A Method for Single Hand Fist Gesture Input to Enhance Human Computer Interaction

Tao Ma[1], William Wee[1], Chia Yung Han[2], and Xuefu Zhou[1]

[1] School of Electronic and Computing Systems, University of Cincinnati, USA
[2] School of Computing Sciences and Informatics, University of Cincinnati, USA
mata@mail.uc.edu, {weewg,han,zhoxu}@ucmail.uc.edu

Abstract. The study of detecting and tracking hand gestures in general has been widely explored, yet the focus on fist gesture in particular has been neglected. Methods for processing fist gesture would allow more natural user experience in human-machine interaction (HMI), however, it requires a deeper understanding of fist kinematics. For the purpose of achieving grasping-moving-rotating activity with single hand (SH-GMR), the extraction of fist rotation is necessary. In this paper, a feature-based Fist Rotation Detector (FRD) is proposed to bring more flexibility to interactions with hand manipulation in the virtual world. By comparing to other candidate methods, edge-based methods are shown to be a proper way to tackle the detection. We find a set of "fist lines" that can be easily extracted and be used steadily to determine the fist rotation. The proposed FRD is described in details as a two-step approach: fist shape segmentation and fist rotation angle retrieving process. A comparison with manually measured ground truth data shows that the method is robust and accurate. A virtual reality application using hand gesture control with the FRD shows that the hand gesture interaction is enhanced by the SH-GMR.

1 Introduction

Hand gesture recognition is a mathematization of the interpolation of human hand gestures assisted by modern computer technology. The purpose is to replace traditional input devices, keyboard and mouse, with a new fashion that makes human interact with computer in a way that is as natural as in the real world [1]. In spatial domain, static hand gestures are recognized due to the different spatial distribution of fingers with respect to palms. A "thumb-up" gesture in sign language means "good". A "fist" gesture might stands for "stop" [2] or other meanings [3]. Laura et al presented a joint segmentation and adaptive classifier that can discriminate 4 static hand gestures under slight occlusion condition [4]. Yi et al.'s classifier that combined both supervised and unsupervised training process recognizes 14 hand gestures [5]. On the other hand, in temporal domain, hand kinematics under various gestures is meaningful as well. Yi and Thomas [6] described articulated hand local motion for a 16 rigid object 3D hand model with inverse kinematics. Human hand motions are very complicated and always occur in both spatial and temporal domain. A well representation of hand

M. Kurosu (Ed.): Human-Computer Interaction, Part IV, HCII 2013, LNCS 8007, pp. 291–300, 2013.

gesture patterns and their kinematics lies in the improvement of natural user experience-as if users interact with the real world. However, current gesture recognition methods can hardly capture all subtle movements for manipulation in the virtual world.

Holding an object and moving it with single hand is a common activity in daily life. Clenching fingers together to form a fist is a natural gesture, which represents grasping [7]. The ability to map this activity in designing interactions that allow inputs for software applications can be very useful. However, the processing of images of hand fist and characterization of the various motions as input to computer is not straightforward. We term this motion behavior as single hand grasping-moving-rotating (SH-GMR). These three actions always occur simultaneously. The detection of moving, or simple translation, is fairly easy, attested by the fact that a variety of algorithms already exist [8]. However, the detecting and tracking on fist rotation has been lacking for a long time. The existing compromised solution of rotating a virtual object is to make use of two hands to decide the rotation angle, which includes rotation about an axis and "steering wheel" rotation [9]. The drawbacks are obvious. Two hands have to grasp the same object at the same time. For tiny objects, it needs supplementary visual guidance for users to hold, such as virtual handles attached on the objects. More importantly, this two-hand gesture prevents users to interact with two objects at a time, which makes the interaction manners of many kinds quite awkward and inefficient. Thus, user experience suffers greatly. With these concerns, we argue that the single hand rotation is better than the two-hand rotation. Moreover, the study of fist rotation is crucial to achieve SH-GMR.

From the image and vision perspective, the fist rotation is defined as a 3 dimensional (yaw, pitch and roll) rotations of a deformable, scale variable and intensity variable object with associated translation movements. It is necessary to distinguish fist from other gestures because other gestures extend at least one finger out [10]. But fist gesture is defined as a hand with all fingers clenched into the palm [11]. Within certain angle of view, fingers are still visible, but they are fully folded and placed side by side with each other. In the following sections, we discuss several methods that can potentially be used for the fist rotation extraction and then give a proper solution to tackle this problem.

2 Related Research

There are several methods that could potentially be used as fundamental methods of FRD. They are skeletal model, volume model, optical flow, local features, and edge features. We review them below and give examples if necessary.

Current skeletal models for kinematic representation of hand are applied for the open handed gesture that at least one finger is stretched out. Lee and Kunii [12] introduced a 27 DOFs hierarchical skeletal hand model with constraints on joint movements, which makes some hand configurations impossible so as to reduce the shape ambiguity. Du and Charbon proposed a 30 DOFs skeletal model for depth image fitting [13].These models are suitable for describing the clenched fist, but they

did not propose how to apply the model fitting algorithms in extracting the fist rotation using their internal constraints and external image forces.

3D dense map, sometimes called point cloud generated by structured light [14], TOF camera or stereo camera, can be used to solve the ambiguity in certain degree. But huge computational cost of 3D fitting prevents them from being applied in real-time tracking [15]. We have not seen any solution that can handle the fist rotation problem. Besides, the depth map captured by current cameras cannot provide enough details in resolution to extract individual fingers from a clenched fist.

Due to the complexity of hand movements, local features should be found that are invariant to hand translation, scaling, rotation, and invariant to illumination changes and 3D projections. We applied the Scale Invariant Feature Transform (SIFT) method [16, 17] to fist rotation detection to see whether it works properly. We test the hand feature matching problem using SIFT demo program developed by Lowe. Our goal is to see weather SIFT can constantly track the same features in continuous frames as long as they are visible. The matching features are connected by straight lines in azure in Figure 1. In each image pair in Figure 1, hand on left side is always the 1th frame of the video; right hands are in the 2nd, 5th, 15th, 30th, 45th, and 65th frames respectively. SIFT finds most of feature pairs correctly, but the number of points reduces sharply as the angle difference becomes larger. The method casts away most of detected features to keep a correct matching, which is not suitable for the extraction.

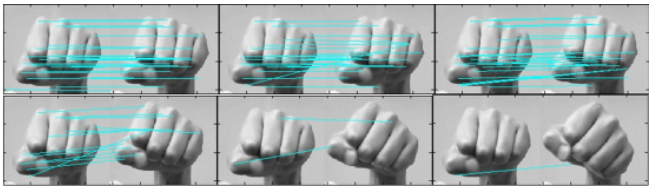
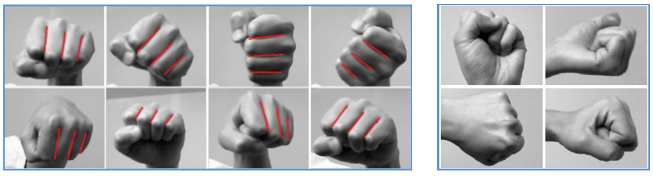


Fig. 1. SIFT matching under different fist angles, between 1st frame and 2nd, 5th, 15th, 30th, 45th, 65th frames



(a) (b)

Fig. 2. Edge features on fists. (a) Manually labeled fist lines under different rotations. (b) Hand gestures that do not fully show the fist lines.

It is easy to find that when five fingers are clenched, brightness between two fingers is darker. Between index, middle, ring and pinky fingers, 3 clearly dark lines can be seen under most lighting conditions. These lines are nearly straight, parallel, and almost appear or disappear at the same time. During the rotation, they maintain their

relative positions unchanged no matter whether fist is facing directly to the camera or not. They are also good to preserve the fist structure. They are more stable and accurate to be tracked, compared with methods listed above. Moreover, instead of extracting relative angle value from frame-to-frame computation, the angle of the lines are absolute value with respect to camera coordinates, which means no cumulative error would be established. We call these 3 lines as "fist lines" for simplicity. As shown in Figure 2(a), fist lines are manually labeled as red color in various fist rotations. The fist lines will not appear under certain field of views. Figure 2(b) shows some fist gestures in which the fist lines are not clear or cannot be found. So far, the edge features turn out to be the best approach to handle the fist rotation.

3 Approach

In this paper, it is assumed that human arms have been segmented from the whole images, and also hands are in fist shape. Arm segmentation from other parts of the human body and fist shape classification from other gestures are beyond the scope of our interests.

3.1 Fist Shape Segmentation

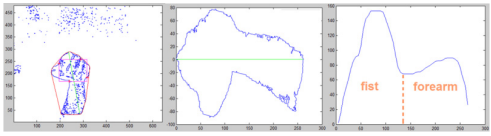
Observation shows that, if seeing along a human arm, the width of the fist is always larger than that of the forearm under all camera views. Values of the width would have a suddenly drop down if sliding from fist side to arm side. This geometrical characteristic is feasible for the fist segmentation. Arm shapes in 2D are first transformed to 1D representation through a dimension reduction process, and then a classifier is used to decide the fist position along the arm.

A contour retrieving algorithm is applied to topologically extract all possible contours in the image [18]. *Contour C* with the largest number of point set is the outermost contour of the arm, shown as Figure 3 (a). Using the data set of the *contour C*, a convex hull and its vertex set *P* [19] are computed. Sometimes image noise causes trivial boundary so that the number of vertices is sharply increased. In this case, a polygon approximation routine is used to reduce the excessive details along the boundary. The number of vertex should better be in the range of 8 to 15 considering both computational cost and accuracy. We compute the Euler distances of all vertex pairs except those who are adjacent. Then we find the longest two distances $\vec{a}$ and $\vec{b}$. The direction of the main axis *l* is set to be the bisector of the angle of the two vectors:

$$\theta_{ma} = \frac{1}{2} a\cos\left(\frac{\vec{a}\cdot\vec{b}}{|a||b|}\right) \tag{1}$$

Two longest distances are used to decide the main axis to prevent the value oscillation introduced by noise. Then, the whole points in *contour C* are rotated and translated:

$$contourC = \begin{bmatrix} \cos\theta_{ma} & -\sin\theta_{ma} \\ \sin\theta_{ma} & \cos\theta_{ma} \end{bmatrix} contourC + T \tag{2}$$



(a) (b) (c)

Fig. 3. Fist segmentation. (a) the arm contour, convex hull, main axis, and the search window marked on the contour image, (b) a fist contour that has been rotated and translated to horizontal position, (c) width curve of the contour in (b).

The rotated main axis l' is located on x-coordinate, shown in Figure 3 (b). Given a x on the rotated contour, two y values are corresponded: one negative and one positive. The difference of these two y values indicates the width of the fist along l'. Computing all the widths along l', we get a smoothed width curve of the *contour C'*, as shown in Figure 3(c). So far, we convert a 2D shape clustering problem to 1D. It is easy to classify fists from forearms by looking at features along the width curve. Various clustering methods can be used, such as K-means, area-based, and etc. The curve on Figure 3 (c) shows that the fist is located at the first half of the contour of Figure 3 (b). Going back to the original *contour C*, a fist bounding box is found according to the result of the width curve, shown in Figure 3 (a) in a magenta rectangle. The bounding box is served as the search window for the fist detection.

3.2 Fist Rotation Detection

Finding the three fist lines is a challenging task for the reason that there are many line and curve features in the search window. Inspired by the observation shown in Figure 2, we find the fist lines are basically straight, parallel, and almost appear or disappear at the same time. Thus, three parallel straight lines with the interval of d are used as the theoretical model to fit the selected feature point data. 3 parameters need to be decided: the slope of the lines θ, the intercept of the middle line b, and the interval d. Note that even though the fist lines are equidistant, their distances appearing on images may not be the same due to the perspective from 3D space to the camera plane. But in this paper, we particularly see the roll rotation as the major direction meanwhile ignoring other DOFs. The mathematical model is:

$$\begin{cases} y = x\tan\theta + b + d/\cos\theta; \\ y = x\tan\theta + b; \qquad\qquad where \quad -90 < \theta < 90 \\ y = x\tan\theta + b - d/\cos\theta \end{cases} \tag{3}$$

Edge Feature Extraction. We use Laplacian of Gaussian (LOG) [20] method to extract features in the search window because it is scale sensitive to blobs that has the similar size. It has strong response to features of extent $\sqrt{2\sigma}$, where σ is the variance of Gaussian function. The LOG kernel can be pre-computed before the convolution on the original image:

$$\nabla^2 G(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

Since features around fist lines are very similar, the LOG method extracts stable and mostly continuous lines between fingers. But it also recovers features located on hand edges. Figure 4 shows edge feature maps under different fist rotations. All the edges are stored into a structured 2-row array E. Mathematically, this edge set E is $E = \bigcup_N \varepsilon$, where N is the number of feature edges ε.

A Rough Angle Estimation with Histogram. The edge feature maps gives us an important clue that after the feature extraction the amount of features on the fist lines is larger than that on other non-fist-line features. It is because the fist segmentation preserves most of fist lines while eliminating most of unrelated features. Figure 4 shows that the distribution of the histograms of the feature maps is highly related to the fist angles. We can approximately compute the coarse fist angle range δ by finding the highest percentage of pixel bins within the histogram of the slopes. The highest bins and its two nearest neighbors are picked to calculate the angle range using the center of gravity method. To compute the slope of the feature segments, the step length between two points should be larger than 5 so as to provide plenty of angle resolution.

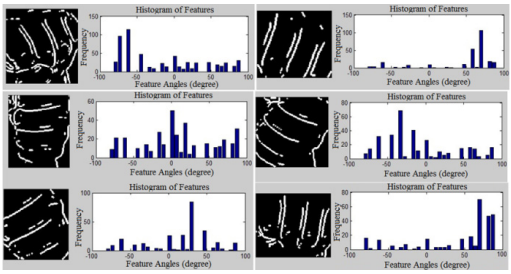


Fig. 4. Feature maps and their corresponding histogram under different angles

Back Projection and Edge Pruning. All pixels within the angle range δ are back-projected to the edge feature map. Edges that contain these pixels are marked and kept in Ω, while other edges are pruned:

$$\Omega = \bigcup_{\varepsilon \in E} [\varepsilon \mid \exists \gamma_\varepsilon :\ \gamma_\varepsilon \in \delta] \tag{5}$$

where γ_ε is the angle value of the feature segments in edge ε. After this process, the number of edge candidates is greatly reduced.

Cutting off, Merging, and Sorting Operation. In Ω, the slope of features within one edge may go out of the angle range. These parts are cut off from the edge and the residues are merged again, indicated as:

$$\varepsilon' = \bigcup_{\chi_\varepsilon \in \varepsilon} [\chi_\varepsilon \mid \gamma(\chi_\varepsilon) \in \delta] \tag{6}$$

where χ_ε is a feature point in ε, and $\gamma(\chi_\varepsilon)$ is the angle value of χ_ε.

If two edges are almost collinear as if they are in the same fist line, they are merged into one edge, described as:

$$\varepsilon_{12} = \bigcup_{\substack{\chi_{\varepsilon_1}\in\varepsilon_1 \\ \chi_{\varepsilon_2}\in\varepsilon_2}} [\chi_{\varepsilon_1}, \chi_{\varepsilon_2} \mid \forall\chi_{\varepsilon_1}, \forall\chi_{\varepsilon_2} : collinear\,(\chi_{\varepsilon_1}, \chi_{\varepsilon_2})] \tag{7}$$

where ε_{12} is the merged feature set from ε1 and ε2, and *collinear( )* is a function that decide whether two feature points are basically collinear.

Last, all the existing edges are sorted according to their positions, and then stored in set Υ. So far, the number of edges in Υ is slightly more than 3, which is very cost effective for the following angle refining process.

Fitting the Mathematical Model with the 3 Selected Edges. For any given 3 edges, $\varphi_1, \varphi_2, \varphi_3$ in Υ, parameters θ, b, and d can be calculated by fitting the theoretical model described in equation (3) to the three edges. To convert equation (3) into linear equations, we let $k = tan\theta$, and $c = d/cos\theta$. Then the equations can be expressed with linear equations as:

$$\begin{aligned} &A\,\tilde{x} = \tilde{y}, \quad where \\ &A = [k, b, c], \quad \tilde{x} = \begin{bmatrix} x_{11}\ x_{12} & x_{21}\ x_{22} & x_{31}\ x_{32} \\ 1\ \ 1 \cdots 1\ \ 1 & \cdots 1\ \ 1 & \cdots \\ 1\ \ 1 & 0\ \ 0 & -1 -1 \end{bmatrix} \\ &\tilde{y} = [y_{11}, y_{12}, \cdots y_{21}, y_{22} \cdots y_{31}, y_{32} \cdots], \\ &(x_{1i}, y_{1i}) \in \varphi_1, \ (x_{2i}, y_{2i}) \in \varphi_2, \quad (x_{3i}, y_{3i}) \in \varphi_3. \end{aligned} \tag{8}$$

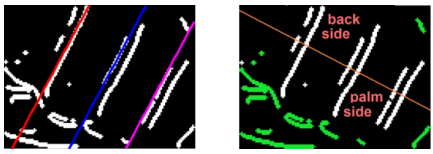
Several methods can be used to solve this over-determined, multiple linear regression problem, such as least square, Gauss elimination, and Singular value decomposition (SVD). The fitting error $E(\varphi_1, \varphi_2, \varphi_3)$ can be derived from the sum of absolute difference (SAD) between fitted lines and edge pixels:

$$E(\varphi_1,\varphi_2,\varphi_3) = \sum_{j=1}^{3} \sum_{(x_{ji}, y_{ji})\in\varphi_j} \frac{\left| k\,x_{ji} - y_{ji} + b + (2-j)c \right|}{\sqrt{k^2+1}} \tag{9}$$

Optimized Fist Lines with Minimum Error. We compute all the combinations of 3 possible fist lines in Υ. The number of combination is given by C_n^3, where n is number of edges in Υ. A correct choice of the fist lines is indicated by $(\varphi_1{*}, \varphi_2{*}, \varphi_3{*})$ that has the minimum fitting error:

$$(\varphi_1{*}, \varphi_2{*}, \varphi_3{*}) = \underset{\varphi_1, \varphi_2, \varphi_3 \in \Upsilon}{\arg\min}\ E(\varphi_1, \varphi_2, \varphi_3) \tag{10}$$

Its angle $\theta^* = atan(k^*)$ is the optimized fist rotation angle within $(-90^o, 90^o)$. A large amount of pixel involved in the fitting process guarantees an accurate and stable outcome. Figure 5(a) shows the three fitting lines are found, marked with red, blue, and magenta.



(a) (b)

Fig. 5. (a) Three fitting lines that minimum the fitting error. (b) Feature points out of the angle range marked with green color.

Deciding the Fist Rotation within 360°. As mentioned above, the line model can only decide rotation within $(-90^o, 90^o)$. Due to the special finger position with relation to the palm, more features can be found near the palm side rather than the back side of the fist. These features mostly have different directions with the fist lines. We empirically discriminate between the palm side and the back side by measuring the distribution difference of features that are out of the range of the rough rotation angle δ. They are marked as green color in Figure 5(b). The center of gravity of the selected 3 fist lines are first computed. Then, through this point, a straight line (the orange line in Figure 5(b)) that is perpendicular to the fist lines splits the search window into two parts. The palm side and back side of hands must be located in these two parts respectively. The part that has more green pixels is the palm side, and vice versa. With the angle θ^* computed in the previous steps, the final rotation angle can be decided within 360^o.

4 Experiment and Application

We pay mostly attention to the accuracy and stability of the proposed FRD. One experiment is implemented to test these two aspects. To generate ground truth (GT) data, two markers in cross shape are stuck on the middle finger so that they can be manually labeled afterwards and be used to calculate hand rotations, shown as Figure 6. Then the GT angle is compared with the angle generated by the FRD every 10 frames.

The GT angle is manually computed every 10 frames. Then it is compared with the FRD output within the same frame, shown as Figure 7. The maximum angle value is 140o due to the physical limit of human hands. The result shows that the proposed FRD method is stable and consistent with the GT data, with the absolute mean difference of 3.27o (0.9% of 360o), and standard deviation of 2.81o (0.8% of 360 o). The largest error occurs between 110o and 140o.

There are several reasons that cause the error. First, the manually labeled GT value may not be accurate due to the image quality. Then, remember that hand is a deformable object. The rotation of the markers may not fully represent that of the fist lines, especially when the hand is twisted almost to its physical limit. Last, as analyzed in previous sections, hand rotation always happens in 3 DOFs. The proposed FRD and the GT measurement only consider one major movement while ignoring others. This will also introduce difference in the comparison.

Considering the real applications of fist rotations in HMI and the accuracy level of human body movements, the proposed FRD is effective to be used as detecting human fist rotations in HMI applications. The computational cost is various depending on the amount of feature points. In our test video with the resolution of 640×480, the size of the search window after the fist shape segmentation is usually within 120×120. Modern computer can easily handle this amount of data in real time.

To illustrate the improvement of the SH-GMR behaviors with the help of the proposed FRD algorithm, we present a simple application that implements a chemical reaction experiment in a 3D virtual reality environment. This system captures 3D hand movements with stereo cameras. With our FRD routine integrated, the system is able to handle SH-GMR. Figure 8 shows a user is implementing a chemical experiment by pouring one sort of liquid from the right flask into the right flask to trigger certain chemical reaction. As shown in Figure 8 (b), in the virtual environment, two flasks are moved close to each other, and right flask is leant to pour the liquid into the left one, with the operation of two hands respectively.

Fig. 6. Two markers stuck on a hand for computing GT angle.

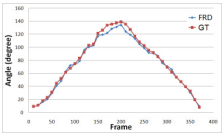
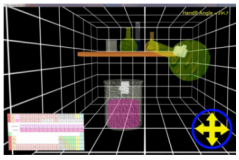


Fig. 7. The comparison of the FRD and GT for every 10 frames

(a)



(b)

Fig. 8. (a) The user's gesture of operating objects in the virtual reality environment. (b) An ongoing virtual chemical experiment controlled by two hands.

5 Conclusion

Hand interaction is highly limited by the current two-hand rotation gesture due to the lack of the research on hand fist kinematics. A single fist rotation detector is crucial to implement single hand grasping-moving-rotating activity that makes two hands fully control different objects possible. We present a feature-based FRD method that provides accurate and stable detection of the fist rotation problem for the purpose of enriching hand gesture databases with finer hand motion sequences. Except the fist rotation, there are plenty of hand gestures and their kinematics that we have not fully utilized. Deeply digging into this area will greatly benefit the hand gesture interaction and also bring user experience to a brand new level.

References

1. Beale, R., Edwards, A.D.N.: Gestures and Neural Networks in Human-computer Interaction. In: IEE Colloq. Neural Nets in HCI (1990)
2. Manresa, C., Varona, J., Mas, R., Perales, F.J.: Real-Time Hand Tracking and Gesture Recognition for Human-Computer Interaction. In: ELCVIA (2000)
3. Binh, N.D., Shuichi, E., Ejima, T.: Real-Time Hand Tracking and Gesture Recognition System. In: GVIP (2005)
4. Gui, L., Thiran, J.P., Paragios, N.: Joint Object Segmentation and Behavior Classification in Image Sequences. In: CVPR (2007)
5. Wu, Y., Huang, T.S.: View-independent Recognition of Hand Postures. In: CVPR, vol. 2, pp. 88–94 (2000)
6. Wu, Y., Huang, T.S.: Capturing Articulated Human Hand motion: A Divide-and-conquer Approach. In: ICCV, vol. 1, pp. 606–611 (1999)
7. Hooker, D.: The Origin of the Grasping Movement in Man. In: Proceedings of APS, vol. 79(4), pp. 597–606 (1938)
8. Bradski, G., Kaehler, A.: Learning OpenCV, 2nd edn. O'Reilly Media (2008)
9. Hinckley, K., Pausch, R., Proffitt, D., Kassell, N.F.: Two-Handed Virtual Manipulation. In ACM Trans. CHI 5(3), 260–302 (1998)
10. Triesch, J., Malsburg, C.V.D.: A System for Person-Independent Hand Posture Recognition against Complex Backgrounds. IEEE Trans. PAMI 23(12), 1449–1453 (2001)
11. Kingston, B.: Understanding Joints: A Practical Guide to Their Structure and Function, 2nd edn. Nelson Thornes (2000)
12. Du, H., Charbon, E.: 3D Hand Model Fitting for Virtual Keyboard System. Applications of Computer Vision. In: IEEE WACV (2007)
13. Rosales, R., Athitsos, V., Sigal, L., Sclaroff, S.: 3D Hand Pose Reconstruction Using Specialized Mappings. In: ICCV, vol. 1(200), pp. 378–385 (2001)
14. Kollorz, E., Penne, J., Hornegger, J., Barke, A.: Hand Gesture Recognition with A Novel IR Time-of-Flight Range Camera-A pilot study. Int. Journal of Intelligent Systems Technologies and Applications Archive 5(3/4), 334–343 (2008)
15. Bruce, L.D., Takeo, K.: An Iterative Image Registration Technique With An Application to Stereo Vision. Proceeding IJCAI 2, 6474–6679 (1981)
16. Lowe, D.G.: Object Recognition from Local Scale-invariant Features. In: IEEE Conf. CV, vol. 2, pp. 150–1157 (1999)
17. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. Journal of Intelligent Systems Technologies and Applications Archive 60(2), 91–110 (2004)
18. Homma, K., Takenaka, E.I.: An Image Processing Method for Feature Extraction of Space-occupying Lesions. JNM 26, 1472–1477 (1985)
19. Marr, D., Hildreth, E.: Theory of Edge Detection. Defense Technical Information Center (1979)
20. Suzuki, S.: Topological Structural Analysis of Digitized Binary Images by Border Following. ACM CVGIP 30(1), 32–46 (1985)