A Virtual Keyboard System Based on Multi-Level Feature Matching

Huan Du, and Edoardo Charbon
Ecole Polytechnique Fédérale, Lausanne, Switzerland
huan.du@epfl.ch, edoardo.charbon@epfl.ch

Abstract — **In this paper a Multi-Level Feature Matching (MLFM) method is presented for 3D hand posture reconstruction of a virtual keyboard system. The human hand is modeled with a mixture of different levels of detail, from skeletal to polygonal surface representation. Different types of features are extracted and paired with the corresponding model. The matching is performed in a bottom-up order by SCG optimization with respect to the state vector of motion parameters. The low level of matching provide initial guess to the high level of matching, refining the precise position of the hand hierarchically. The matching results show that this method is effective for tracking human hand typing motion, even with noisy 3D depth map reconstruction and roughly detected fingertips. Examples of applications include virtual reality, gaming, 3D design, etc.**

Keywords — **Virtual Keyboard System, 3D Hand Model, 3D Hand Tracking, Feature Matching**

I. Introduction

As the demand of ubiquitous computing thrives, the human-computer interaction (HCI) issue has become very significant. Ordinary keyboards however, are limited in conveying complex or multi-dimensional information. Virtual keyboard systems are proposed as a new generation of HCI devices and paradigms. A virtual keyboard is known as a touch-typing device that does not have a physical manifestation of the sensing area, that is, the sensing area which acts as a button is not per se a button but instead is programmed to act as one [1]. It has many applications in human-computer input, virtual reality, game control, 3D designs, etc.

To date, there exists a number of virtual HCI implementations reaching various levels of sophistication. Examples of such systems are touch-pads, miniaturized keypads, cyber-gloves [2] and pressure-sensitive bands [3], to name a few. However, in all these cases complex pick-up devices, add-ons, or surgical implants are always required, making the HCI system expensive, inconvenient and less attractive. On the other hand, vision-based devices are less intrusive to human users, and provide fairly high flexibility and accuracy for both implementation and application. Due to the inherent complexity of capturing and interacting with 3D entities, the trend is to move to zero-form-factor approaches, involving full 3D methods for both sensing and tracking purposes.

Tracking highly articulated structures, like hands, is a nonlinear search problem in a high dimensional space. The existence of many local optima and the requirement of massive measurements make the hand tracking a big challenge in computer vision for over a decade. To mitigate the problems caused by ambiguities, occlusions, and image measurement noise, sophisticated modeling methods and feature extraction techniques have been introduced into the human hand tracking [4] - [6]. However, due to high-dimensional variability and nonlinearity of human dynamics, tracking complex human motion, such as hand typing motion, is still challenging.

Two general techniques have been proposed for visual hand tracking. One is an appearance-based method, and the other is a model-based method. The appearance-based method establishes mapping between image features and hand poses and directly estimates hand postures from the images. Rosales [7] mapped the low level visual features to hand joint configuration, with a supervised learning framework for training the mapping function. Stenger et al. [8] proposed an effective indexing framework for Bayesian tracking based on the tree representation of a large database of synthetic hand images. The major advantage of using appearance-based methods is the simplicity of their parameter computation for the temporal postures. However, the mapping may not be one-to-one, and the loss of precise spatial information makes it especially less suited for hand position reconstruction.

Alternatively, model-based methods project a deformable 2D / 3D hand model to the image space and match it with the observed image features. Rehg and Kanade [9] introduced a highly articulated 3D hand model in their DigitEyes hand tracking system. Lee and Kunii [10] employed the skeletal model to simulate the human hand in the real image. Bray et al. [11] constructed a skinning model driven by the underlying skeleton system and used it in 3D hand tracking with a rapid stochastic gradient descent optimization framework. De La Gorce and Paragios [12] introduced a 3D hand model with mixture of ellipsoids and deformable polyhedrons for 2D projection and silhouette computation. They also applied motion constraints and kinematic motion constraints into the optimization framework. A realistic 3D model of the human hand has at least 26 degrees of freedom (DoFs). Tracking based on 3D hand model is then formulated as a nonlinear search problem in a high dimensional space. The search for good solutions in large spaces and with high dimensional data is

hard to tackle and can cause immense computational costs.

We propose a virtual keyboard system using 3D hand tracking scheme based on Multi-Level Feature Matching method. The hand is modeled by a hierarchical representation with different levels of detail. The matching is carried out in a bottom-up order, that the lower level of matching provides a rough estimation of the hand posture to the adjacent higher level of matching as its initial guess. In each level of matching, a certain type of image feature is extracted and matched against its correspondence in that level of hand representation. The sketch of our system is presented in Fig. 1. The matching step implements an SCG optimizer based on the state vector of motion parameters. The details of the tracking steps are described in the following parts.

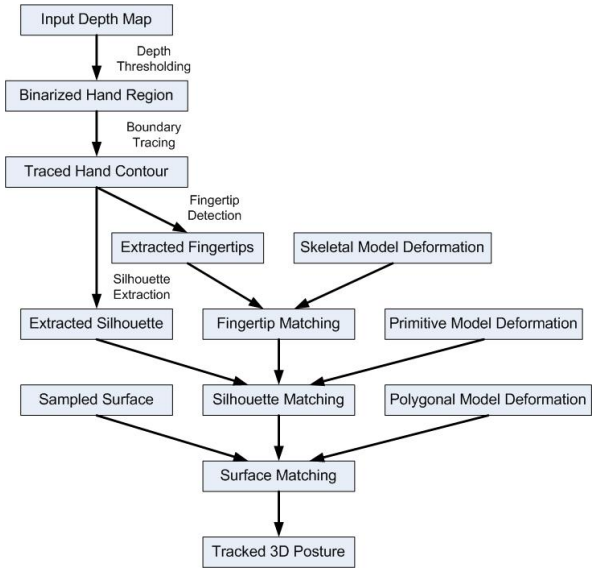
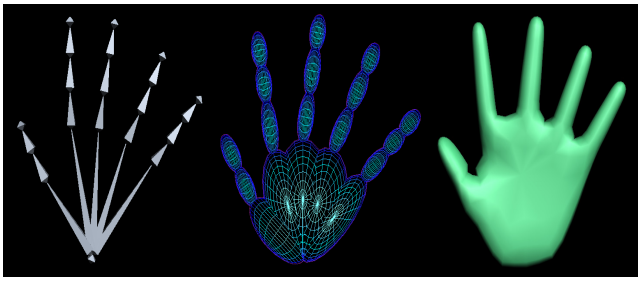


Fig. 1. Overview of our Multi-Level Feature Matching framework

II. 3D TRACKING CONFIGURATION

A. Hand Model Representation

Different 3D hand models have been proposed in the literature. Either simple articulated structure as stick figure [13], or primitive models as conics and convex polyhedron [14], or complex volumetric models as triangulated surface [15] have been adopted. We consider a mixture of hand models that represents the hand in different levels of detail. This mixture model contains three levels of representation: the skeletal level, the ellipsoidal-primitive level, and the surface level, also driven by the skeletal system, a detailed polygonal-surface level. In the skeletal level, similar to the ones used in the inverse kinematic systems, a 3D skeletal hand model is employed to simulate the human hand dynamics. In the ellipsoidal-primitive level, each phalange is modeled as an ellipsoid primitive and driven by the underlying skeletal system. Finally, in the polygonal-deformable polygonal surface is used to represent the human hand skin. The 3D illustration of the hand model of different levels is shown in Fig. 2.



(a) (b) (c)

Fig. 2. Multi-level hand representation: (a) skeletal level (b) ellipsoidal-primitive level (c) polygonal-surface level

The skeletal representation supports both the global hand motion as the translational and rotational movement of the palm, and the local finger motions as the bending and twisting of the joints. The base is the wrist and the palm is modeled as five metacarpal bones, each with a finger attached to it. The constraints of the human hand motion reduce the model to 30 DoFs: one DoF (extension / flexion) for each distal interphalangeal (DIP), interphalangeal (IP) and proximal interphalangeal (PIP) joints, two DoFs (extension / flexion and adduction / abduction) for each metacapophalangeal (MCP) joints except for the thumb (one DoF for extension / flexion), and one DoF (twist) for each trapeziometacarpal (TM) joints except for the thumb (two DoFs for adduction / abduction and twist). The palm has six DoFs for the wrist's translational and rotational movement. All the three levels of representation share the same parameter space of 30 DoFs.

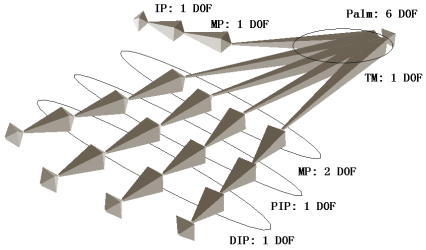


Fig. 3. The 3D representation of our deformable skeletal hand model

B. Hand Model Deformation

The basic skeletal dynamics is formulated using an open kinematic chain. Let $\mathbf{M}_{i-1}^{i}$ be the transformation matrix from joint *i* to joint *i-1*, which is used to describe the translational and rotational relationship between the adjacent links of the kinematic system, and is of the form

$$\mathbf{M}_{i-1}^{i} = \mathbf{R}_{i-1}^{i}\mathbf{T}_{i-1}^{i} \tag{1}$$

where $\mathbf{T}_{i-1}^{i}$ is the homogeneous translation matrix and $\mathbf{R}_{i-1}^{i}$ is the homogeneous rotation matrix from joint *i* to joint *i-1*.

By multiplying all the transformation matrices that correspond to the preceding joints in the hand hierarchy, the transformation matrix of joint *n* to the global coordinate system joint *0* can be denoted as

$$\mathbf{M}_{n} = \mathbf{M}_{0}^{1}\mathbf{M}_{1}^{2}\mathbf{M}_{2}^{3}...\mathbf{M}_{n-1}^{n} \tag{2}$$

In 3D space, the quadric surface of the ellipsoidal-primitive model can be represented in homogeneous coordinates as a symmetric 4×4 matrix **Q** such that

$$\mathbf{X}^{\mathbf{T}}\mathbf{Q}\mathbf{X} = \mathbf{0} \tag{3}$$

The matrix **Q** is of the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^{\mathbf{T}} & c \end{bmatrix} = (\mathbf{C}^{-1})^{\mathbf{T}} \cdot (\mathbf{M}^{-1})^{\mathbf{T}} \cdot \mathbf{L} \cdot \mathbf{M}^{-1} \cdot \mathbf{C}^{-1} \tag{4}$$

where **M** is the transformation matrix of the joint attached to, **C** is the translational matrix and **L** is the scaling matrix along the principle axes of the local coordinate system of the joint. The matrix **C** and **L** are of the form:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} 1/l_x^{\,2} & 0 & 0 & 0 \\ 0 & 1/l_y^{\,2} & 0 & 0 \\ 0 & 0 & 1/l_z^{\,2} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{5}$$

Here (c_x, c_y, c_z) is the ellipsoid primitive's center in the local coordinate system, and (l_x, l_y, l_z) are the radii of the ellipsoid. For any configuration of hand posture, we can first compute the transformation matrix **M** of all the joints using (1) and (2), and then update the ellipsoidal surface by finding points **X** satisfying (3).

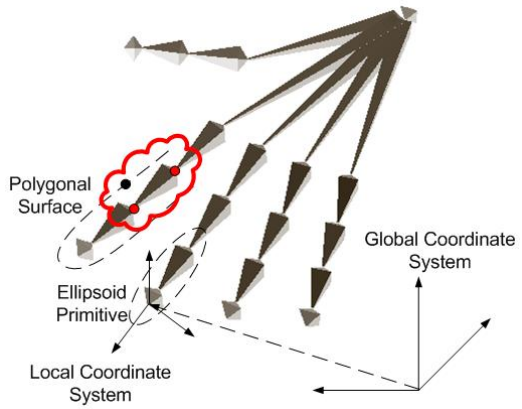


Fig. 4. Transformation of multi-level models based on skeletal representation parameters

For polygonal-surface model, each vertex of the surface is coupled to a subset of the skeleton joints. During the deformation of the kinematics, the shape of the polygonal surface is updated by linearly blending the new positions of each vertex [16]. Denote γ_i as the blending weight and $\mathbf{v}_0$ as the initial position of the vertex **v**, its current position $\mathbf{v}_c$ can be given as

$$\mathbf{v}_c = \sum_{i=1}^{n} \gamma_i \mathbf{M}_i \mathbf{M}_{i0}^{-1} \mathbf{v}_0 \tag{6}$$

where $\mathbf{M}_i$ is the transformation matrix of the *i-th* joint in current posture and $\mathbf{M}_{i0}$ is of the same joint in initial posture.

C. 3D Ranging

Optical depth map reconstruction may be performed using one of three main techniques: triangulation, time-of-flight (ToF) and stereo vision. Triangulation based techniques achieve millimeter depth accuracy at a cost of high power dissipation and computational complexity. Moreover high 3D framerate at high depth precision are generally difficult to achieve. Modulation type ToF rangefinders [17] require relatively powerful laser or LED sources, and accuracy is limited by the speed at which the sensor can be clocked. The stereo vision approach [18] simulates how human vision manages to perceive objects in depth, and derives 3D structures by comparing the images acquired with different cameras. The 3D ranging device used in our virtual keyboard system is very similar to the structured light system proposed by Forster [19]. The color-encoded stripe pattern is projected onto the target scene and captured by the pre-calibrated camera. Based on the deformation of the stripes and the decoded position information, the projector-camera con-figuration acts as a stereo-vision framework but has much higher precision than the multi-camera configurations, which is based on image feature detection and registration. The reconstructed 3D depth map has high depth precision, large image resolution and moderate framerate.

III. Feature Extraction

Our feature extraction process is based on noisy 3D depth map of human hands. The depth map is generated using the structured light system. It is first binarized by applying a certain threshold to remove the background objects and keep only the hand region, and then projected back to 2D image plane for further processing. After the background subtraction and binarization, the boundary of the hand is traced by denoting the pixels within the hand region that have 8-connectivity neighborhood with the non-hand region. Since our proposed matching method is based on multi-level features and models, here we need to extract different types of feature for different levels of model, as shown below.

A. Fingertip

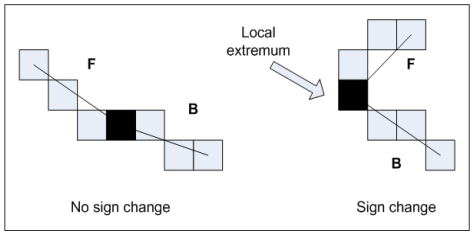


Fig. 5. Fingertip detection by finding local maxima of k-curvature along the hand contour

Fingertips are presented in the 2D image as the peaks of the hand contour. Fingertip detection in the 2D image can be formulated as finding the local maxima of the k-curvature for each pixel on the boundary. K-curvature is defined as the angle between two vectors $[P_{i-k}, P_i]$ and $[P_i, P_{i+k}]$, where k is a constant and $P_i = (x_i, y_i)$ is the coordinate of the contour pixel. On the other hand, the interdigital clefts are shown as the valleys of the contour and can be detected by finding the local minima of the k-curvature. K-curvature computation can be simplified by using the approximation of sign change [20], as shown in

Fig. 5. After the fingertips are detected in 2D image plane, they are mapped back into the 3D space using the depth map.

Applying the transformation matrix in (2), the fingertips of the skeletal model are transformed as the end-effectors of the kinematic chain to the global coordinate system. These synthesized fingertips are then fitted to the corresponding ones detected in the depth map measurement, with the matching approach explained in Section IV.

B. Finger Silhouette

Using the method introduced above, we can detect not only the fingertips as the peaks, but also the interdigital clefts as the valleys of the hand contour. Based on the relative order of the fingertips and the interdigital clefts, the silhouette of each finger can be traced along the hand contour in the 2D image plane. We only take a certain length of segment used for feature matching. Exceedingly short segments are discarded since they do not have enough position information to help locate the fingers and may misguide the matching to an unlike direction. Too long segments are cropped to an appropriate length for computation efficiency and matching accuracy.

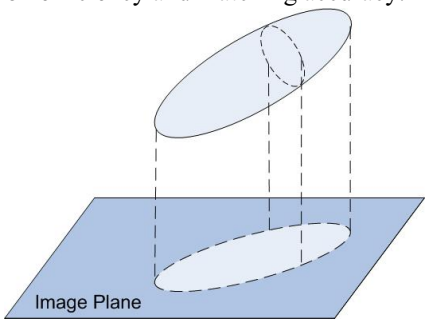


Fig. 6. A 3D ellipsoid projection into the image plane

The silhouette of the ellipsoidal-primitive model can be computed directly using the 2D projective geometry. Since our measurements are a 3D depth map, for simplicity we assume a pseudo-orthographic projection from 3D space to 2D image plane. In this way each ellipsoid is projected into an ellipse in the image plane. From (3) we have

$$\begin{bmatrix}\mathbf{X_I}^{\mathbf{T}} & z\end{bmatrix}\begin{bmatrix}\mathbf{A} & \mathbf{b}\\ \mathbf{b}^{\mathbf{T}} & c\end{bmatrix}\begin{bmatrix}\mathbf{X_I}^{\mathbf{T}}\\ z\end{bmatrix}=0 \tag{7}$$

where $\mathbf{X_I}^{\mathbf{T}} \in \mathbf{R^2}$ is the point coordinate in 2D image plane. As shown in Fig. 6, for the silhouette of the projected primitive, it has unique solution of (7) for z, which can be expressed as

$$\begin{cases}\mathbf{X_I}^{\mathbf{T}}(\mathbf{b}\mathbf{b}^{\mathbf{T}}-c\mathbf{A})\mathbf{X_I}=0\\ z=-\dfrac{\mathbf{b}^{\mathbf{T}}}{c}\mathbf{X_I}\end{cases} \tag{8}$$

By solving Equation (8), we can extract the silhouette of the primitive model and matched against the segments detected in depth map projection. The nearest neighbor strategy is used to setup the corresponding relationship between the points on model silhouette and on depth map segment.

C. 3D Surface

Since our 3D depth map measurements are dense and noisy, the extraction of common features, such as the normal and tangent vectors of small patches, is computationally expensive and imprecise. Our tactics is to use only a limited number of points randomly sampled from the 3D measurements as the feature points for the matching. However, the discrete nature of the sample set introduces local minima into the optimization function of the matching step, and may trap the optimization process far from the actual posture. A stochastic sampling strategy is used to circumvent local minima and to lower the chance of getting stuck in them. As a result, random changes in the sample set are introduced at each iteration step.

The polygonal-surface model updates its vertices using the linear blend skinning technique, as shown in (6). The correspondences to the measurement samples are chosen as the nearest neighbors among the vertices on the polygonal surface.

IV. Feature Matching

We consider the matching problem as estimating an appropriate configuration of hand position and joint angles. A state vector $\boldsymbol{\theta}$ is parameterized to model the hand posture with 30 DoFs, as shown in Fig. 3. Similar to the feature extraction step, the matching is implemented in a hierarchical order, from low to high level of hand model, to refine the precise position of the hand. For all levels of matching, it is formulated as a nonlinear optimization problem, which is defined by a cost function and certain state constraints. We use the 3D Euclidean distance between the corresponding feature point set of the measurements and the hand model as the cost function,

$$f_{cost}=\sum D(X_{measurement}, X_{model}) \tag{9}$$

and the optimization process aims at minimizing these distances with respect to the state vector $\boldsymbol{\theta}$. The minima are searched with a Scaled Conjugate Gradient (SCG) [21] estimator. The details of the cost function and the Jacobian matrix computation for the optimization are introduced below.

A. Skeletal Level

The feature points extracted for this level of matching are the fingertips. The corresponding fingertips of the 3D measurements and the skeletal model are paired according to their relative order in the 2D projection. The overall 3D Euclidean distance between the pairs is computed as the cost function for SCG optimization.

For each iteration of SCG optimization, the fingertip positions of the skeletal model is updated using (2)

$$\mathbf{X}_i=\mathbf{M}_i\cdot\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T \tag{10}$$

And the Jacobian matrix for SCG optimization of this matching level can be formulated as

$$\mathbf{J}_i=\frac{\partial\mathbf{X}_i}{\partial\boldsymbol{\theta}}=\frac{\partial\mathbf{M}_i}{\partial\boldsymbol{\theta}}\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T \tag{11}$$

where $\partial\mathbf{M}_i/\partial\boldsymbol{\theta}$ can be expressed as

$$\frac{\partial \mathbf{M}i}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{M}_0^1}{\partial \boldsymbol{\theta}}\mathbf{M}_1^2 \ldots \mathbf{M}_{i-1}^i + \mathbf{M}_0^1 \frac{\partial \mathbf{M}_1^2}{\partial \boldsymbol{\theta}} \ldots \mathbf{M}_{i-1}^i + \ldots + \mathbf{M}_0^1 \mathbf{M}_1^2 \ldots \frac{\partial \mathbf{M}_{i-1}^i}{\partial \boldsymbol{\theta}} \quad (12)$$

B. *Ellipsoidal-Primitive Level*

For the ellipsoidal-primitive model, the feature extracted is the silhouette of the fingers. The extracted silhouette is then down-sampled to lower the computational complexity. The corresponding point pairs are setup by finding the closest point in 3D space of the silhouette from the primitive model. This model's silhouette is updated by solving (8), and if we assume the local coordinate of the point remains the same during the transformation, it can be computed by mapping the silhouette points from the global coordinate system back to the local joint coordinate system, as in (13)

$$\mathbf{X}_i' = \mathbf{M}_i^{-1} \cdot \mathbf{X}_i \quad (13)$$

where $\mathbf{X}_i'$ is the local coordinate of the points on the ellipsoid attached to the *i-th* joint. The computation of the Jacobian matrix is simplified to a similar form of (11)

$$\mathbf{J}_i = \frac{\partial \mathbf{X}_i}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{M}_i}{\partial \boldsymbol{\theta}} \mathbf{X}_i' = \frac{\partial \mathbf{M}_i}{\partial \boldsymbol{\theta}} \mathbf{M}_i^{-1} \mathbf{X}_i \quad (14)$$

and $\partial \mathbf{M}_i / \partial \boldsymbol{\theta}$ is of the same form as in (12).

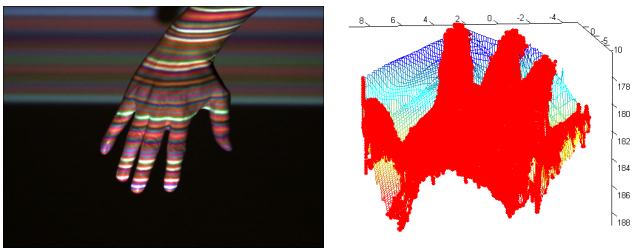
C. *Polygonal-Surface Level*

In this level, a limited number of samples are taken from the measurements and the corresponding point pairs are matched in the similar way presented above. Since the vertices of the polygonal surface are updated using (6), its Jacobian matrix is of the form

$$\mathbf{J}_i = \frac{\partial \mathbf{v}_c}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{n} \gamma_i \frac{\partial \mathbf{M}_i}{\partial \boldsymbol{\theta}} M_{i0}^{-1} \mathbf{v}_0 \quad (15)$$

V. Experiment Results

In this paper we focus on tracking the human typing motion with 3D depth maps as the input. The 3D depth maps are generated using the structured light (SL) system with pre-calibrated projector-camera pair. The depth precision for this ranging system is at the mean error of 0.12mm and the standard deviation of 1.13mm (using a flat calibration plane as the reference target). The captured image resolution before 3D reconstruction is 640 × 480 and the frame rate for online image capturing is about 30 frames per second.

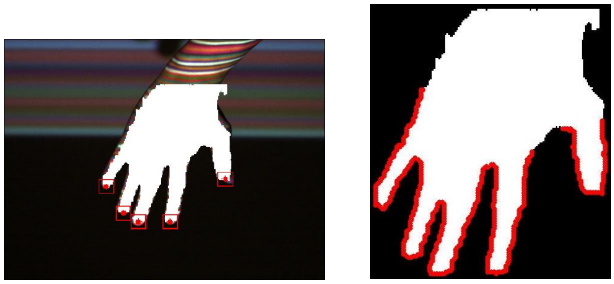


(a) (b)

Fig. 7. (a) Original image from the SL system. (b) 3D depth map generated by the SL system.

A. *Feature Extraction*

Once the 3D depth map is captured, as shown in Fig. 7 (b), we can use it as the input for background subtraction and hand contour segmentation. Next, the fingertip detection and finger silhouette extraction steps are employed to extract different features for multi-level feature matching, as explained in Section III. Fig. 8 shows the results of (a) the detected fingertips, which is marked as the red-lined squares and centered on the detected local maxima, and (b) the labeled and cropped finger silhouette, which is marked with red dots. The feature extraction step takes less than 0.1 second on an Intel Core™2 1.83GHz CPU, using a Matlab implementation. This step can be implemented with a paralleled framework for faster performance. Further speedup for tracking scheme can be easily achieved by using an optimized implementation.



(a) (b)

Fig. 8. (a) Fingertip detection result (b) Finger silhouette extraction result

B. *Hand Matching Results*

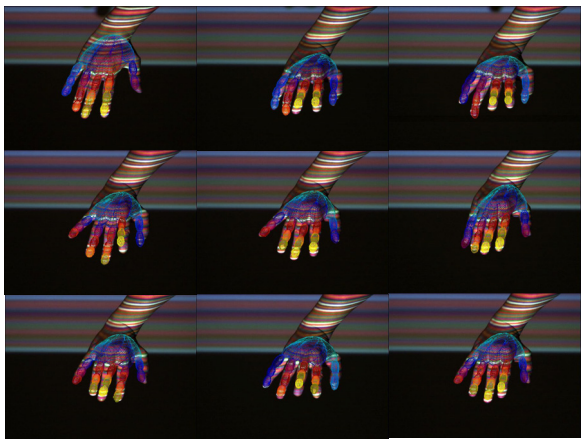


Fig. 9. Tracking 150 frames of human typing motion, with noisy 3D depth map. The reconstructed polygonal skin surface is projected onto the image

Fig. 9 shows the hand matching results using the MLFM framework proposed in this paper for the same subject. The reconstructed hand is represented using the ellipsoidal-primitive model wrapped around the underlying skeleton, and is projected back into 2D image plane for comparing with the original hand postures. The mean matching error is less than 0.1cm, using the definition in [22]. The total processing time is 0.4 sec/frame by a C implementation with no major optimization. Fig. 10 shows the synthesized 3D human hand typing on a virtual keyboard.

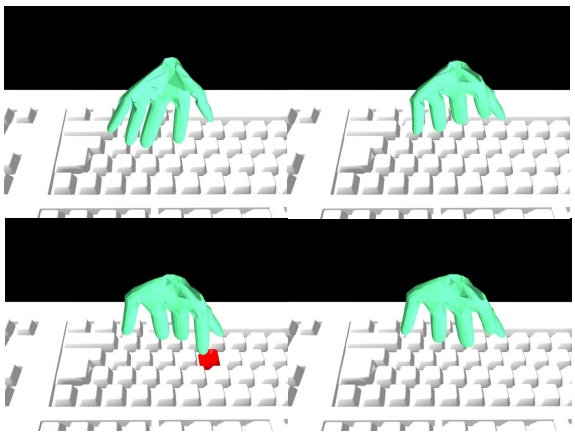


Fig. 10. Synthesized 3D human hand typing on a virtual keyboard, with key-pressing event detected

We also compared our matching scheme with another 3D hand matching approach designed for virtual keyboard system [22]. This reference framework applied a physical based model fitting (PMF) method with detailed 3D hand skin model. Fig. 11 shows that our approach converges faster and has less matching error, thus it is better suitable for a real time virtual keyboard system, whereby event detection accuracy and speed are essential.

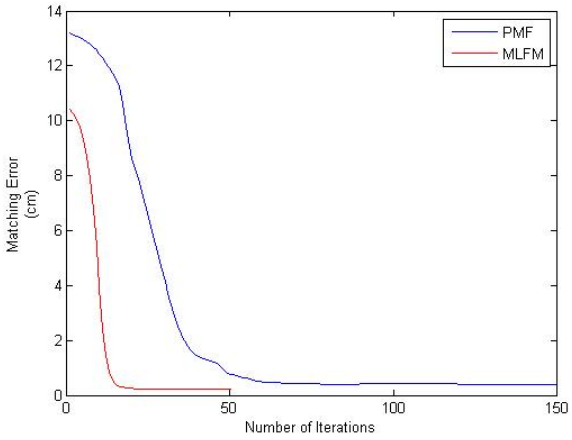


Fig. 11. Comparison of MLFM and PMF by matching error and number of iterations

VI. Conclusion

In this paper, we propose a virtual keyboard system based on the Multi-Level Feature Matching method. We show that this framework can generate high precision matching results for hand typing postures in near real-time. Moreover, the experimental results show that the matching scheme is effective for reconstructing the hand typing postures with roughly detected fingertips and hand contour from the noisy 3D depth maps, and the mean matching error is small enough for virtual keyboard applications.

Further research will be carried out in stylistic motion learning direction. With more captured hand postures, the tracking process can be trained to build a prior model for the normal human typing motions. Another research direction is to automatically calibrate the 3D hand model according to 3D measurements of user's hands during the warm-up period of the virtual keyboard system. Since the main size information of our hand model is controlled by the underlying skeleton system, the only problem is how to adapt the bone lengths and angles to some specific hand measurements.

References

[1] Kölsch, M. and Turk, M. "Keyboards without Keyboards: A Survey of Virtual Keyboards," *Workshop on Sensing and Input for Media-centric Systems*, Santa Barbara, CA, June 20-21, 2002.

[2] D. H. Won, H. G. Lee, J. Y. Kim and J. Park, "Development of A Wearable Input Device Recognizing Human Hand and Finger Motions as A New Mobile Input Device," *International Conference on Control, Automation and System*, pp. 1088-1091, Oct. 2001

[3] Senseboard Tech. AB, http://www.senseboard.com

[4] A. Agarwal and B. Triggs, "3D human pose from silhouettes by relevance vector regression," *Proc. of the Computer Vision and Pattern Recognition (CVPR)*, pp. 882-888, July 2004.

[5] T. Darrell, I. Essa and A. Pentland, "Task-Specific Gesture Analysis in Real-Time Using Interpolated Views," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 18, no. 12, pp. 1,236-1,242, Dec. 1996.

[6] D. M. Gavrila and L. S. Davis, "Towards 3D Model-Based Tracking and Recognition of Human Movement: A Multi-View Approach," *Proc. Int. Workshop on Automatic Face and Gesture Recognition (AFGR)*, pp. 272-277, June 1995.

[7] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff, "3D hand pose reconstruction using specialized mappings," *IEEE Int. Conf. on Computer Vision (ICCV)*, Vol. 1, pp. 378-385, July 2001.

[8] B. Stenger, A. Thayananthan, P. H. S. Torr and R. Cipolla,"Filtering using a tree-based estimator," *IEEE Int. Conf. on Computer Vision (ICCV)*, pp.1102-1109, Oct. 2003.

[9] J. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: An application to human hand tracking," *Proc. of European Conference on Computer Vision (ECCV),* Vol. 2, pp. 35-46, May 1994.

[10] J. Lee and T.L. Kunii, "Constraint-based hand animation," *Models and Techniques in Computer Animation,* pp. 110-127, June 1993.

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

[12] M. de La Gorce and N. Paragios, "Monocular Hand Pose Estimation Using Variable Metric Gradient-Descent," *Proc. British Machine Vision Conference (BMVC)*, pp. 1269-1278, Sept. 2006.

[13] A. Meyering and H. Ritter, "Learning to Recognize 3D-Hand Postures From Perspective Pixel Images," *Artificial Neural Networks 2*, I. Elsevier Science Publishers B.V., 1992.

[14] B. Stenger, P. Mendonca and R. Cipolla, "Model-based hand tracking using an unscented kalman filter," *Proc. British Machine Vision Conference (BMVC)*, pp. 63-72, Sept. 2001.

[15] M. Bray, E. Koller-Meier and L. Van Gool, "Smart particle filtering for 3D hand tracking," *Proc. Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, pp. 675-680, May 2004.

[16] A. Mohr and M. Gleicher, "Building efficient, accurate character skins from examples," *ACM Trans. on Graphics (SIGGRAPH)*, pp. 562-568, July 2003.

[17] R. Lange, "3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology," Ph.D. Thesis, ETH-Zürich, 2000.

[18] B. Porr, B. Nuerenberg and F. Woergoetter, "A VLSI-Compatible Computer Vision Algorithm for Stereoscopic Depth Analysis in Real-Time," *Int. Journal on Computer Vision (IJCV)*, vol. 49, issue 1, pp. 39-55. Aug. 2002

[19] F. Forster, "Real-time range imaging for human-machine interface," PhD thesis, Technische Universität München, Germany, 2005.

[20] H. Du, T. Oggier, F. Lustenburger and E. Charbon, "A virtual keyboard based on true-3D optical ranging," *Proc. British Machine Vision Conference (BMVC)*, Oxford, pp. 220-229, Sept. 2005.

[21] M. F. Moller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Networks*, pp. 6:525-533, 1993.

[22] H. Du and E. Charbon, "3D hand model fitting for virtual keyboard system," *IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 31-36, Feb. 2007.