World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:11, 2011

Gesture Recognition by Data Fusion of Time-of-Flight and Color Cameras

Piercarlo Dondi, Luca Lombardi, and Marco Porta

Abstract—In the last years numerous applications of Human-Computer Interaction have exploited the capabilities of Time-of-Flight cameras for achieving more and more comfortable and precise interactions. In particular, gesture recognition is one of the most active fields. This work presents a new method for interacting with a virtual object in a 3D space. Our approach is based on the fusion of depth data, supplied by a ToF camera, with color information, supplied by a HD webcam. The hand detection procedure does not require any learning phase and is able to concurrently manage gestures of two hands. The system is robust to the presence in the scene of other objects or people, thanks to the use of the Kalman filter for maintaining the tracking of the hands.

Keywords—Gesture recognition, human-computer interaction, Time-of-Flight camera.

I. Introduction

A Time-of-Flight (ToF) camera is a particular kind of active sensor able to supply in real-time depth measures of an environment. Since the introduction of the first models in 2003 [1], researchers have achieved interesting results with these new devices, in particular in computer vision and computer graphics [2]. One of the most active field is certainly Human-Computer Interaction (HCI), where many different solutions have been proposed in the last years, especially for gesture recognition.

This paper presents a new kind of interaction that allows a user to move a virtual object in a 3D space. The proposed method uses as input the data supplied by a ToF camera and the color information supplied by a traditional HD webcam. The procedure can be subdivided in two main steps: firstly, depth data are used for recognizing the entire body of the user and for limiting the interest area; then, color is introduced to refine the results and to extract specific details from the retrieved cluster (in this case, the hands). The recognition of specific gestures of the hands allows the user to translate, rotate, and zoom 3D objects in real-time. Hand detection is based on the analysis of generic chromatic characteristics, whereas gesture recognition is based on geometrical transformations of the clusters − so the algorithm does not need any learning phase.

Experimental tests show that the system is robust to the presence of false positive clusters (e.g. user's head or other people), thanks to the adoption of well selected distance thresholds and to a tracking method based on Kalman filter.

A first implementation of our system for gesture detection has been presented in [3], where a single hand has been used as a pointer for typing on a virtual keyboard. This work extends the previous solution, introducing a more general system for gesture recognition and the capability of concurrently managing two hands instead of one. At the same time the new method improves also the rejection rate of false positive clusters.

The paper is organized as follows: section II describes the characteristics of the adopted cameras, focusing in particular on the ToF one; section III presents some previous work concerning ToF-based HCI applications; section IV analyzes our segmentation and hand detection algorithms; section V describes the interaction modalities with the 3D object and the experimental tests; at last, section VI draws some conclusions.

Piercarlo Dondi, Luca Lombardi and Marco Porta are with the Department of Computer Engineering and Systems Science, University of Pavia, Pavia, Italy (e-mails: piercarlo.dondi@unipv.it; luca.lombardi@unipv.it; marco.porta@unipv.it).

II. Cameras Specifications

ToF cameras are active imaging sensors that provide distance measures using laser light in the near-infrared spectrum. Two main technologies are used for these devices: pulsed light and modulated light. In the first case a coherent wavefront (similar to a "light wall") hits the targets and then the distances are measured analyzing the deformation in the reflected "wall". In the second case, the camera emits a modulated light and the depth data are gained by phase delay detection. Currently the latter is the most widespread technology, adopted by the great part of manufacturers (Canesta, PMDTec and Mesa Imaging).

There are both pros and cons in the use of a ToF camera respect to other depth devices such as stereo cameras or laser scanners. A ToF camera can work in real-time, since distances are directly supplied by the sensors without complex additional computation. There is no need for external reference points or color contrast to estimate distances, and the shape of objects does not influence measures. The illumination of the scene is self-provided, so external illumination is superfluous. On the other hand, ToF cameras still have a limited resolution and are affected by different kinds of noise. The most notable ones are "flying pixels", caused by areas with abrupt changes in depth (e.g. the corners of an object), "motion artifacts", generated by moving subjects, and multipath scattering [2]. The device is designed only for an indoor use: even if artificial light does not influence the sensors, the presence of sunlight introduces significant alterations. Moreover, the precision of measures strictly depends on the reflectivity of objects: if it is too high it can saturate the sensor, while if it is too low the object may not be correctly detected.

The two cameras used in this work are a SR3000, developed by Mesa Imaging, and a Logitech HD Pro Webcam C910 (Fig. 1). The first one is a modulated-light ToF camera whose active leds emit an infrared laser light around 850nm with a

frequency of 20MHz. Its useful range, without ambiguities, is up to 7.5m, while its frame rate is about 18-20 fps. SR3000 provides two maps per frame, both with a QCIF resolution (176x144 pixels): the first one contains distance data; the other represents the intensity of reflected light. Intensity values depend only on near-infrared light, since the sensor is insensible to visible light.

The second camera is instead a standard webcam, used with a resolution of 640x480 pixels. The C910 can reach 30 fps, but in our case the recordings have been made at the same speed of SR3000 for maintaining the synchronization of the two video streams. The calibration of the two sensors is achieved with a solution similar to that presented by Reulke in [4], able to provide a direct mapping of color image on the depth map. Instead of ortophoto generation approach used by Reulke, a less precise but more faster perspective transformation has been performed, with the purpose to reduce the computational time. Possible slight misalignments between color and ToF images, due for example to rapid user movements or to clusters too close to the cameras, do not significantly influence the precision of hand detection.

Fig. 1. The two cameras used in the experiments: the SR3000 (bottom) and the Logitech HD Pro Webcam C910 (top)

III. State of Art

In the last years research has shown a great interest in gesture recognition applications with ToF cameras. The system proposed in [5], for example, carries out gesture recognition with a two step procedure: firstly the range and the intensity images are fused in a new data set (the "phase" image); then this new image is used as an input for a segmentation method based on the combination of two clustering approaches, namely K-Means and Expectation Maximization. This approach has been used also in robotics, for controlling an industrial robot [6].

One of the most used techniques for hand movement detection is certainly the Principle Component Analysis (PCA). In [7] the PCA technique allows to obtain an approximate estimation of the location and orientation of the hand. Subsequently, the result is refined by fitting the retrieved data with a complex 3D hand model derived from a physically-based muscle simulation and elastic skin attributes. The procedure is simplified by initially discarding all the elements that fall outside a predefined depth threshold. In [8] the PCA technique is exploited to perform head and hands tracking in 3D space. The Head-Finger Line (HFL), determined by estimating face and fingertips orientation in the 3D space, and the orientation of the forearm are used to assess the pointing direction. The direction of the forearm is determined using PCA, while hand tracking is implemented using a Particle Filter applied to foreground color images. In the context of augmented reality applications, a method to estimate hand position while behind an interactive display is presented in [9]. Here PCA is used to model the hand as an oriented box, which is then provided to and processed by a "physics engine".

A system able to recognize twelve different static gestures is presented in [10]. The gestures are classified by the projection of the hand onto the X and Y axes of the image, while the arm area is removed. Depth data of the ToF are taken into account to solve ambiguities such as gestures with same projections but different alignments. View invariant recognition of dynamic arm gestures is presented in [11], where motion is perceived by firstly differencing range images and then applying a bandpass filter. The system is made invariant to rotation around the vertical axis. A more complex technique, that involves also the use of an RGB camera, is described in [12]. Here, a pre-trained skin color model (generated with Gaussian mixture approach) is combined with a histogram-based adaptive model dynamically updated using the color information extracted from the face. This first segmentation is mixed with a depth-based one to obtain a more robust result capable to manage worst conditions such as hand overlapping with the face or people in the background. Also, gesture recognition is achieved by combining color and depth data. The system recognizes six different hand postures and can identify gestures and movements of both hands. This procedure is used to manipulate 3D models in real-time.

Another typical application of gesture recognition is the slideshow control presented in [13], where the "thumbs-up" gestures, toward left or right, are used to move between slides, while a finger indicating the screen is interpreted as a "virtual laser pointer". The pointing direction is calculated in 3D, at first through a segmentation of the person in front of the camera with respect to the background, and then by detecting the 3D coordinates of head and hand. An analogous use for the interaction with a beamer projector is described in [14]. An algorithm exploiting fingertips to draw information about the hand is described in [15]. The perceptive interface is able to recognize single stroke gestures composed of the numbers zero to nine, and hand gesture input can be disabled by simply forming a fist.

A further context is represented by medical applications, where it is often necessary to use interfaces which avoid physical interaction with an input tool. A ToF camera can be very useful to this purpose, because it allows relatively easy implementations of touch-less interaction. In [16], five different gestures are considered, namely "translation", "cursor", "click", "rotation" and "reset". After performing a coarse segmentation of the hand, the separation between hand and forearm is carried out in the 2D domain, firstly by searching for the largest circle (which corresponds to the palm) that contains only foreground pixels, and then by iteratively augmenting the size of the circle and analyzing the intersections with foreground pixels themselves. The final result allows the exploration and navigation through 3-D medical image data.

A similar solution is proposed in [17], where hand gestures are used to control the mouse cursor or manipulate medical images. Hand segmentation, in this case, occurs by simply thresholding distance and amplitude data.

IV. Hand Detection

The high variability of hand shapes makes hand detection a complex problem. An approach based only on color or only on depth may be easier, but in both cases the obtained performances are not good due to different kinds of imprecisions. In a color-based approach, background objects with colors similar to the skin generate false positives; on the other hand, in a depth-based solution a hand can be distinguished without ambiguities only if its position is always far enough from the rest of the body. The adoption of an RGBZ space (color plus distance) can compensate the respective weaknesses of the devices and can reach precise results with a limited computational cost.

The proposed solution starts from a ToF-based foreground segmentation method that retrieves the entire body of the user, followed by a color-based refinement that extracts the possible hand candidates from the cluster. The whole procedure does not require any learning phase for the algorithm, nor a priori knowledge of the environment.

A. ToF based Foreground Segmentation

Our foreground segmentation algorithm can be subdivided into two main phases: firstly a thresholding of the depth map, based on the corresponding values of the intensity map, reduces the area of interest; then, a region growing, that starts from seeds planted on peaks of the intensity map, completes the procedure. The method is based on the consideration that foreground objects receive more light than those in background, so they result more evident in the intensity map.

A proper intensity threshold (λ_{seed}) is used to find the best seeds for region growing. The value of the threshold is estimated for every frame using the Otsu's method.

The set of seeds S is defined as follows:

$$\{I_x > \lambda_{seed},\ \|x - s\| > \gamma,\ \gamma > 1\} \rightarrow \{x \in S\} \quad (1)$$

where x is a point of the distance map, I_x is its corresponding intensity value and s is the last seed found. The control about the distance between seeds guarantees a better distribution of them on the image and also reduces their number, in order to decrease the time needed for the growing step. Theoretically, only one seed per cluster is needed; the relative position of the seed on the cluster is not relevant for a correct region growing.

Formula (2) defines the similarity measure S between a cluster pixel x and a neighboring pixel y.

$$S(x, y) = |\mu_x - D_y| \quad (2)$$

D_y is the distance value of pixel y and μ_x is a local parameter related to the mean distance value around x (6). The lower S is, the more similar pixels are. When a seed is planted, μ_x is initialized to D_x. With a 4-connected neighborhood, a pixel x, belonging to a cluster C, absorbs a neighbor y if it complies with the following criteria:

$$\{x \in C,\ S(x, y) < \theta,\ I_y \in L\} \rightarrow \{y \in C\} \quad (3)$$

where L is a set of intensity points generated using (4) and (5), designed to threshold the data so as to compensate some of the distortions caused by sunlight; θ is a constant parameter, experimentally estimated, related to clusters separation.

$$\{(I_y \geq \lambda) \vee [(I_y < \lambda) \wedge (I_{8n} > \beta * \lambda)],\ \beta \in [0, 1]\} \rightarrow \{y \in A\} \quad (4)$$

$$L = A \cup M \quad (5)$$

where λ is an intensity threshold proportional to λ_{seed}, I_{8n} is the intensity of all the neighbors of pixel y considering the 8-connection, and M is the set A after the application of a series of morphological operations experimentally established (two dilations, five erosions and a final dilation, respectively).

When a neighbor y of seed x is absorbed, we compute the average distance value μ_y in an incremental manner as follows:

$$\mu_y = \frac{\mu_x * \alpha + D_y}{\alpha + 1} \quad (6)$$

where α is a learning factor of the local mean of D. If pixel y has exactly α neighbors in the cluster, and if the mean of D in this neighbor is exactly μ_x, then μ_y becomes the mean of D when y is added to the cluster.

Every region grows excluding the just analyzed pixels from successive steps. The process is iterated for all seeds in order of descending intensity. Regions too small are discarded, to remove false positive areas that pass the thresholding (for example small surfaces with a high reflectivity).

A more complete discussion of the method can be found in one of our previous works [18].

B. Sub-segmentation: Hands Extraction

The ToF segmentation retrieves all the foreground subjects in the scene. Considering that the application field of this hand detection system is the interaction with a virtual object, the user must be relatively close to the camera to see the screen; so we can exclude a priori all the retrieved clusters placed too far from the cameras (generally over 2 m). In this way error sources, e.g people moving behind the user, are automatically removed with no additional computational cost.

After these preliminary phases, the interest area is limited to a single cluster: the user placed in front to the camera (Fig. 2(a)). Color information can now be introduced to detect the hands. Firstly, the input RGB image is converted to HSV, a more suitable color model; then all the points of the cluster outside the set W (7) are left out. Set W is defined as follows:

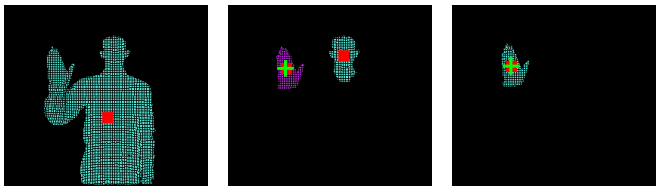
$$\{y : -10° < H_y < 10°,\ S_y > th_s,\ V_y > th_v\} \rightarrow \{y \in W\} \quad (7)$$

where H_y, S_y and V_y are, respectively, the hue, saturation and value of pixel y. The first two constraints set the color area with a hue in the skin range; the saturation threshold excludes all the white zones; finally, only points with a high value of

lightness are included, to reduce to the minimum the number of false positive clusters, e.g. clothes with skin-like colors.

The studied application does not require a precise segmentation of the hand; an approximation is usually enough (Section V-A). This simplification gives an interesting advantage in the choice of the parameters for defining W: it is possible to apply very strictly color thresholds for finding the hands – losing some details for removing certainly wrong areas is an affordable cost. Small inaccuracies, e.g. holes, are in any case fixed applying a morphological dilation on the retrieved sub-clusters.

The described procedure extracts from the original cluster (Fig. 2(a)) different sub-clusters (Fig. 2(b)), but not all of them are valid hand candidates (e.g. the head). The result can be further refined considering some physical constraints: a hand can be placed only in an area included between the body and the maximum arm extension. This useful region can be found, for each user, with a quick initialization phase. At the beginning we ask the user to stay in front of the camera and raise one hand. The system can easily distinguish between the hand (the cluster closer to the camera) and the rest of the body (the cluster in the upper position). The positions of the centroids of the hand and of the body are saved and used respectively as minimum and maximum distance thresholds. The final outcome is shown in figure 2(c): the head disappears and only the hand remains. This solution has also the additional advantage to exclude all the parts of skin-like clothes that pass the test of (7).



(a) (b) (c)

Fig. 2. Different sub-segmentation steps: (a) initial segmentation of the entire body (cluster visualized as cloud of points); (b) sub-segmentation applying only W; (c) sub-segmentation applying (8).

Better performances can be achieved applying this sub-segmentation algorithm not after the entire ToF-based foreground segmentation, but at the end of the thresholding phase: this way, a single region growing is executed on a reduced set of points. For this reason, the sub-segmentation can be considered as an extension of the standard thresholding algorithm (4) (5). The region growing method (3) can therefore be adapted as follows:

$$\{x \in C, S(x,y) < \theta, I_y \in L, y \in W, \delta_m < D_y < \delta_M\} \rightarrow \{y \in C\} \quad (8)$$

where D_y is the distance value of pixel y and δ_m and δ_M are, respectively, the minimum and the maximum distance threshold previously defined.

The selection criteria used for deciding what is/are the active cluster/s will be described in Section V. In any case, after the choice, the selected clusters are followed in the next iterations using a tracker based on Kalman filter. The tracking allows the recognition of the chosen hand/s also in presence of other moving clusters placed within the active area (Fig. 4(a)). Short term occlusions are managed exploiting the predictions supplied by the Kalman filter to estimate the more probable path of the disappeared cluster. If the cluster reappears in a position close to the predicted one, it is recognized as an active hand; if not, the system waits until another cluster respects the selection criteria.

V. Interaction with 3D Object

A. Gestures Recognition

The interaction with the virtual object (the standard Utah teapot) in the 3D space is achieved with a specific series of gestures. The available movements can be grouped into two categories: translation and rotation.

For translation, only one hand is needed. This mode is activated when the hands are not aligned (Fig. 4(a)). When that occurs, the cluster closer to the camera is chosen as an active hand, so it is followed by the tracker in the next frames in accordance with the previously described behavior. To estimate the position of the hand, we use its centroid, the most stable point in the cluster: errors in depth evaluation caused by the various noise sources mainly affect points on the edges of the objects. The system performs a mapping of the position of the hand in the camera frames to the position of the object. The ToF measures also allow a precise and fast estimation of the Z coordinate of the hand, so variations in distance are interpreted by the object as a zoom in or a zoom out command (Fig. 3, second column). In this mode, the shape of the hand is not important, as only the position of the centroid is crucial; thus, the user can assume the position that s/he finds more comfortable.

The rotation mode is activated when two hands stay aligned for at least 2 seconds (Fig. 4(b)). A rotation of the hands corresponds to a rotation of the object. In particular, the left hand controls Y-axis rotations (Fig. 3 third column) and the right hand the X-axis rotation (Fig. 3 fourth column). The rotation directions are achieved analyzing the moments of inertia of the hand. When the moment along the X-axis is greater than the moment along the Y-axis, the rotation starts; vice versa the object stops. The direction of the rotation (positive or negative) depends on the sign of the mixture moments XY. Also in this case, the two hands are tracked to avoid false positive detections during the movements. For this kind of interaction the hands must be opened with the palm facing the camera. These gestures do not involve the presence of specific elements, e.g. fingers, so also ”raw shapes” of the hands are enough for correct working. If one of the hands leaves the scene, the system starts again in translation mode, using the other hand as control.

Some visual feedbacks help the user to understand what mode is enabled and what is/are the active cluster/s: in translation mode, a green cross marks the only active hand (Fig. 4(a)); in rotation mode, a blue dot appears on the hand controlling the Y-axis rotation and a yellow dot on the other hand, controlling the X-axis rotation (Fig. 4(b)).

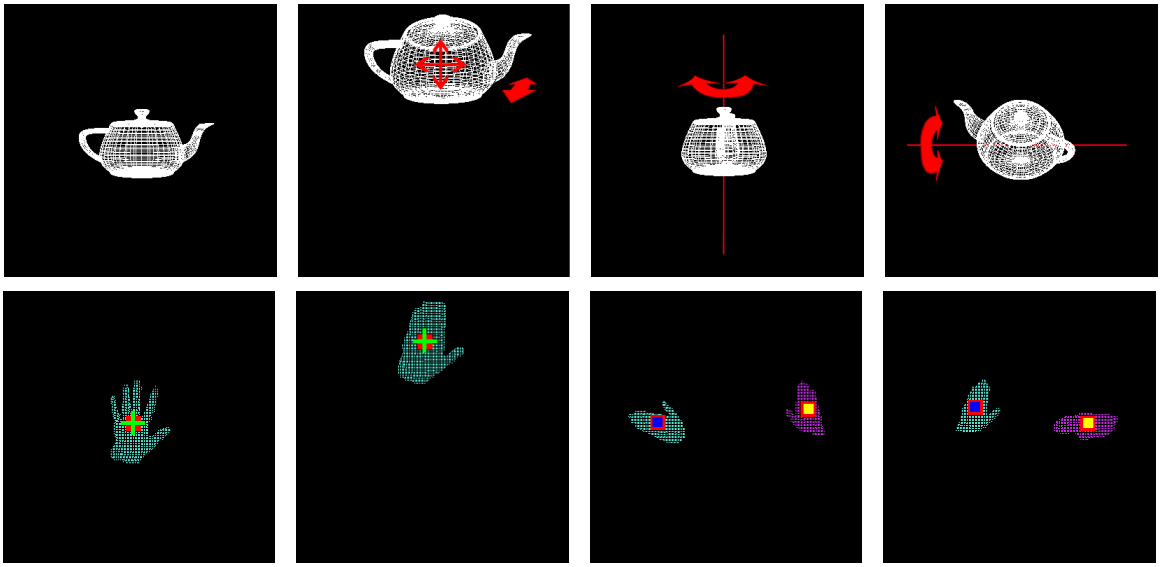
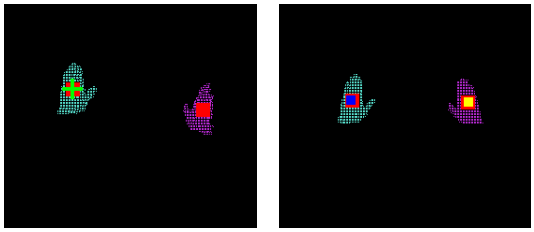


Fig. 3. Possible movements of a 3D object (top line) and corresponding gestures of the hand/s (bottom line). From left to right: starting situation, no movement, one hand selected; translation and zoom, one hand moving along axes (X,Y,Z); rotation on vertical axis, two aligned hands, the left one inclined; rotation around the horizontal axis, two aligned hands, the right one inclined.



(a) (b)

Fig. 4. Visual feedbacks: (a) translation - clusters not aligned, only one hand active (the one with green cross); (b) rotation - two aligned hands marked by two different squares, a blue one for the left hand (Y-axis rotation) and a yellow one for the right hand (X-axis rotation)

B. Experimental Results

The system has been informally tested several times during its development, to assess its quality and tune program parameters for best performance. However, at the end of the implementation phase we have also carried out some more formal experiments, aimed at validating the robustness of the interaction mechanism. 15 users were involved in these trials.

The two interaction modes were examined separately with two sub-tests:

1) Translation – The user raised one hand and randomly moved it for a while (about 10 seconds), to move and zoom the 3D object in the virtual space. The outcome of the trial was considered positive if the object was correctly "fastened" to the hand, and therefore was never lost while being shifted around the screen;
2) Rotation – The user raised both the hands, to trigger the second interaction mode. The tester had to spin the left and right hands for a while (about 20 seconds) to rotate the object around the X and Y axes. In this case too, the outcome of the trial was considered positive if the object was never lost during its virtual manipulation, as well as correctly revolved around its axes.

All the 15 testers succeeded in the first sub-test, while 14 out of 15 succeeded in the second. Considering that the reason for such a fail was mainly due to the incorrect execution of the gestures by the tester rather than to system inaccuracy, the results can be regarded as fairly good.

The tests have been performed on a computer equipped with an Intel Core 2 Quad Q9300 2.60 GHZ processor and a Nvidia GeForce GTX 260 graphic card. The proposed approach ensures a good compromise between computational time and precision of the results. The system easily reaches the 18 fps required by the ToF camera. Offline tests show that the system can go up to 32 fps, so it can be used also with more recent models of ToF camera (e.g. SR4000), that guarantee better performances and a higher frame rate, without further optimizations.

VI. Conclusion

This paper presents a new gestural interaction technique that allows the user to control the movement of a virtual object in the 3D space. The proposed approach exploits the potential of the data fusion of an RGB and a ToF camera, to obtain a flexible system, totally independent of the background, robust to false positive clusters detection, and with no need of any learning phase to recognize hands and gestures. Future improvements will include more precise sub-segmentation, in order to totally exclude parts of the arms during hand detection (currently avoided only considering distance constraints) and the addition of other gestures to obtain more complex interactions.

References

[1] T. Oggier, M. Lehmann, K. R., M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc, "An all-solid-state optical range

camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger)," in *Proceeding of SPIE Vol. 5249*, 2003, pp. 634–645.

[2] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-Flight Cameras in Computer Graphics," *Computer Graphics Forum*, vol. 29, no. 1, pp. 141–159, 2010.

[3] P. Dondi, L. Lombardi, and M. Porta, "Human-Computer Interaction through Time-of-Flight and RGB cameras," in *Proceedings of ICIAP 2011, 16th International Conference on Image Analysis and Processing*, vol. 2. Springer, September 2011, pp. 89–98.

[4] R. Reulke, "Combination of distance data with high resolution images," in *Proceedings of IEVM06, Image Engeeniring and Vision Metrology*, 2006.

[5] S. Ghobadi, O. Loepprich, K. Hartmann, and O. Loffeld, "Hand segmentation using 2D/3D images," in *Proceedings of Image and Vision Computing 07*, December 2007, pp. 64–69.

[6] S. E. Ghobadi, O. E. Loepprich, F. Ahmadov, J. Bernshausen, K. Hartmann, and O. Loffeld, "Real time hand based robot control using 2D/3D images," in *Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II*, ser. ISVC '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 307–316.

[7] P. Breuer, C. Eckes, and S. Mller, "Hand gesture recognition with a novel IR time-of-flight range camera: a pilot study," in *Proceedings of 3rd International Conference on Computer vision/computer graphics collaboration techniques (MIRAGE'07)*, 2007, pp. 247–260.

[8] Z. Li and R. Jarvis, "Visual interpretation of natural pointing gestures in 3d space for human-robot interaction," in *Proceedings of Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on*, December 2010, pp. 2513–2518.

[9] A. Treskunov, S. Kim, and S. Marti, "Range camera for simple behind display interaction," in *Proceedings of MVA2011 IAPR Conference on Machine Vision Applications, Nara, Japan*, June 2011, pp. 160–163.

[10] E. Kollorz, J. Penne, J. Hornegger, and A. Barke, "Gesture recognition with a time of flight camera," *Int. J. Intell. Syst. Technol. Appl.*, vol. 5, pp. 334–343, November 2008.

[11] M. B. Holte, T. B. Moeslund, and P. Fihl, "View invariant gesture recognition using the csem swissranger sr-2 camera," *Int. J. Intell. Syst. Technol. Appl.*, vol. 5, pp. 295–303, November 2008.

[12] M. Van den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, January 2011, pp. 66–72.

[13] M. Haker, M. Bhme, T. Martinetz, and E. Barth, "Deictic gestures with a time-of-flight camera," in *Proceedings of Gesture in Embodied Communication and Human-Computer Interaction 8th International Gesture Workshop, GW 2009*, S. Kopp and I. Wachsmuth, Eds., January 2009, pp. 110–121.

[14] T. Oggier, B. Bttgen, F. Lustenberger, G. Becker, B. Regg, and A. Hodac, "Swissranger SR3000 and first experiences based on miniaturized 3D-TOF cameras," in *Proceedings, 1st Range Imaging Research Day*. Springer, September 2005, pp. 97–108.

[15] N. Haubner, U. Schwanecke, R. Drner, S. Lehmann, and J. Luderschmidt, "Recognition of Dynamic Hand Gestures with Time-of-Flight Cameras," in *Proceedings of ITG/GI Workshop on Self-Integrating Systems for Better Living Environments 2010 (Sensyble Workshop)*, 2010, pp. 33–39.

[16] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber, "3-D gesture-based scene navigation in medical imaging applications using time-of-flight cameras," in *Proceedings of Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, June 2008, pp. 1–6.

[17] J. Penne, S. Soutschek, L. Fedorowicz, and J. Hornegger, "Robust real-time 3D time-of-flight based gesture navigation," in *Proceedings of Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, September 2008, pp. 1–2.

[18] P. Dondi and L. Lombardi, "Fast real-time segmentation and tracking of multiple subjects by time-of-flight camera," in *Proceedings of VISAPP 2011, 6th International Conference on Computer Vision Theory and Applications*, March 2011, pp. 582–587.