2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Fast 3D Hand Estimation for Mobile Interactions

Yuru Pei
Machine Intelligence Department
Peking University
Beijing 100871, China
Email: peiyuru@cis.pku.edu.cn

Gengyu Ma
uSens Inc
San Jose, CA 95110, USA
Email: magengyu@gmail.com

***Abstract*—The ubiquitous hand gesture plays an important role in the natural human machine interaction (HMI). Recently, the consumer color and depth cameras have been used to estimate hand shapes and postures for the mid-air HMI. Under the observation that 3D hand contours possess much information of hand postures, we estimate 3D hand contours from infrared images with a limited computation complexity for the HMI on mobile devices. A variant of the dynamic programming (vDP) algorithm is proposed to handle complex self-occlusions in 3D hand estimations, where a set of heuristic rules are introduced to avoid finger missing. Furthermore, the constraints are used to reduce the searching space in contour alignments. Given 3D hand contours, a set of hand gestures, including touching, swiping, and pinching, can be applied to mid-air interactions. The proposed method is much faster than the traditional depth estimation of the whole hand, and can achieve up to 500 Hz on PC, and 100 Hz on mobile devices.**

I. Introduction

The recent developments in image capturing devices and software for the natural human machine interaction (HMI) have facilitated more and more intuitive human-machine interfaces. The 3D hand plays an important role in the HMI. Consumer color and depth cameras, including Kinect [3] and the Leap Motion controller [4], have been used to acquire real-time 3D hand positions and actions. A variety of techniques have been proposed for 3D articulated hand estimation from depth images [19], [20], [21]. The active illumination of the structured light or the time-of-flight often requires a suitable operation space for reliable depth estimations. The binocular color and infrared cameras impose relatively few restrictions on hand image capturing. The consumer binocular infrared camera, e.g. the leap motion controller, can capture hand gestures in a relatively small space close to the camera and process up to 120 frames per second. Furthermore, the binocular device has simpler hardware and lower price than the depth camera. The stereo setup [15] and the data-driven methods [17], [18], [2] have been used to get 3D hands. However, computation complexities of the above techniques are relatively high and sometimes with parallel processing of GPUs, which are not suitable for the HMI on mobile devices.

Under the observation that hand contours possess much posture information, 3D contours can be used for the gesture recognition in a large variety of mid-air mobile interactions. Since the time complexity of depth estimation from 2D images depends on the pixel number to be reconstructed, the 3D hand contour estimation is much efficient than that for the whole hand. The 3D contour reconstruction has been addressed in hand tracking tasks, where the iterative closest point (ICP) [1] and the dynamic programming (DP) based methods [15] were used to align the 2D hand contours. Since the holistic ICP and the dynamic time warping (DTW) methods only have limited capacities to deal with partial correspondences between contours, only relatively simple postures with little self-occlusions can be handled. The misalignment will occur when part of finger contours do not have counterparts in complex occlusion cases.

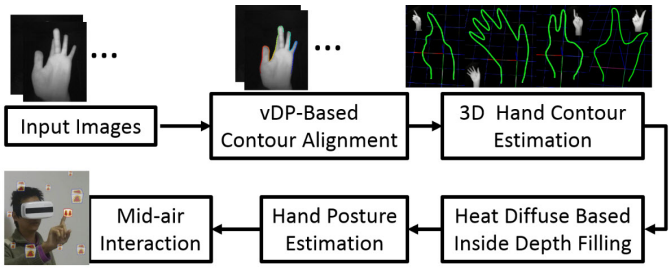

Fig. 1. System flowchart.

In response to this problem, we propose a variant of the dynamic programming (vDP) algorithm to establish hand contour correspondences for fast 3D hand reconstruction from infrared images (Fig. 1). In order to avoid matching ambiguities, the side-labels of contours, together with the heuristic matching rules, are introduced for reliable contour alignment. 3D finger missing due to self-occlusions can be avoided. Moreover, the heuristic rules can greatly reduce the searching space in the cost matrix for an efficient contour alignment. Once given 3D hand contours, a set of hand gestures, including touching, swiping, and pinching can be applied to mid-air interactions. For instance, the fingertips extracted from 3D hand contours can be used in virtual touching for object selection. However, in the swiping-like task, the bounding boxes of 3D hand contours could not provide accurate intersection detections between objects and palms. A two-stage filling scheme is employed to acquire the depth inside hand contours. Firstly, a region-based linear interpolation along the scan line is used to initialize the depth inside hand contours. Secondly, the depth is refined based on the heat diffusion with contour constraints, which is modeled by an Euler-Lagrange equation

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

with Dirichlet boundary conditions. The diffusion problem is relaxed by solving a large linear system. The proposed method can process up to 500 infrared images per second on PC and 100 on the Samsung Galaxy S6 mobile platform [5], which is efficient enough for mobile devices.

The main point of this work is to propose a vDP-based method for the reliable contour alignment in 3D hand estimation, as well as hand postures for the mid-air interactions. The proposed method can ease the computation burden and suitable for mobile devices.

II. Related Work

The hand gesture plays an important role in the HMI. A bundle of works addressed the automatic hand detection, tracking, modeling, and gesture recognitions in vision-based systems [9], [11]. The hand contour can provide much information on hand postures, which is independent of skin colors and illuminations [13]. The ICP algorithm and an assumption of the affine motion model were used for 3D hand contour reconstruction [1]. The DTW was used for matching and reconstructions of contour points for the purpose of 3D hand contour tracking [15]. The holistic alignment of the ICP method, or the DTW method with limited searching capacities often failed to find correct contour alignments in the case of self-occlusions. In this work, a vDP-based method with augmented searching power is employed for contour alignments and 3D hand estimations.

A number of mid-air interaction systems have been proposed to augment the touchscreen or the acceleration sensors based mobile interactions. LucidTouch combined a multi-touch input surface with a pseudo transparent display [22]. A compact high-frame-rate camera and white LEDs were used for in-air typing [10]. The SixthSense as a wearable gestural interface used a tiny projector and a camera coupled to a pendant-like mobile wearable device [8]. The hybrid classification-regression forests were used to estimate dense 3D hands with a modified 2D camera and the simple LED-based illumination [2]. The similar cascaded random forests were used to infer hand shapes and positions from a monocular RGB imagery for mid-air interactions in unmodified portable devices [17], [16], and the hand pose estimation from consumer depth cameras [19], [20]. Jang *et al.* [6] handled self-occlusions of fingers in selecting objects in an AR/VR space with an egocentric viewpoint of the camera-attached HMD. Convolutional networks were employed for the real-time hand pose recovery [21]. However, the computation complexities of most above techniques are relatively high [19], [20], [6] and not suitable for mobile devices. Some color-camera-based systems had limited capacity to provide 3D gestures for the HMI [17], [16]. Considering the limited computation power of the mobile devices, we propose a vDP-based method for a fast 3D hand posture estimation.

III. vDP-Based Contour Alignment

In this work, we cope with the 3D reconstruction of hand contours from stereo infrared images. In the stereo setting, the reliable contour correspondence is essential to 3D contour estimation. Given the contour matching, the contour depth can be estimated based on the stereo geometry. The contour correspondence falls into a sequence matching problem, and the dynamic programming based methods, e.g. the DTW, is feasible to solve correspondences [15]. However, due to repetitive structures and self-occlusions, it's a nontrivial task to acquire reliable contour correspondences. Confronted with this problem, we propose a vDP-based method for the contour alignment.

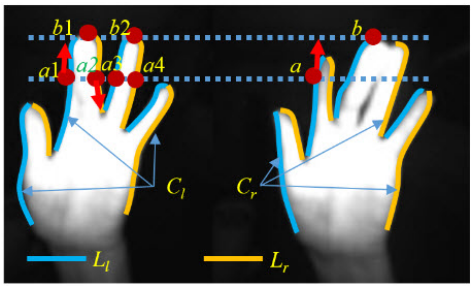

Fig. 2. Illustration of contour alignments.

A. Pairwise Contour Point Similarity

Let C_l and C_r denote the hand contours on the two infrared images respectively. From the stereo matching view, the matching points should be on the epipolar line [15]. It's intuitive the point should be close to the epipolar line of its counterpart considering device noises. In the rectified stereo image pairs, the distance between contour points can be measured simply by the difference in the y-direction. The epipolar-based local feature is defined as: $f_e(c_i) = c_{i,y}$, where $c_{i,y}$ denotes the y-direction coordinate of contour point c_i. However, when the epipolar line intersects the contour with more than one point, the mapping confusion will occur. For instance, point $a1$-4 are all matching candidates of point a, and points $b1$ and $b2$ are both matching candidates of b as shown in Fig. 2. The spatial consistency can be used to discriminate the wrongly matching. We introduce the shape descriptor defined as the first order difference of neighboring contour points, and $f_r(c_i) = \frac{c_i - c_{i-1}}{\max\limits_{c_j, c_{j-1} \in C} \|c_j - c_{j-1}\|}$. f_r can be seen as the contour direction. Although point $a2$ and $a4$ are on the epipolar line and viewed as the candidate counterparts of a as shown in Fig. 2, the directions of $a2$ and $a4$ are almost reverse to that of a. The local shape descriptor can remove most such wrong epipolar-based matchings. Considering the repetitive shape pattern of fingers, the displacement from the palm center is used to describe the position of the contour point, and $f_c(c_i) = \frac{c_i - \bar{c}}{\max\limits_{c_j \in C} \|c_j - \bar{c}\|}$, where palm center $\bar{c}$ is defined as the contour centroid.

The pairwise distance d between the contour points C_l and C_r is computed as a combinatorial difference of epipolar-based features, together with the shape descriptor including

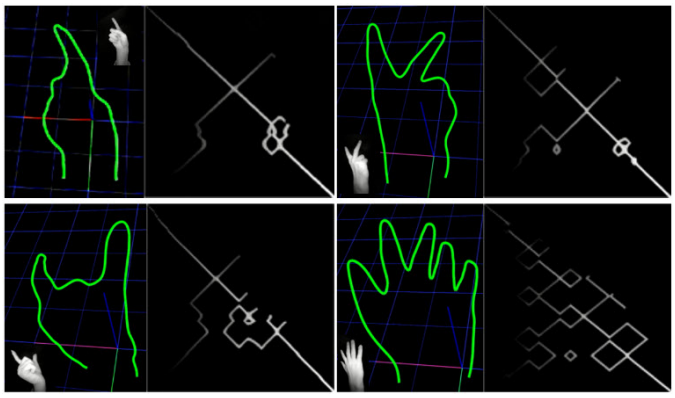

Fig. 3. Cost matrices of four sampled hand contours with input infrared images corner-plotted.

the contour direction vector and the displacement from the palm center.

$$d(i,j) = \alpha_e \left\| f_e(c_i) - f_e(c_j) \right\| + \alpha_r \gamma \left\langle f_r(c_i), f_r(c_j) \right\rangle + \alpha_c \left\langle f_c(c_i), f_c(c_j) \right\rangle, \text{ and} \quad (1)$$

$$\gamma = \begin{cases} \infty, \text{if } \left\langle f_r(c_i), f_r(c_j) \right\rangle < 0, \\ 1, \text{ otherwise,} \end{cases}$$

where γ is the penalty coefficient used to avoid the wrong matching pairs with reverse directions. α_e, α_r, and α_c are constant coefficients, and set at 0.5, 1, and 1 respectively in our experiments.

B. Cost Function

Given the pairwise distance matrix, the goal is to find the hand contour alignment with the smallest cost. The dynamic programming technique is efficient to solve such problem with optimal substructures. The optimal matching between C_l and C_r relies on the solution of sub problems. Let function $p(i,j)$ denote the minimum cost from the starting point to point pair (c_i^l, c_j^r). $p(i,j)$ can be estimated based on the optimal cost in the previous subsequences. It's deserved to note that there is no guarantee that all points in one contour have counterparts in the other due to occlusion. A comparatively large searching region is needed to handle the self-occluded fingers. The cost function of contour alignment is defined as follows.

$$p(i,j) = \begin{cases} d(i,j), \text{ if } i = 1 \text{ or } j = 1, \\ \Phi(i,j) + d(i,j), \text{ otherwise,} \end{cases} \quad (2)$$

where $\Phi(i,j) = \min\{p(i-k_i, j-1), p(i-1, j-k_j)\}$. The first row is related to the starting point in the contour. The gaps k_i and k_j in the x- and y-directions can vary from 1 to $i-1$ and $j-1$ respectively.

C. Constraints

In order to avoid the mapping ambiguity, the side-label is assigned to the contour point. Here, the odd intersected points with the scan line are denoted as the left-sided point L_l, while the even intersected points are denoted as the right-sided points L_r as showed in Fig. 2. The foreground finger is on the right side of L_l, and on the left side of L_r. Traditional DP-based method for contour matching only considered the contour as a sequence of points, but ignored the relation between contours and foreground hand regions. By virtue of the side label, we can use a variety of rules to improve the contour matching. (i) It's intuitive that the matching points should have the same side label based on the spatial consistency, and the points in the current matching pair $c_i^l, c_j^r \in L_l$ or $c_i^l, c_j^r \in L_r$. (ii) Since the foreground finger is on the right side of L_l, the gap from the preceding left-sided point to the current left-sided one should not be large to avoid finger missing. (iii) Similarity, the gap from the preceding left-sided point to current right-sided one should be small enough to avoid finger missing. (iv) The disparity between neighboring matching pairs, e.g. $(c_{i,x}^l, c_{j,x}^r)$ and $(c_{i',x}^l, c_{j',x}^r)$, should be small, and $\|c_{i,x}^l - c_{j,x}^r\| \simeq \|c_{i',x}^l - c_{j',x}^r\|$.

In the case of complex self-occlusions of fingers, relatively large transfers are needed in both contours. For instance, the middle finger is occluded by the index finger in one image, while occluded by the ring finger in the other image as shown in Fig. 2. It means that the searching for the optimal path needs to be performed in the sub-block of the processed contour points. Fortunately, since the epipolar-based local feature should be similar and the above side-label-based rules should be satisfied, the searching space is limited. In our system, the number κ of possible matching candidates is approx. 5. The time complexity is $O(\kappa n_c)$ for contour alignment with n_c points. The cost matrices of four sampled hand contours are illustrated in Fig. 3. As we can see, the cost matrix is sparse, and the candidate space is limited by virtue of the proposed constraints.

IV. Depth Interpolation

Given the contour alignment, it's straightforward to compute the 3D hand contour by the stereo geometry. The 3D hand contour is enough to detect and locate the fingertips used in the touching-like interactions. However, when the virtual objects interact with the palm in the swiping-like interactions, 3D depth missing inside hand contours can cause confusions. The bounding box determined by the 3D hand contour is not accurate enough for the intersection detection.

Depth filling with boundary conditions can be mathematically modeled by the Euler-Lagrange equation: $\Delta g = 0$ over Ω with $g|\partial\Omega = V_C$, where Δ is the Laplace operator. Ω denotes the hand region with contour $\partial\Omega$. The depth function $g(v_i)$ with contour V_C can be seen as the solution of the following minimization problem as in [12]:

$$E = \int\int_{\Omega} |\nabla g|^2 \text{with } \ g|\partial\Omega = V_C, \quad (3)$$

where ∇ is the gradient operation. We employ the discrete image grid of the infrared image, where each pixel has

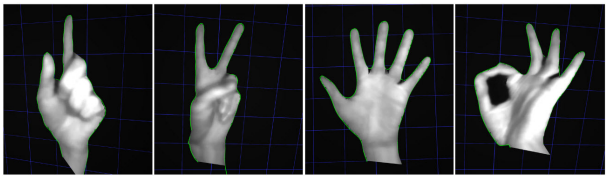

Fig. 4. The depth interpolation inside 3D hand contours.

regular four-connected neighbors. By combining the boundary condition, the energy function is relaxed as follows.

$$E = \sum_{i=1}^{n_c} \|g(v_i) - V_c(v_i)\|^2 + \beta \sum_{i=1}^{n_v} \sum_{v_j \in Neib(v_i)} \|g(v_i) - g(v_j)\|^2. \quad (4)$$

The first term is the boundary conditions to make the depth $g(v_i)$ of contour point v_i equal to the 3D estimated contour $V_C(v_i)$ based on the stereo geometry as described in Section III. The second is the smoothness term to make the depth difference of neighboring points small enough. β is a constant coefficient, and set at 0.01 in all experiments. n_v denotes the number of points in the hand region. By setting the first derivative of Eq. 4 to zero, the energy function is converted to a large sparse linear system.

$$g(v_i) + \beta \left(|Neib(v_i)| g(v_i) - \sum_{v_j \in Neib(v_i)} g(v_j) \right) = V_c(v_i). \quad (5)$$

Here, we employ the straightforward linear interpolation to provide the initial value of $g(v_i)$. Since the depth inside one finger is relatively independent of others, the hand region is coarsely divided according to finger valleys. In our system, since the contour points are side labeled, the separation points in the finger valleys are easily located by the local minimum of finger contour points with different side-labeled neighbors. The linear interpolation is performed inside each region. Given the initial depth estimation by the linear interpolation, the conjugated gradient method is used to solve depth $g(v_i)$ inside the contours. The 3D hands after depth interpolation are shown in Fig. 4 with textures from infrared images. The time complexity is $O(n_v \rho^{0.5})$, where ρ is the spectral condition number of the coefficient matrix in the linear system.

V. Experiments

In order to validate the proposed method, we perform experiments in the toy data set with the hand image pairs rendered from a virtual 3D hand. We also perform the 3D contour estimation from the real images captured by the infrared camera.

In our system, the number of input contour points varies from 200 (hand is about 20 cm away from camera) to 1000 (5 cm away). In order to get a constant processing time, we down sample the contour to approx. 400 points. A grid of 20×20 is used for depth interpolation. One of the time-consuming procedure is the silhouette extraction. Here we directly call the OpenCV function $cv :: findcontour()$. We test the proposed method on a Samsung Galaxy S6 mobile platform. The average processing time is 10 ms, which almost catches up with the frame rate of the infrared camera of 120Hz. Moreover, for the touch-like interactions, there is no need to estimate the inside depth, and the time cost is even lower. The average time cost of main procedures, including the contour extraction (CE), the feature definition (FD), the vDP-based contour alignment and 3D reconstruction, and the depth interpolation (DI), on the PC and the mobile device (MD) are shown in Table I. This system is the fastest mid-air hand interaction system on the mobile platform as far as we know.

TABLE I
AVERAGE TIME COST OF MAIN PROCEDURES ON PC AND MOBILE DEVICES (MD). CE-CONTOUR EXTRACTION; FD-FEATURE DEFINITION; DI-DEPTH INTERPOLATION

Time (ms)	CE	FD	vDP	DI	Total
PC	0.93	0.21	0.69	0.20	2.03
MD	4.8	1.0	3.4	0.81	10.0

A. Evaluations

We compare the proposed method with the matching based on the commonly-used DTW [15], and the ICP [1] methods as shown in Fig. 5 and Table II. The contour alignment can be solved by finding the largest common sub-contours, which is the same as the longest common substring (LCS) problem [7]. Here we also compare with the LCS-based method. We use the y-coordinate (epipolar in rectified images) as the character of the string. The proposed method outperforms other dynamic programming and ICP-based methods, especially in the regions without counterparts due to self-occlusions. For instance, when the middle finger is partially occluded by the index and the ring finger respectively, parts of the index finger contour in C_l and the ring finger contour in C_r have no counterparts as illustrated in Fig. 5, 3rd column. The relatively large transfer gaps in both hand contours are needed for reliable matchings as described in section III. The proposed method can handle self-occlusion cases and acquire correct matchings. Whereas, the other methods all failed as shown in Fig. 5.

The depth of 3D reconstructed hand contours by the proposed method, the LCS [7], the DTW [15], and the ICP [1] methods, along with the ground truth are illustrated in Fig. 6. The 3D contours estimated by the proposed methods are more consistent with the ground truth, especially in the self-occluded regions.

As described in Section III-A, the epipolar-based and the shape-based features are used as the contour descriptors. We compare the reconstruction errors based on different feature channels as shown in Table II. In our experiments, the feature fusion (vDP$_{fusion}$) outperforms those using the epipolar-based (vDP$_{epi}$) or shape-based feature (vDP$_{shape}$) alone. The same occurs to the DTW and ICP-based contour alignment, where

Fig. 5. The contour alignment and 3D hand contour estimation by the proposed methods with constraints (vDP) and without constraints (vDP-nc), the ICP, the LCS, and the DTW methods (from top to bottom) with error matchings yellow-blocked. 3D estimated hand contours are green colored with fingertips red colored.

TABLE II
3D RECONSTRUCTION ERRORS (MM) OF THE PROPOSED METHOD, THE LCS, THE DTW, AND THE ICP METHODS. *nc*-WITHOUT CONSTRAINTS; *co*-WITH CONSTRAINTS.

Methods		Errors (mm)	
		nc	*co*
Ours	vDP$_{shape}$	27.3	18.8
	vDP$_{epi}$	14.5	9.72
	vDP$_{fusion}$	12.8	**8.55**
DTW[15]	DTW$_{shape}$	80.1	
	DTW$_{epi}$	73.4	
	DTW$_{fusion}$	68.8	
ICP[1]	ICP$_{shape}$	50.2	
	ICP$_{epi}$	43.4	
	ICP$_{fusion}$	37.8	
LCS[7]		84.2	

the feature fusions (DTW$_{fusion}$ and ICP$_{fusion}$) outperform those using epipolar-based feature alone (DTW$_{epi}$ and ICP$_{epi}$) or shape-based feature alone (DTW$_{shape}$ and ICP$_{shape}$).

In our system, the heuristic constraints are used to avoid matching ambiguities. The searching space is largely reduced based on the side-labels and transfer constraints as described in Section III-C. We compare the contour alignment with and without constraints as shown in Table II. The constrained version can improve the reconstruction accuracies compared with unconstrained ones.

B. Interactions

Given the 3D hand, it's straightforward to interact with virtual 3D objects. We only need to detect the collision of the hand and objects. A loose axis-aligned bounding box

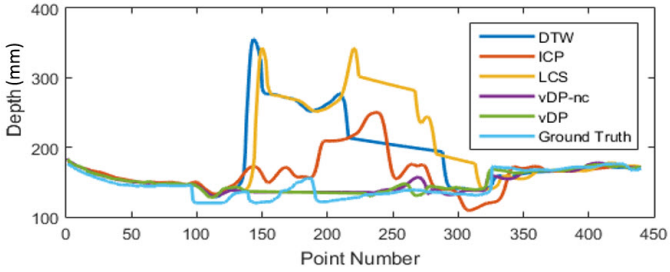

Fig. 6. Depth comparison of 3D reconstructed hand contours by the proposed methods with constraints (vDP) and without constraints (vDP-nc), the LCS [7], the DTW [15], the ICP [1] methods, along with the ground truth.

(AABB) is used to represent the 3D object. In the fingertip touching task, the system needs to detect the intersection of the fingertip with the object. When the collision occurs, the object is selected. In order to reduce the high-frequency noise and avoid the temporal lags, a temporal filter is imposed on the hand motion vectors q.

$$q_i = \frac{1}{A}\sum_{j=0}^{k-1} w_j q_{i-j}, \tag{6}$$

where the normalization factor $A = \sum_{j=0}^{k-1} w_j$. The coefficient w is dynamically adapted based on the exponential moving average technique [14], and $w_j = \exp(-j\,\|q_i - q_{i-j}\|)$. As shown in Fig. 7 and the video, the 3D fingertips estimated from the hand contours can be used to select virtual objects in the space. The cubes touched by the fingertips are red-colored, and those touched by other parts of the hand are gray-colored. The cubes untouched are blue-colored.

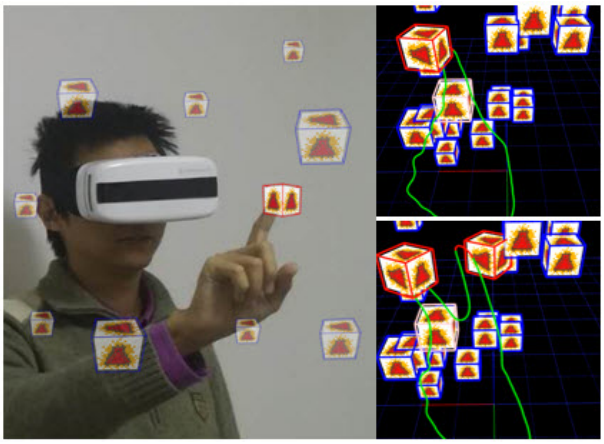

Fig. 7. The virtual touching with one and two fingers. The touched cubes by fingertips are red outlined. The touched by other parts of the hand is gray outlined, and the untouched blue outlined.

The commonly-used SVM classifier is used for the gesture recognition. In the system, we define a set of 11 hand gestures, including *swipe up*, *swipe down, swipe left, swipe right, scroll clockwise, scroll counterclockwise, push, pull, one fingertip touch, two fingertips touch,* and *pinch*. The confusion matrix of the gesture recognition is shown in Fig. 8. The recognition result is nearly perfect. The only confusions occur between the *swipe left* and *scroll counterclockwise*, along with *swipe right* and *scroll clockwise*, due to the nearly identical starting phase of the gestures.

VI. Conclusion

In this paper, we proposed a vDP-based method for 3D hand contour estimation from stereo infrared images. The side-label based heuristic rules are introduced to avoid the ambiguity in the contour alignment. By virtue of the heuristic rules, the searching space in the cost matrix can be greatly reduced. The efficient iterative method is employed for depth filling inside hand contours for the palm-related interactions. We compared our method with commonly-used dynamic programming methods, including DTW and LCS, along with the ICP-based method in contour alignment. The proposed method can greatly improve the reconstruction accuracies and acquire reliable hand contours even in the case of complex finger occlusions. Since only the contour points are processed, the 3D reconstruction is apparently more efficient than the dense 3D hand estimation. The proposed system can achieve up to 100 Hz for the 3D hand posture estimation on mobile devices, which almost catches up with the image capturing rate of the consumer infrared camera.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China under Grant 61272342, and the Seeding Grant for Medicine and Information Sciences of Peking University under Grant 2014MI24.

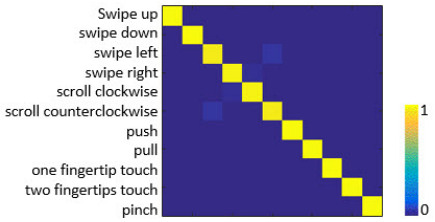

Fig. 8. The confusion matrix of contour based gesture recognitions.

References

[1] A. A. Argyros and M. I. Lourakis, "Binocular hand tracking and reconstruction based on 2d shape matching," in *Proc. ICPR*, vol. 1, 2006, pp. 207–210.
[2] S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek, "Learning to be a depth camera for close-range human capture and interaction," *ACM Trans. Graphics (TOG)*, vol. 33, no. 4, p. 86, 2014.
[3] https://developer.microsoft.com/en us/windows/kinect.
[4] https://www.leapmotion.com/.
[5] http://www.samsung.com/us/explore/galaxy-s-6-features-and specs/.
[6] Y. Jang, S.-T. Noh, H. J. Chang, T.-K. Kim, and W. Woo, "3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint," *IEEE Trans. VCG*, vol. 21, no. 4, pp. 501–510, 2015.
[7] D. Maier, "The complexity of some problems on subsequences and supersequences," *Journal of the ACM (JACM)*, vol. 25, no. 2, pp. 322–336, 1978.
[8] P. Mistry and P. Maes, "Sixthsense: a wearable gestural interface," in *ACM SIGGRAPH ASIA Sketches*, 2009, p. 11.
[9] D. Mohr and G. Zachmann, "A survey of vision-based markerless hand tracking approaches," *CVIU*, 2013.
[10] T. Niikura, Y. Hirobe, A. Cassinelli, Y. Watanabe, T. Komuro, and M. Ishikawa, "In-air typing interface for mobile devices with vibration feedback," in *ACM SIGGRAPH Emerging Technologies*, 2010, p. 15.
[11] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 677–695, 1997.
[12] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM Trans. Graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 313–318.
[13] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.
[14] S. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 42, no. 1, pp. 97–101, 2000.
[15] J. Romero, D. Kragic, V. Kyrki, A. A. Argyros, and L. Sweden, "Dynamic time warping for binocular hand tracking and reconstruction." in *Proc. ICRA*, 2008, pp. 2289–2294.
[16] J. Song, F. Pece, G. Sörös, M. Koelle, and O. Hilliges, "Joint estimation of 3d hand position and gestures from monocular video for mobile interaction," in *Proc. ACM CHI*, 2015, pp. 3657–3660.
[17] J. Song, G. Sörös, F. Pece, S. R. Fanello, S. Izadi, C. Keskin, and O. Hilliges, "In-air gestures around unmodified mobile devices," in *Proc. ACM UIST*. ACM, 2014, pp. 319–329.
[18] L. Song and M. Takatsuka, "Real-time 3d finger pointing for an augmented desk," in *Proc. Sixth Australasian conf. on User interface-Volume 40*, 2005, pp. 99–108.
[19] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE CVPR*, 2015, pp. 824–832.
[20] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *Proc. ICCV*, 2015, pp. 3325–3333.
[21] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graphics (TOG)*, vol. 33, no. 5, p. 169, 2014.
[22] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen, "Lucid touch: a see-through mobile device," in *Proc. ACM UIST*, 2007, pp. 269–278.