}^3$ and a silhouette cost value $V(s_{i,j}) \in \mathbb{R}^+$. Note that $s_{i,j}$ depends on x, where x is the vector of the 26 hand parameters. Both I and V are dependent on the set of optimal hand parameters x^* in the sense that x^* determines the real image seen by the cameras. I and V are used in the construction of our cost function $C_{x^*}(x)$.

$C_{x^*}(x)$ comprises of two parts, a silhouette cost function $C_s(i, x)$ and a cost function using the colour constancy assumption $C_c(i, x)$ per sample point i. Let α be a scaling factor for $C_s(i, x)$. Then the overall cost function $C_{x^*}(x)$ we wish to minimise is

$$C_{x^*}(x) = \frac{1}{N}\sum_{i=1}^{N}(\alpha C_s(i,x) + C_c(i,x)), \tag{2}$$

where N is the cardinality of a set of points on the surface of the hand model that is chosen to be sufficiently dense.

Silhouette Cost Function Silhouette information is used as a global constraint on the region which the projected hand model can occupy. Silhouette images are obtained from the real images via background subtraction.

The chamfer 5-7-11 distance transform [12] is then applied over the silhouette image, assigning a distance value V to each pixel based on the pixel's proximity to the closest pixel that belongs to the hand silhouette. The silhouette cost function over j camera views is given by

$$C_s(i,x) = \sum_j V(s_{i,j}), \tag{3}$$

Colour Constancy Cost Function The colour constancy assumption is used for local fine tuning by resolving pose ambiguities in silhouette information. For two camera views, it is given by

$$C_c(i,x) = \frac{1}{2}||I(s_{i,1}) - I(s_{i,2})||^2. \tag{4}$$

Using a 2-norm for the colour constancy cost function and a 1-norm for the silhouette cost function is an attempt to make the overall cost function more robust. When a parameter is far from the optimal value, the silhouette cost function dominates, causing $C_{x^*}(x)$ to behave linearly. When a parameter is close to the optimal value, the colour constancy cost function dominates, and $C_{x^*}(x)$ becomes quadratic. Section 3 will show that C_{x^*} has a unique global minimum for almost all possible values of x^*.

3 A Unique Minimum Exists

A unique global minimum for C_{x^*} does not exist for all x^*, *i.e.* each possible pair of real camera images. As a trivial example, one cannot determine the parameter values of a joint if the joint is occluded in both camera views. However, we will show that for a substantial subset of the possible x^*, there is always a unique global minimum at x^*. We use the term 'substantial' to mean that the exceptions can be described by a finite set of (not necessarily polynomial) equations. Such exception cases will be highlighted. The following assumptions are used to ease the analysis:

1. The y-axes of both camera image planes are parallel to each other, but the x-axes are not.

2. The palm is modelled by a rectangular cuboid and the digits of the hand are modelled by chains of cylinders. Our proof of proposition 1 will rely on a suitable choice of sample points. This choice becomes only easier for a more structured hand model. Hence the proof applies *a fortiori* to our hand model (figure 2).
3. The hand model has a Lambertian surface and has a uniform texture. Hence the YUV value of a point on the hand is completely determined by the surface normal and the light direction.
4. Only one light source illuminating from the front.

We also exclude the aforementioned trivial example in the analysis by assuming that all hand segments are at least partially visible in both camera views.

Proposition 1. *The cost at the optimal position* $C_{x^*}(x^*) = 0$. *Perturbing* x^* *to* $x \neq x^*$ *strictly increases* $C_{x^*}(x)$.[3]

The first part of the proposition is obvious, because at the optimal position, all the sample points lie in the silhouette and each point on the Lambertian surface of the hand model will have the same YUV value seen from different camera views. In the latter part, perturbing x^* will cause certain parts of the hand to move. We denote the points on the hand affected by this perturbation as 'active points'. Figure 3 is the tree of possibilities that can occur for the active points. We now examine each of these cases in detail.

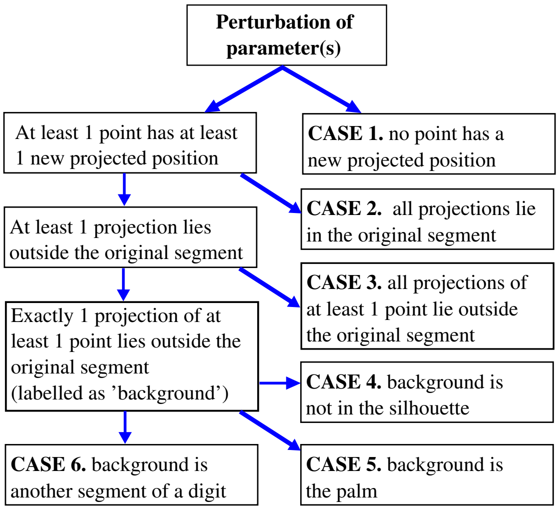
Perturbation of parameter(s)
At least 1 point has at least 1 new projected position
CASE 1. no point has a new projected position
At least 1 projection lies outside the original segment
CASE 2. all projections lie in the original segment
CASE 3. all projections of at least 1 point lie outside the original segment
Exactly 1 projection of at least 1 point lies outside the original segment (labelled as 'background')
CASE 4. background is not in the silhouette
CASE 6. background is another segment of a digit
CASE 5. background is the palm

Fig. 3. Possible scenarios under a perturbation.

[3] Note that proposition 1 does not preclude the existence of other stationary points in the cost function.

3.1 Case 1

Eight points that lie in a non-degenerate configuration in Euclidean space uniquely define the epipolar geometry of the camera pair [13]. Conversely, a known epipolar geometry uniquely defines the projections of eight points that belong to a non-degenerate configuration. Let γ be the set of eight points in a non-degenerate configuration, chosen from the set of active points on a rigid segment of the hand. Then, a perturbation will move at least one of the eight points in γ. Thus, case 1 cannot occur.

3.2 Case 2

We ignore the trivial example of a cylindrical segment rotating around the main cylindrical axis as this type of movement is not possible for the digits of the hand without making the palm rotate, which in turn causes other digits to move outside their original positions.

For a cylindrical segment to lie inside the original region of a given camera view after perturbation, it can only move in a conic region of the plane spanned by the end points of the cylinder to the camera's optical centre. Given that there are two cameras (see figure 4), the intersection of the two conic planes is the only region where movement is allowed.

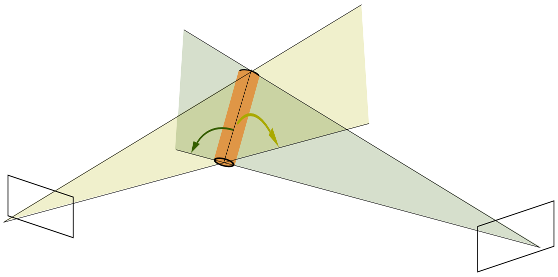

Fig. 4. Foreshortening of a cylindrical segment when the main axis does not lie on the epipolar plane.

This intersection specifies the position of the cylinder uniquely unless the conic regions lie on the same plane, namely the epipolar plane spanned by one end of the cylinder (see figure 5, right). If the cylinder's main axis lies on this plane, then movements on the plane can cause the resulting projection to lie within the original segment for both cameras. For convenience, we shall denote this set of movements as κ.

Pure translational movements on the plane belong to κ (see figure 5, left) only if the projection of the cylindrical segment is longer than the baseline of

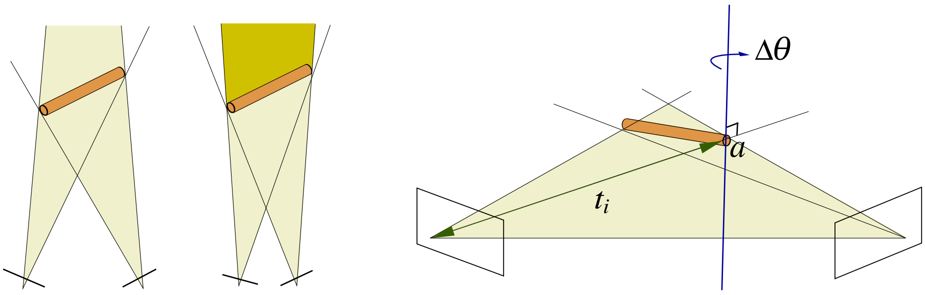

Fig. 5. Left: A cylinder undergoing a pure translation violates the condition for case 2, unless the camera baseline is shorter than the length of the cylinder (*e.g.* 2nd diagram from the left). The dark yellow area indicates the region where the cylinder can translate to. Right: Foreshortening in both cameras when the cylindrical segment rotates on the epipolar plane spanned by a, the centre of rotation.

the camera pair. In our setup, the baseline is much longer than all the segments of the hand, so pure translational movements can be ignored.

A combination of rotational and translational movements on the plane belong to κ if the projection of the cylinder's main axis to the camera image plane is shorter in both cameras after the rotational movement, and the translation movement only moves the perturbed segment within the original region.

Without loss of generality, we take the epipolar plane to be the plane spanned by the x and z axis in the world coordinates. It can be shown that for C_{x^*} not to increase after a rotation, $\triangle\theta$, and a restricted translation, the following equality must hold:

$$\frac{T_1}{T_1 + l\sin\triangle\theta} = \frac{T_2}{T_2 + l\sin\triangle\theta}, \tag{5}$$

where l is the coordinate of a surface point along the major axis of the cylinder. Note that for the ith camera,

$$T_i = t_{i,x}\sin\alpha_i + t_{i,z}\cos\alpha_i, \tag{6}$$

where α_i is the rotation angle and t_i is the translation vector that transforms a point from the local coordinates of cylinder to the coordinates of the ith camera.

For the equality to hold (and therefore C_{x^*} not to increase), either the perturbation is zero (*i.e.* $\triangle\theta = 0$) or $T_1 = T_2$.

Substituting the geometry of camera placement implies

$$\alpha_1 = tan^{-1}(-\frac{D_z}{D_x}) - \theta_r, \tag{7}$$

where θ_r, the rotation angle, and D, the translation vector, are the transformation parameters that convert points from the local coordinates of camera 1 to camera 2. Hence (7) is the only choice for x^* that might not cause C_{x^*} to increase.

The same argument can be applied to the palm, as the palm is attached to the digits, which are cylindrical chains. However, this ambiguity for the palm can only occur if a) the palm and the digits all lie on the epipolar plane or b) all digits of the hand are touching their adjacent digits to form a convex shape. Condition b) ensures that there no gaps between the fingers that would otherwise lead to case 4 when movement occurs on the epipolar plane.

3.3 Case 3

By the continuity argument, one can show that it is not possible to move a hand segment completely off the original segment in both camera views without causing other segments of the hand to partially move from their original position or to leave the silhouette. Therefore one can use the arguments in cases 4, 5 or 6 for the other hand segments to show that C_{x^*} increases.

3.4 Case 4

If one of the active point projections lies outside the original segment and falls outside the silhouette region, then C_{x^*} increases due to the silhouette cost function C_s.

3.5 Case 5

Let p_1, p_2 be the projections of the active point p in the two camera views. p_1 is projected to a surface point s_c on the original cylindrical segment while p_2 is projected onto a surface point s_p on the palm. Suppose the YUV value at p_1 is the same as in p_2, which implies that the surface normal at s_p and s_c are equidistant to the light direction. Then this point will not increase C_{x^*}.

However note that p is chosen from a closed set, and the projection of closed sets remains closed. Therefore the neighbourhood of p will also be projected onto the palm. We can always choose p' from this neighbourhood such that the surface normal at s'_c and s_c are not equidistant to the light direction. Since the palm is modelled as a plane on a cuboid, it has a constant surface normal. Therefore p'_1 will be different to p'_2, and so p' increases C_{x^*}.

The only situation where C_{x^*} does not increase is when the light direction l comes from behind the palm or is orthogonal to the surface normal n_{palm} of the palm, *i.e.* $l \cdot n_{palm} \geq 0$. In this situation, the palm is completely black. This can lead to ambiguity as there always exists a closed set of points with different surface normals on the cylinder that is always black. If p belongs to this set, C_{x^*} will not increase as the neighbourhood of points will also be black. One should note that this situation has been excluded previously in the list of assumptions.

3.6 Case 6

Firstly we assume that none of the active points fall into case 4 or case 5, otherwise we can use those points instead to show that C_{x^*} increases after the perturbation.

Let p_1, p_2 be the projections of the active point p in the two camera views. p_1 lands on a surface point s_c of the original segment c, while p_2 lands on a surface point s_d of another segment d. We first take the simple situation where the main axes of c and d are parallel to the y-axes of both cameras (see figure 6).

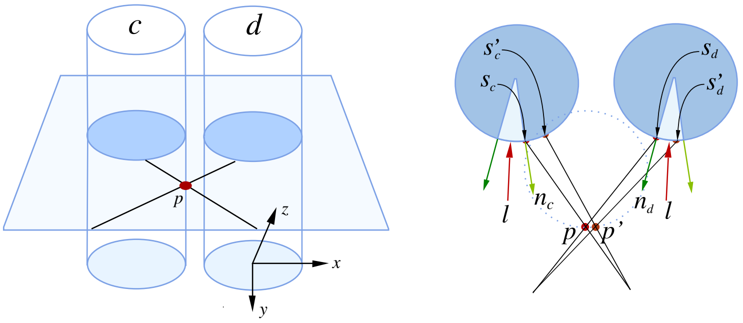

Fig. 6. Left: The setup of the simple situation. Right: The cross-section of the setup. l indicates the projected light direction. The dotted circle indicates the perturbed cylinder. The lighter regions are parts where the YUV value is greater than u.

Assume that p_1 and p_2 (and thus s_c and s_d) have the same YUV value u, which is not black and is not the brightest YUV value on both cylinders. Also assume that the surface normals n_c and n_d at s_c and s_d respectively are different. We know that the light direction is equidistant to n_c and n_d. This produces the lighting pattern as seen in figure 6. Points to the right (anticlockwise) of s_c on the cylinder are darker than u while points to the left are lighter than u. This pattern is reversed on d, where points to the right of s_d are lighter than u.

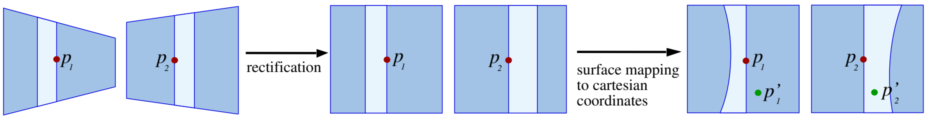

Fig. 7. The projections of the neighbourhood of p after image rectification. The light regions are parts where the YUV value is greater than u.

Suppose we examine a square neighbourhood of p in 3D space, visible in both cameras (see figure 7). Its projection is an affine transformation of the square, and is different in both cameras. Suppose the two views are rectified. Note that the bounding lines marking the lighter region do not cross over each other after rectification. If we account for the fact that one can only sample on the neighbourhood of p in 3D space that belongs to the surface of the perturbed cylinder, then these bounding lines become curves after rectification of the surface to cartesian coordinates. As seen in figure 7, one can always choose a point p' in the final rectified neighbourhood such that p'_1 lands on the lighter region while p'_2 lands on the darker region or vice versa. Hence C_{x^*} will increase.

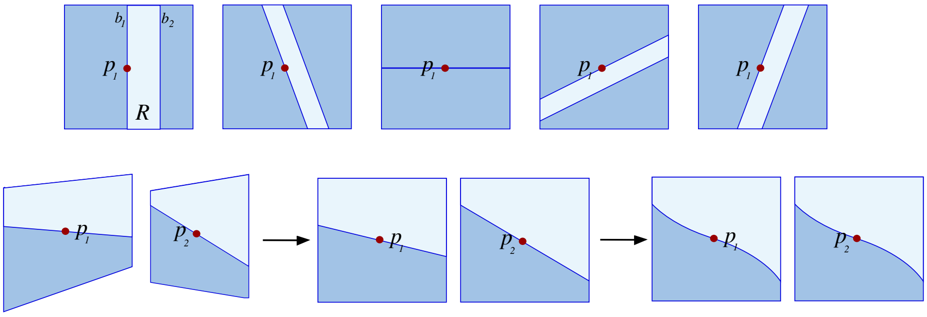

Fig. 8. Top: R varying upon different z-rotations. Note the degenerate (3rd) case. Bottom: Ambiguity is possible if the bounding line matches exactly after rectification.

We now generalise to other cylinder configurations. For convenience, we use c as an example, and denote the band of lighter region as R, the light direction as l, the bounding line of R that contains s_c (or p_1 in the projection) as b_1 and the other bounding line in the same image as b_2. Rotation of the cylinder around the z-axis changes the slope of the bounding lines and narrows R as the n_c becomes more aligned to the light direction. Degeneracy D occurs when n_c is the surface normal on c that is closest aligned to the light direction. Then R does not exist. Note that the light direction determines the particular z-rotation that results in this degeneracy. As n_c rotates away from the light source, R reappears but b_2 now flips to the other side of b_1 (see figure 8).

A rotation on the x-axis determines the brightest value on the cylinder and changes the slope of the bounding lines seen in the projection. However the surface area of R in 3D space remains the same. Degeneracy occurs when an x-rotation causes $n_c \cdot l \geq 0$. Then s_c and its neighbourhood will be black.

Additionally, given that the projection of the neighbourhood of p is an affine transformation, it may be possible to construct cases where b_1 of one image aligns with b_1 of the other image after rectification (see figure 8), creating ambiguity. Ambiguity also arises when $n_c = n_d$ or when D occurs in both cylinders. Then whether one can choose a p' that causes C_{x^*} to increase will depend on the rate of curvature change on c, d and the perturbed cylinder. It also depends on the camera placement. Other than these degenerate cases, one can always choose p' such that p'_1 lands on R and p'_2 lands outside of R or vice versa. Note that these degenerate cases are rare.

4 Stochastic Approximation

Noise is a highly noticeable process and occurs at various parts of the tracking system. For example there is measurement noise from the camera and discretisation noise in the evaluation of image gradients that are computed using Sobel masks. Also, we are only sampling a small subset of $n \ll N$ points from the hand model to evaluate an approximation of the true cost function. This introduces sampling noise. For a tracking system to perform adequately, the effect of noise must be addressed and minimised.

Stochastic approximation [14, 15] is a technique for finding the root of a function $f(x)$ where only noise-corrupted measurements of function values are available. This can be applied to an optimisation setting like the one in our tracker if we set $f(x)$ to be the gradient of our cost function $C_{x^*}(x)$. Then finding the root of f equates to finding the critical point (the minimum) of C_{x^*}.

Let the random variable $Y_t(x)$ be the noisy observation of $f(x_t)$, *i.e.* the gradient of $C_{x^*}(x_t)$ in parameter space, and a_t be the step size in our optimisation procedure. Then a possible iterative scheme for stochastic approximation is

$$X_{t+1} = X_t - a_t Y_t. \tag{8}$$

This is the so-called Robbins-Monro method [14, 15]. Note that X_t is a random variable and x_t is the actual event of X_t at time t. X_t of (8) converges to x^* in mean square and with probability 1, *i.e.*:

$$\lim_{t\to\infty} E[(X_t - x^*)^2] = 0 \quad \text{and} \quad P(\lim_{t\to\infty} X_t = x^*) = 1, \tag{9}$$

if the following conditions are met:

1. A bound on the step size a_t,

$$\sum_{t=1}^{\infty} a_t = \infty, \sum_{t=1}^{\infty} a_t^2 < \infty. \tag{10}$$

2. Y_t is unbiased,

$$E(Y_t) = f(x_t). \tag{11}$$

3. Y_t has uniformly bounded variance in the sense,

$$\sup \{Var(Y(x)) : x \in \mathbb{R}^K\} < \infty. \tag{12}$$

4. f is well-behaved around x^* in the sense

$$\inf \{(x - x^*)^T f(x) : \epsilon < ||x - x^*|| < \epsilon^{-1}\} > 0, \tag{13}$$

for all $\epsilon \in \mathbb{R}, 0 < \epsilon < 1$.

The following subsections will demonstrate how our hand tracking system meets these requirements.

In the classical Robbins-Monro scheme [14], the step size a_t of the iterate update (8) is set to $\frac{q}{t}$ for some constant q. In practice, this method may not be desirable due to its slow convergence rate.

In practice, we use the SMD algorithm [16, 17] which tends to converge much faster. Although there is no convergence proof available yet for SMD, it has been shown empirically to work well under noisy conditions in many practical situations. Prior applications of SMD to body/hand tracking work include [18, 5]. The following arguments are independent of the chosen optimisation algorithm.

4.1 Y_i is an Unbiased Estimator

Noise from image gradients is unbiased since the Sobel mask used for calculating image gradients is centred and symmetric. Bias due to camera noise is minimised as the cameras are calibrated prior to use.

In addition, bias in Y_i heavily depends on the sampling scheme used to evaluate the gradient estimates. We are aware that our way of proving condition (13) in section 4.3 potentially introduces certain requirements/constraints on the sampling scheme, thereby producing bias. However, this is easily mitigated by taking more sample points.

4.2 Y_i has Uniformly Bounded Variance

The observed gradients Y_i generated by the cost function have a bounded variance. For the silhouette cost function, the variance in distance estimation is uniformly bounded by the size of the image. Therefore the gradient estimates generated by it via finite differences will also be bounded. The variance in Y_i due to the colour constancy cost function is also uniformly bounded since the range of YUV values at each pixel is uniformly bounded.

4.3 $f = \bigtriangledown C_{x^*}$ is Well-behaved

Condition (13) does not hold globally for all x^*, but we can show it holds locally for most x^*. To satisfy (13), it is sufficient that f has a zero root at x^* and that the Jacobian J_f of f (*i.e.* the Hessian of C_{x^*}) is positive definite (*i.e.* our cost function C_{x^*} has a strict local minimum at x^*). The former part was shown in section 3. We now show that J_f is positive definite.

The tracker can be viewed as a composite function $C \circ M$, where C is the cost function and M the remaining parts of the tracker. Because $C_{x^*}(x^*) = 0$ (proposition 1), the Hessian H of $C \circ M$ (or the J_f of f) at the minimum can be rewritten [17] as

$$J_f = H = \frac{1}{N} \sum_{i=1}^{N} J_{M,i}^T H_c J_{M,i}, \qquad (14)$$

where H_c is the Hessian of the cost function and $J_{M,i}$ is the Jacobian of M at the ith sample point.

H_c is the sum of the Hessian H_{c_s} of the silhouette cost function and the Hessian H_{c_c} of the colour constancy cost function. H_{c_c} is given as,

$$H_{C_c} = \begin{pmatrix} I_{3\times3} & -I_{3\times3} \\ -I_{3\times3} & I_{3\times3} \end{pmatrix}. \tag{15}$$

H_{c_s} can either be positive definite or zero at the minimum; the latter due to ambiguity for certain poses of the silhouette. We can ignore the H_{c_s} terms as they cannot decrease the rank of H. It is sufficient to show that J_f attains full rank due to H_{c_c} alone.

J_f is at least positive semi-definite since the summands in (14) are positive semi-definite. Also the rank of J_f is non-decreasing when adding samples since the summands are added.

As an empirical verification, each frame in the test video sequence was tested to see if taking an adequate amount of sample points led to the J_f estimate achieving full rank at the minimum. On average, approximately 100 points were required for the J_f estimate to achieve full rank.

5 Tracking Results

The tracker has been tested over a short video sequence of 60 frames (640 × 480 pixels) that shows the hand extending to grip an imaginary object (figure 9). The sequence contains elements of lateral translation, wrist rotation, and articulated motion of the digits in the form of gripping. To obtain a ground truth assessment of the tracker accuracy, the hand model is initially fitted frame by frame, by eye over the real captured sequence. The parameter values obtained from this procedure are then taken to be the ground truth. Using these parameter values, a synthetic sequence is rendered using OpenGL. Tracking performance is evaluated by running the tracker on the synthetic sequence over $G = 50$ trials. The experiment was conducted on a P4 3.4GHz machine.

Approximately $n = 280$ active sample points were used to track the moving hand. The optimisation algorithm was allowed to perform a maximum of 50 iterations per frame. In terms of computational speed, an average of 1.4s were required to track one frame, of which 0.8s were required for image preprocessing such as extracting silhouette and calculating image gradients. The remaining 0.6s were spent by the iterations of the optimisation procedure. The code for the tracker has not been optimised and the parallelizable structure of the system has not been exploited.

The error measures used to evaluate performance are based on the difference in distance between actual and predicted joint positions in Euclidean space. The first error measure used is the overall mean error. Let $p_{k,g}$ and a_k be the predicted and actual 3D positions of the kth joint for the gth trial. Then the overall mean error is given as

$$\frac{1}{GK}\sum_{g}^{G}\sum_{k}^{K}||a_k - p_{k,g}||, \tag{16}$$

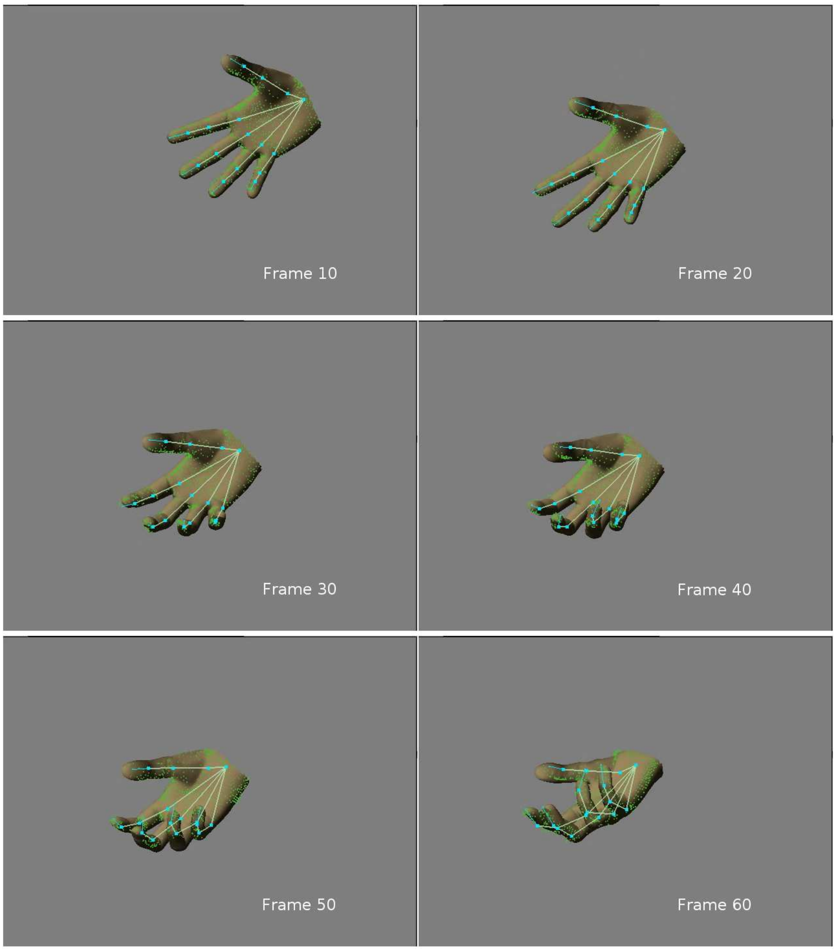

Fig. 9. Tracking results at every 10th video frame

where K is the total number of joints in the hand model. Figure 10 (left) shows the tracking accuracy over the test sequence. The overall mean error is given in Euclidean and image space.

To put the error measurements into perspective, the hand model in a relaxed open palm position can be roughly bounded by a $180 \times 100 \times 30mm$ cuboid and is located approximately $1m$ from the cameras (roughly bounded by a rectangle of 220×150 pixels in image space). The cameras are pointed towards the hand in a convergent setup, at an angle of $30°$ from each other. The baseline between the two cameras is $0.85m$.

To classify whether the tracker has irrecoverably lost track of the hand, we introduce another measure. A tracker is classified as having passed the tracking

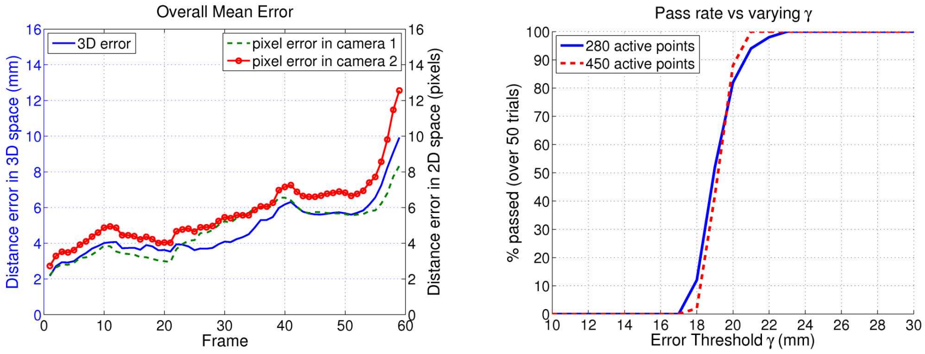

Fig. 10. Left: Overall mean error of the video frames over time. Right: Pass rates for varying γ values.

sequence if during the entire sequence, the error distance between predicted and actual position of each joint is below an error threshold γ. Figure 10 (right) plots the results for varying γ. The figure also shows that increasing the number of active sample points improves the pass rate.

6 Conclusion

A 3D hand tracking system has been presented that uses silhouette and the colour constancy assumption. A theoretical proof of local stochastic convergence has been provided for the tracker. It shows that except for certain degenerate hand pose configurations, local stochastic convergence holds. It is possible that such a system can be generalised to multiple cameras or to track other articulated structures such as the human body. Experimental results on synthetic images are promising, although we believe that a more sophisticated sampling scheme will improve tracking accuracy.

Acknowledgements

We wish to thank the BIWI laboratory, ETH Zürich for permitting the use of their hand model.
NICTA is funded by the Australian Government's Department of Communications, Information Technology and the Arts (DCITA) and the Australian Research Council (ARC) through Backing Australia's Ability and the ICT Centre of Excellence program.

References

1. Erol, A., Bebis, G.N., Nicolescu, M., Boyle, R.D., Twombly, X.: A review on vision-based full DOF hand motion estimation. In: Vision for Human-Computer Interaction. (2005) III: 75–75

2. Wang, L., Hu, W.M., Tan, T.N.: Recent developments in human motion analysis. Pattern Recognition **36**(3) (2003) 585–601
3. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3D body tracking. In: CVPR. (2001) I:447–454
4. Sudderth, E.B., Mandel, M.I., Freeman, W.T., Willsky, A.S.: Visual hand tracking using nonparametric belief propagation. In: Workshop on Generative Model Based Vision. (2004) 189
5. Kehl, R., Gool, L.J.V.: Markerless tracking of complex human motions from multiple views. Computer Vision and Image Understanding **103**(2-3) (2006) 190–209
6. Carranza, J., Theobalt, C., Magnor, M., Seidel, H.: Freeviewpoint video of human actors. In: ACM Transactions on Graphics, San Diego, USA, ACM SIGGRAPH (2003) 569–577
7. Theobalt, C., Carranza, J., Magnor, M.A., Seidel, H.P.: Combining 3d flow fields with silhouette-based human motion capture for immersive video. Graph. Models **66**(6) (2004) 333–351
8. Sundaresan, A., Chellappa, R.: Multi-camera tracking of articulated human motion using motion and shape cues. In: ACCV. (2006) II:131–140
9. Lu, S., Metaxas, D., Samaras, D., Oliensis, J.: Using multiple cues for hand tracking and model refinement. In: IEEE Computer Vision and Pattern Recognition or CVPR. (2003) II: 443–450
10. Balan, A.O., Sigal, L., Black, M.J.: A quantitative evaluation of video-based 3D person tracking. In: International Workshop on Performance Evaluation of Tracking and Surveillance. (2005) 349–356
11. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: SIGGRAPH '00, New York, NY, USA (2000) 165–172
12. Borgefors, G.: Distance transformations in digital images. Comput. Vision Graph. Image Process. **34**(3) (1986) 344–371
13. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edn. Cambridge University Press (2004)
14. Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics **22** (1951) 400–407
15. Blum, J.R.: Multidimensional stochastic approximation methods. Annals of Mathematical Statistics **25** (1954) 737–744
16. Schraudolph, N.N.: Local gain adaptation in stochastic gradient descent. In: ICANN, Edinburgh, Scotland, IEE, London (1999) 569–574
17. Schraudolph, N.N.: Fast curvature matrix-vector products for second-order gradient descent. Neural Computation **14**(7) (2002) 1723–1738
18. Bray, M., Koller-Meier, E., Müller, P., Schraudolph, N.N., Gool, L.V.: Stochastic Optimization for High-Dimensional Tracking in Dense Range Maps. IEE Proceedings Vision, Image & Signal Processing **152**(4) (2005) 501–512