A Virtual Keyboard Based on True-3D Optical Ranging

Huan Du[1], Thierry Oggier[2], Felix Lustenberger[2], Edoardo Charbon[1]
[1]Ecole Polytechnique Fédérale Lausanne, 1015 Lausanne, SWITZERLAND
huan.du | edoardo.charbon@epfl.ch
[2]CSEM SA, Badenerstrasse 569, 8048 Zurich, SWITZERLAND
thierry.oggier | felix.lustenberger@csem.ch

Abstract

In this paper, a complete system is presented which mimics a QWERTY keyboard on an arbitrary surface. The system consists of a pattern projector and a true-3D range camera for detecting the typing events. We exploit depth information acquired with the 3D range camera and detect the hand region using a pre-computed reference frame. The fingertips are found by analyzing the hands' contour and fitting the depth curve with different feature models. To detect a keystroke, we analyze the feature of the depth curve and map it back to a global coordinate system to find which key was pressed. These steps are fully automated and do not require human intervention. The system can be used in any application requiring zero form factor and minimized or no contact with a medium, as in a large number of cases in human-to-computer interaction, virtual reality, game control, 3D designs, etc.
Keywords: virtual keyboard, computer vision, range camera, finger tracking, feature extraction, Swissranger, time-of-flight imaging

1 Introduction

As the demand for computing environments evolves, new human-computer interfaces have been implemented to provide multiform interactions between users and machines. Nonetheless, the basis for most human-to-computer interactions remains the binomial keyboard/mouse. Ordinary keyboards however, to be comfortable and effective, must be reasonably sized. Thus they are cumbersome to carry and often require wiring. To overcome these problems, a smaller and more mobile touch-typing device [1] has been proposed which does not have physical support. This device is known as virtual keyboard [2] or zero-form-factor interface.

Finger tracking and finger tracking based interfaces have been an actively researched problem for several years now. For example, glove-based systems, such as the "Key-glove" by Won, Lee et al. [3], require the user to wear a tethered glove to recognize signal variations caused by the movement of fingers. Recently, other kinds of sensors have also been used in wearable virtual keyboards. Senseboard for example has

developed a virtual keyboard system [4] based on two devices made of a combination of rubber and plastic. The devices recognize typing events by analyzing the data from pressure sensors attached to the user's palm. Other systems based on more sophisticated sensing devices have been proposed. One such example is the SCURRY system [5] which uses an array of gyroscopes attached to user's hands to detect the movements of fingers and wrist.

Computer vision researchers as well have made significant advances in the development of vision based devices that require no wearable hardware. Canesta and VKB for example have designed virtual keyboard systems [6][7] using infrared cameras to detect the interaction between the fingers and a projected image from a laser diode. Stereo vision has also been employed to obtain finger positioning and to perform rudimentary finger tracking. Lee and Woo for example describe ARKB [8], a 3D vision system based on a stereo camera.

This paper describes a system based on the Swissranger SR-2 camera demonstrator [9], a novel 3D optical range camera that utilizes both the gray-scale and depth information of the scene to reconstruct typing events. The tracking of the fingers is not based on skin color detection and thus needs no training to adapt to the user. After initial calibration for environment setup, the system is fully automated and does not require human intervention. Besides its application in a planar setup, the system can be employed in a real 3D space for virtual reality and gaming applications. Since physical contact is not a requirement of the system, important applications designed for disabled computer users and for users operating in hostile and sterile environments are envisioned.

The paper is organized as follows. In Section 2, the system architecture is described, including all main hardware modules and software components of the system. Section 3 outlines the camera-calibration process, including the models for all known sources of error caused by the camera. The dynamic event detection algorithm is discussed in Section 4, where we describe the techniques for fingertip and keystroke detection. Results are presented in Section 5.

2 System Architecture

Figure 1 shows the physical setup of the system. The 3D range camera is placed several centimeters over the input surface, with a well-defined angle facing the working area. The size of the working area, limited by the spatial resolution of the camera, is 15 cm × 25 cm, which is comparable to a full-size laptop-computer keyboard. The display projector is mounted on the camera, facing the same area, which would generate the visual feedback for the keyboard and input information.

The proposed system consists of three main hardware modules: (1) 3D optical range camera, (2) visual feedback, and (3) processing platform. The range camera is connected to the processing platform, presently a personal computer (PC), via a USB2.0 interface. The visual feedback module communicates with the computer via serial port.

The Swissranger SR-2 3D optical range camera simultaneously measures gray-scale and depth map of a scene. The sensor used in the camera is based on CMOS/CCD technology and it is equipped with an optical band-pass filter to avoid all background light. It delivers gray-scale and depth measurements based on the time-of-flight (TOF) measurement principle with a spatial resolution of 160 × 124 pixels. The light source of the camera is an array of LEDs modulated at 20 MHz with a total optical power output

of approx. 800 mW, however only a fraction of this total light power is utilized in the current setup. The depth resolution under these conditions is only about 1.5 cm without any spatial filtering, and may reach 0.6 cm with the spatial filtering with a window size of 5×5 pixels. A detailed discussion of the theoretical background and practical implementation of the camera can be found, e.g., in [10].

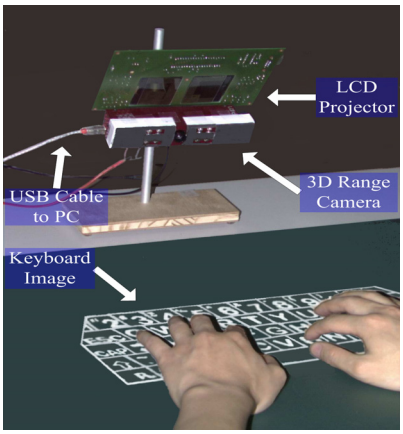


Figure 1. Projected-keyboard demonstration-system setup

The depth information supplied by the range camera allows developing simpler and more efficient computer-vision algorithms to estimate the position of fingertips and to locate the corresponding stricken key. Simplicity and efficiency are key elements to enable real-time or even portable applications.

However, there are still some challenges associated with the range camera utilized in this project. A number of problems, such as e.g., light scattering, and "close target" artifacts impact achievable depth accuracy. Moreover, the image resolution of the camera is relatively low, thus restricting its use to applications with large view window. As a result the working area of the keyboard is limited today to a sub-optimal size. The current power consumption and size of the range camera are also impediments to its use in truly portable applications. Naturally, the current demo system is composed, in part, by prototypes. Nonetheless, based on our research experience in the field of full-field TOF-based rangefinders, we believe that all these problems will be solved in the near future, and all-solid-state 3D cameras will soon fall in adequate size and price ranges.

The visual feedback module is constructed using projection of a dynamically generated image based on a mini LCD. Whenever the processing algorithm detects a key-striking or key-bouncing event, it sends an UPDATE command to the visual feedback module with specific key information. The feedback module updates the generated display according to the command and thus the user can see the change of the keyboard image as well as textual or graphical updates. Additional audio feedback is used to help the user identify successful keystrokes.

The processing algorithm consists of five main modules as shown in Figure 2: (1) depth map error correction, a camera dependent module based on specific models designed for the range camera, (2) background subtraction, (3) central column estimation, (4) fingertip detection, and (5) keystroke detection. Note that software modules (2) to (5) are camera independent modules applying computer vision algorithms to track the movement of fingers and to detect the typing event.

The 3D range camera is calibrated at startup. The projection matrix of the camera is estimated during calibration. The depth map delivered by the range camera is first processed by the error-correction routine to compensate for errors caused by parallax and distributed clock skews in the camera. The rectified range measurements, combined with gray-scale image, are then subtracted from the reference image and binarized to extract the hand region. After applying the *central column* estimation, which is defined as the pixel segments associated with fingers that are good candidates for an event, by searching the local extrema in x-coordinate along the hand boundary, and applying the fingertip detection by extracting features with curve modeling, precise location of fingertips can be found in the hand region. Finally, the keystroke detection is obtained by fitting depth curve applying another feature model, and the corresponding hitting positions are mapped back to the world coordinate system to infer the stricken keys. The updated key status is then sent to visual feedback module to generate refreshed display.

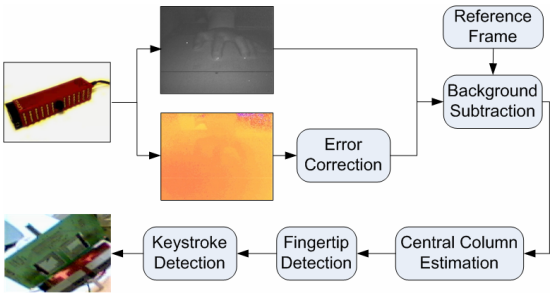
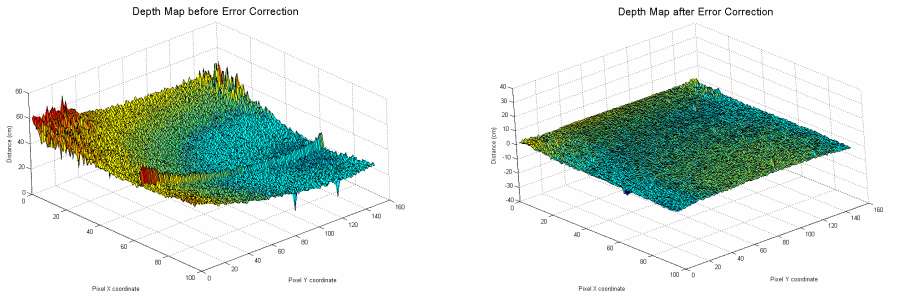


Figure 2. Software flowchart

3 Camera Calibration

The 3D optical range camera requires certain calibration procedures. To estimate the projection matrix M_c of the camera, we use the classic method proposed by Tsai [11] with respect to a world coordinate frame attached to the table. We also find that there exist some camera specific errors in the range measurement, and we analyze the cause of these errors and how to model and correct them.



(a) (b)

Figure 3. (a) Depth map of the desk before error correction and (b) depth map of the desk after error correction

There are mainly two types of errors in the range measurement. One is caused by the parallax effect, the other by skew in the clock distribution of the sensor. Both errors result in location dependent under- or overestimation of depth. The parallax error is estimated using triangular geometry of the scene projection. The clock-skew error linearly increases with the x- and y-coordinates in the image plane. The gain or slope in x- and y-direction may be estimated using LMS. Figure 3 shows the depth map of a flat table surface before and after error correction, and Table 1 lists the statistical value of the two cases. The mean value of the latter case has counted in the offset.

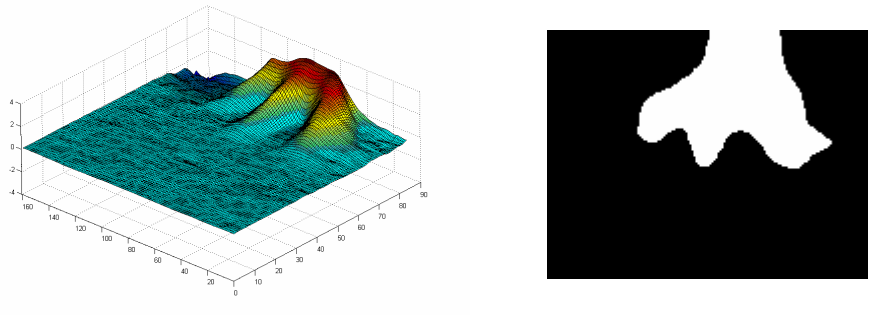
	Mean (cm)	Standard deviation (cm)
Before correction	38.42	12.73
After correction	35.84	1.46

Table 1. Statistics of depth map before and after error correction

4 Dynamic Event Detection

To detect the movement of fingers, the proposed system applies a series of computer-vision algorithms to the gray-scale image and depth map. The first step in the process is the segmentation of the scene into its foreground and background components. The foreground is defined as the fingers and forearms. Once the segmentation has been performed, the process of detecting the fingertips starts with estimating and extracting features.

For the detection of the hand region, the algorithm uses a previously acquired background image as its reference frame. The background is modeled using both, gray-scale image and depth map. During detection process, the input frame is compared to the reference frame, and its differential distance map is computed.



(a) (b)

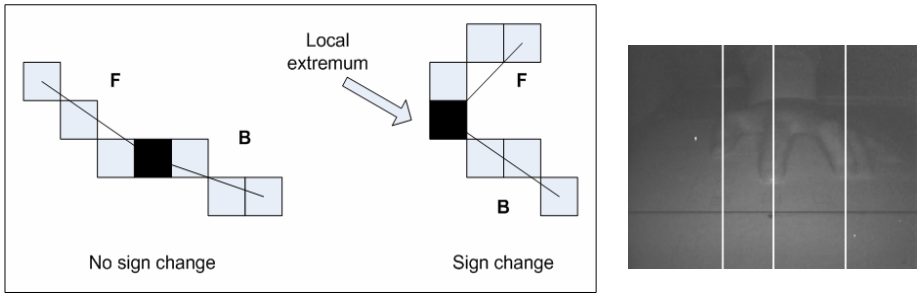
Figure 4. (a) Differential depth map and (b) Binarized hand region after background subtraction

When the pixels of the resulting differential map that have values larger than a given threshold (one for the depth map and one for the gray-scale image), it corresponds to the foreground. Binarizing the differential map results in a separation of the hand region from the background scene. Median filtering is then applied to the binary frame to reduce the impact of noise. The resulting hand image is clear and contains only few

gaps. Figure 4(a) shows the differential depth map, and Figure 4(b) shows the segmented hand region computed for an example image.

According to our analysis of the differential map after background subtraction, the difference is very small near the edge of the fingers and it is submerged by Gaussian noise, either in the gray-scale field or in the depth field. Therefore, it would be difficult to detect the precise location of a fingertip based solely on the thresholded hand region. Alternatively, the central-pixel column associated with a finger is first estimated from the binarized hand region. Then, the fingertip is detected by extracting features from the central column curve of the depth map. Since the frame rate of the demo system of 33 frames per second is not high enough to apply cross-frame tracking in such a close distance, since the target tracking requires additionally a relatively complicated hand model, which is not suitable for our real time algorithm, no temporal coherence is assumed for this system.

The most likely position of the central columns can be derived from the finger trunks based on the segmented hand region. We first compute the boundary of the hand by noting that any pixel on the hand region, which has 4-connectivity with a non-hand region, belongs to the boundary. Then we compute an approximation of *k*-curvature for each pixel on the boundary. *K*-curvature is defined by 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 list of contour points. Segen and Kumar used local curvature extrema [12], but with our specifically configured angle and elevation of the camera, we need only to compute local extrema in x-coordinate instead of the angle.



(a) (b)

Figure 5. (a) Example of local-extrema detection and (b) Result of central column estimation

With an appropriate value of *k*, the local extrema in the x-coordinates can be used to find the central columns of fingers. For each pixel on the boundary of the hand region, we extend two vectors *k* pixels away along the contour. Since it is an approximation of *k*-curvature and the computation for local extrema is simplified as sign detection, we may get a series of continuous "local extrema" near the actual central columns. In such a situation the center pixel of this series is taken as the estimation of the central column. Figure 5(a) shows an example of local extrema in x-coordinate. The figure indicates how sign change is used to detect local extrema. The current pixel is marked with black and the forward and backward vector from this pixel are denoted as *F* and *B*. If *F* and *B* have a different sign in x-direction, the current pixel is marked as local extrema, otherwise it is skipped. Figure 5(b) shows the result of central column detection in gray-scale image. The detected central columns are marked with white lines.

To detect the fingertip in the segmented hand image, the feature model of the differential depth curve is applied. The differential depth curve represents the depth values of the differential map along the central columns of the fingers in x-direction. It appears to be the piecewise linear superposition of a linear segment with a gradient close to zero, in correspondence of the finger, and a curve segment which can be modeled as a non-decreasing second order polynomial**.** We take the intersection point of two sections of the curve as the fingertip.

Figure 6 shows the differential depth curve of the central column and the fitted curves of its piecewise parts, respectively.

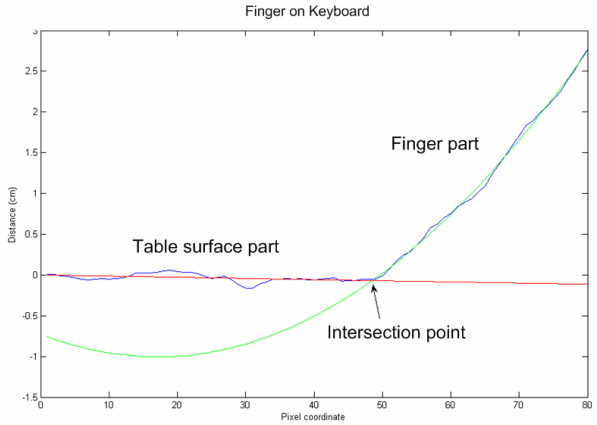


Figure 6. Differential depth curve and the fitted curve

The contact between fingertips and virtual keys is also detected by feature models. The depth curve of the fingertip appears to be a smooth second-degree parabola for the case that the finger touches the working surface or it is very close to it. For the case in which the finger lifts away from the working surface, the depth curve appears to have a discontinuity in the parabolic curve. The keystroke hypothesis is tested by curve fitting with second order polynomial. The hypothesis is verified when the fitted curve exhibits a deviation larger than some predefined threshold. This large deviation is caused by the perspective projection of the scene and the discontinuity in the depth map between the finger and the table.

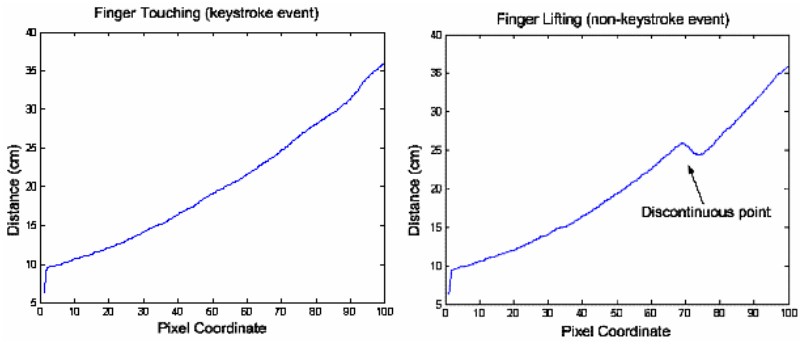


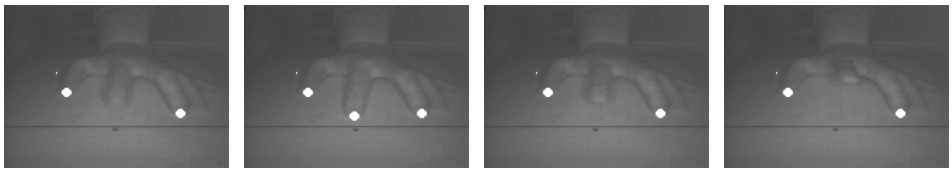
Figure 7. Depth curve of collision case and non-collision case

Figure 7 shows the depth curve in the finger-touching and the finger-lifting cases, respectively. The position of the typing fingertip is then mapped back to the world coordinate with previously calibrated projection matrix, and the corresponding virtual key is inferred from the differential depth map which codes the x- and z-coordinates of the keystroke.

5 Experimental Results

In this section, preliminary experimental results of the proposed system are presented. The range camera is connected to a PC with Pentium4 1.8GHz CPU. With these resources, the system could capture images and process them in approx. 30ms intervals, i.e., 33 frames per second on average. This frame rate is high enough for normal typing, which requires a finger speed of at most 10 cm/s. The camera was pre-calibrated with respect to a world coordinate frame associated with the surface, and the construction of the reference frame was achieved in less than 3 seconds.

Figure 8 shows a true human hand's typing motion being tracked by our algorithm and the detected keystroke event in the image sequence, which are marked with white dots. A sequence of frames leading to a specific keystroke is shown. The keystroke of the dynamic fingertip is detected accurately in Figure 8(b). Note that also the other fingers are correctly detected, as they are stationary keystrokes. Stationary keystrokes can be interpreted as REPEAT. Current challenge for the system is the finger occlusion problem for two-handed typing, which is common to most of the vision based hand tracking systems. A solution to this problem is to apply more complicated 3D hand models so that the position of the occluded finger can potentially be estimated. However, this may dramatically lower the system frame rate and cannot fulfill our real-time requirement.



(a) (b) (c) (d)

Figure 8. (a), (b), (c) and (d) Result from image sequence

The proposed system could also be extended to the application of a virtual mouse. The finger tracking method can precisely locate the position of a moving finger in the working area, and detect the click event in the same way as the detection of keystroke event. The only challenge is to track multiple fingers and record their traces in the scene for this application. In the trivial case we can assume that there is only one finger in the scene as the input source. However, for the case where the left and the right button are both simulated, or for some gesture controlling interface, temporal coherence information can be employed and a more complicated tracking algorithm should be devised.

Table 2 lists the results of the system's usability tests, which involve users of different races, typing skills and genders, reflecting the natural distribution of left- and right-handed subjects. The subjects were required to type test patterns indoors under

different lighting conditions and with slow, normal and fast typing speed. The finely designed test patterns cover all key positions and most of their possible combinations. Statistics were compiled to evaluate the false detection, misses and incorrect detection rate. False detection occurs when a key is not pressed but a character is issued. A miss occurs when a key is pressed but no character is issued. Incorrect detection is equivalent to a misprint. User set 1 consists of users who never used this system beforehand, and user set 2 of users who practiced with the system for less than 10 minutes. From Table 2 we found that the dominant detection error is the false detection in both cases. This error is mainly caused by the low lateral resolution of the camera, which could not resolve very small floating distance between the fingers and the table, though that users could not adapt to a flat typing surface is also a reason. Experiments showed that with short time of practice, accuracy and typing speed are both observably enhanced. It was also shown that an appreciably experienced user can reach the normal typing speed of approx. 30 words per minute with the presented system.

	False detection rate	Missed-strokes rate	Incorrect detection rate	Typing accuracy	Average typing speed (words/min)
User set 1	8.5%	3.5%	1%	87%	21.6
User set 2	6.7%	2.7%	1%	89.7%	30.8
Total	7.4%	3%	1%	88.6%	27.1

Table 2. Summary of results of the system-usability tests

In general, the surrounding lighting condition would impact the scene segmentation method if based on static reference frame. However, the infrared LED light source and the camera filter is narrow-banded and centered on a wavelength that only few natural light sources would have as their main component. This enables to extract the hand region according to pre-computed reference frame with very low processing complexity.

In this paper, only static background conditions were discussed. One major problem of having a mobile input device is that of dynamic scene. We are currently working on methods of automatically updating the reference frame if the movement of background is detected. Furthermore, for the comfort of user's wrist and elbow, generally virtual keyboard systems are developed based on 2D flat keyboard area. However, with the depth information supplied by the range camera, our system could be extended to 3D keyboard without any actual typing surface, which is also the direction for our future study.

6 Conclusions

A virtual keyboard system based on a true-3D optical range camera is presented. Keystroke events are accurately tracked independently on the user. No training is required by the system that automatically adapts itself to the background conditions when turned on. No specific hardware must be worn and in principle no dedicated goggles are necessary to view the keyboard since it is projected onto an arbitrary surface by optical means. The feedback text and/or graphics may be integrated with such

projector, thus enabling truly virtual working area. Experiments have shown the suitability of the approach which achieves high accuracy and speed.

Acknowledgements

The project was supported by a grant of the Swiss National Science Foundation – Grant Nr.: 620-066110. The authors wish to thank Nicholas Blanc as well as the entire crew from the Image Sensing group of CSEM SA for their technical support with the Swissranger SR-2 range camera.

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] Elliot, C., Schechter, G., Yeung, R., and Abi-Ezzi, S. *TBAG: A High Level Framework for Interactive, Animated 3D Graphics Applications*, In Proceedings of ACM SIGGRAPH 94, ACM Press / ACM SIGGRAPH, New York, pp. 421-434, 1994.

[3] Won, D. H., Lee, H. G., Kim, J. Y., and Park, J. H. *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, 2001.

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

[5] Spring, T. 2001. *Virtual Keyboards Let You Type in Air*, PCWorld article, http://www.pcworld.com/news/article/0,aid,0568,tk,dnWknd 1117,00.asp

[6] Canesta Inc., http://www.canesta.com

[7] Vkb Inc., http://vkb.co.il

[8] Lee, M. And Woo, W. *ARKB: 3D vision-based Augmented Reality Keyboard*, International Conferece on Artificial Reality and Telexisitence, pp. 54-57, 2003.

[9] CSEM SA, http://www.swissranger.ch

[10] Oggier, T., Lehmann, M., Kaufmann, R., Schweizer, M., Richter, M., Metzler, P., Lang, G., Lustenberger, F., and Blanc, N. *An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger)*, In Proceedings of SPIE 2003, pp. 534-545, 2003.

[11] Tsai, R. Y. *A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*, IEEE Journal of Robotics and Automation, Vol. 3, pp. 323-344, 1987.

[12] Segen, J. and Kumar, S. *Human-Computer Interaction using Gesture Recognition and 3D Hand Tracking*, In Proceedings of ICIP, Chicago, pp. 188-192, 1998.