Markerless 3D Interaction in an Unconstrained Handheld Mixed Reality Setup

Daniel Fritz, Annette Mossel and Hannes Kaufmann

Vienna University of Technology, Favoritenstr, Wien, Austria

***Abstract*— In mobile applications, it is crucial to provide intuitive means for 2D and 3D interaction. A large number of techniques exist to support a natural user interface (NUI) by detecting the user's hand posture in RGB+D (depth) data. Depending on the given interaction scenario and its environmental properties, each technique has its advantages and disadvantages regarding accuracy and the robustness of posture detection. While the interaction environment in a desktop setup can be constrained to meet certain requirements, a handheld scenario has to deal with varying environmental conditions. To evaluate the performance of techniques on a mobile device, a powerful software framework was developed that is capable of processing and fusing RGB and depth data directly on a handheld device. Using this framework, five existing hand posture recognition techniques were integrated and systematically evaluated by comparing their accuracy under varying illumination and background. Overall results reveal best recognition rate of posture detection for combined RGB+D data at the expense of update rate. To support users in choosing the appropriate technique for their specific mobile interaction task, we derived guidelines based on our study. In the last step, an experimental study was conducted using the detected hand postures to perform the canonical 3D interaction tasks selection and positioning in a mixed reality handheld setup.**

***Categories and Subject Descriptors*—H.5.2 [Information Interfaces and Presentation]: User Interfaces| Interaction styles, I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism | Virtual Reality**

***General Terms*— Algorithms, Performance, Reliability**

***Index Terms*— Natural User Interface, RGB+D Hand Posture Detection, Handheld Augmented Reality, Comparative Study**

I. INTRODUCTION

Over the last years, there was a tremendous progress in the field of natural user interaction (NUI) as a new paradigm of human computer interaction (HCI). From multi-touch displays to finger and gesture recognition using RGB images as well as depth data, a large number of 2D and 3D interaction methods have been developed to provide intuitive interfaces. While multi-touch approaches require complex gestures to enable 3D interaction, detecting the user's hand posture provides a more straightforward

Figure 1: RGB+D NUI for mobile AR/VR

and thus natural 2D/3D user interface. Such an interface allows users to freely move their hand when interacting with a stationary or mobile device. Natural 3D interaction is especially of interest in mixed reality (MR) environments. While in desktop MR scenarios the interaction environment can be tailored to meet certain requirements of the applied method - such as constant illumination and non-cluttered background - to ensure robust hand recognition, interaction with a handheld device might be performed elsewhere. Thus, a mobile NUI must be able to cope with less constrained settings. Previous work demonstrates that depth data, processed on a stationary device, can be combined with RGB data to enable more reliable hand segmentation in cluttered or poorly illuminated environments. However, to provide a truly mobile NUI, RGB and depth data must be processed and fused directly on the handheld device. Based on RGB+D data, a developer can choose from a large number of existing hand detection approaches to build a powerful mobile NUI. However, a study comparing and elaborating the properties of each technique to provide guidelines for mobile NUIs is missing yet.

This paper presents the following three main contributions:

- A systematic study to evaluate existing RGB and RGB+D approaches for markerless hand posture recognition in a mobile interaction task.
- Guidelines to help developers choose the most suitable markerless posture recognition technique for a certain application scenario.
- A powerful software framework to process and fuse RGB and depth data on a mobile device. Thereby, we aim for a markerless NUI for handheld VR/AR to provide intuitive and robust hand posture recognition (Figure 1).

II. RELATED WORK

Over the last years, 2D multi-touch has become the de-facto standard on handheld devices for interaction. However, due to the implicit characteristics of 2D touch input, only 2D gestures can be provided. Various multi-touch approaches have been designed to manipulate 3D objects but are not usable for natural and intuitive 3D interaction due to the missing 3D input. In order to enable natural 3D interaction, the third dimension needs to be taken into account and integrated to provide 3D gestures. On mobile devices, the built-in camera is usually used for vision-based methods (marker-based or markerless) to determine the hand and its 3D position for interaction. Marker-based approaches use color markers or gloves in order to perform hand segmentation. However, they cannot serve for natural interaction since pre-conditioning of the hand with additional equipment is required. Thus, we focus on markerless-based methods for hand segmentation and 3D interaction.

A great number of approaches for markerless hand detection based on RGB data use skin color as the main feature for segmentation [2, 5]. For these methods no special training is necessary; however, color as a feature is sensitive to variations in lighting condition. It is also impossible to distinguish hands from other skin-colored objects. Baldauf et al. [2] use the RGB camera of a smartphone and skin-specific values as threshold to segment skin-colored pixels. Morphological operations are applied to remove noisy pixels and fill holes in potential skin areas. After calculating the contours, the largest connected area is presumed to be the hand. Fingertips are determined by evaluating the distance between hand contour points and the inner and outer hand circle as well as using the hand's orientation. Other approaches use Haar-like features [6] for hand detection [3, 11]. Although classifiers based on Haar-like features are computationally expensive and require an offline training phase, they can differentiate hands from a face or other skin-colored objects. The work of Rodriguez et al. [11] uses a Haar-like feature classifier to detect a hand in a RGB webcam image. Then, two filters are applied to remove repetitions and false positives. The hand detection provides size and 2D position; the missing hand's depth value to allow for 3D interaction is estimated by evaluating the hand size (relative depth estimation).

Combining a RGB- and a depth-camera results in RGB+D data, which is used by a number of approaches [14, 15] to improve RGB-based hand segmentation and to provide absolute depth values for more precise 3D interaction. Van den Bergh and Van Gool [14] use a RGB- and a Time-of-Flight-camera (ToF) to generate RGB+D data. The low resolution of the depth camera is sufficient to detect foreground and remove background objects and to determine the hand's distance from the camera. The high resolution of the RGB camera allows for accurate and robust hand detection. Using a Haar-like feature-based classifier, the user's face is detected and the absolute distance of the face is determined using the ToF depth data. Based on this distance, a threshold is introduced to remove the background in the RGB data. Within the remaining RGB pixels, the hands are detected by applying skin color segmentation.

III. DETECTING HAND POSTURE & 3D POSITION

Based on the related work, we chose five different markerless approaches to detect hand postures in imaging data that have been proven to be reliable, fast and/or robust. Therefore, we combined and extended existing techniques to be executed on a handheld device, as described in Section III.1. We did not apply color techniques for posture detection in RGB data due to its limitations to distinguish between hands and similarly colored objects. Using the identified hand, we integrated three different techniques to either estimate the relative or the absolute hand 3D position, as described in Section III.2. Our selected approaches use either RGB data for hand segmentation, posture detection and relative depth estimation or RGB+D data for improved hand segmentation and absolute depth estimation. Figure 2 shows the general setup of the prototype and interaction space.

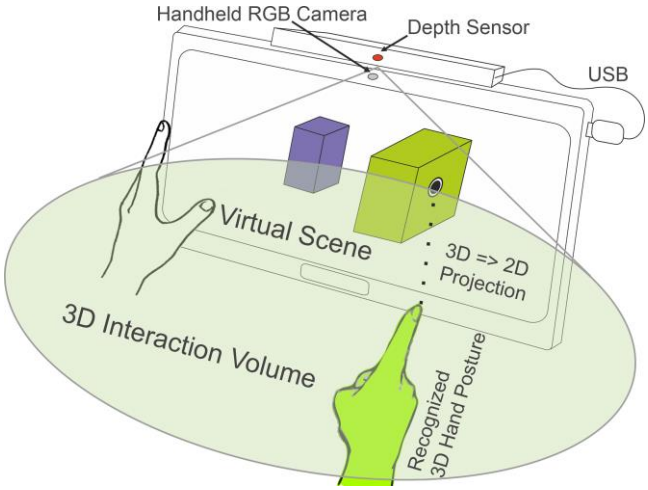

Figure 2: Mobile hardware setup and 3D hand interaction

The user can interact using two different postures: *Posture 1* (five fingers outstretched, see Figure 3a) controls the virtual hand to select a virtual object. With *Posture 2* (pointer finger outstretched, see Figure 3b), the user can position a selected object.

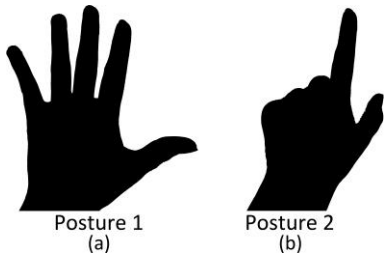

Figure 3: Postures for 3D interaction

III.1 Hand Posture Detection

The approaches selected for detecting the hand posture are divided into two types. The first approach solely employs cascaded classifiers on RGB data while the set of second approaches use RGB+D data as input and perform a number of image processing operations to determine the postures.

III.1.1 Pure Cascaded Classifier

The *First Approach* (**A1**) is based on RGB data and two Haar-like feature-based cascaded classifiers [6], *C1* and *C2*, to detect *Posture 1* and *Posture 2*.

The classifiers deliver the hand's 2D position and hand size, which is subsequently used for relative depth estimation to obtain the hand's 3D position for interaction. As an alternative, the 3D position is established with the maximum gray scale value of the hand, assuming the higher the gray value the closer the hand is to the camera. Both techniques are explained in Section III.2. The data flow at runtime is depicted in Figure 4.

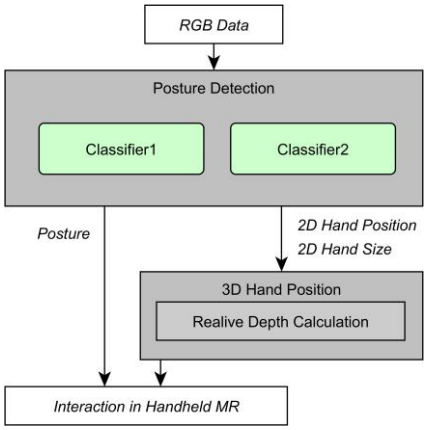

Figure 4: Application flow of A1 with relative depth estimation

III.1.2 Cascaded Classifier + Image Segmentation

The *Second Approach* (**A2**) uses a Haar-like feature-based cascaded classifier *C3* to detect the palm only. This classifier is trained with the outstretched thumb and the bottom half of the hand's palm as features and does not depend on the other fingers for detection. Within the detected hand region, different image processing operations are then applied to determine the hand's contour, resulting in the following sub-types of A2:

A2-C: Canny Edge Detection [8] (Figure 5a)

A2-H: Threshold with automatic adaption (Figure 5b), based on minimum/maximum values of hue, saturation and value (HSV) of the region at the center of the hand (Figure 5c), minus 10% (top and bottom) to remove outlier

A2-T: Fixed threshold with optional user adaption (Figure 5d)

A2-D: Using RGB+D data to remove fore- and background for robust hand segmentation (Figure 5e)

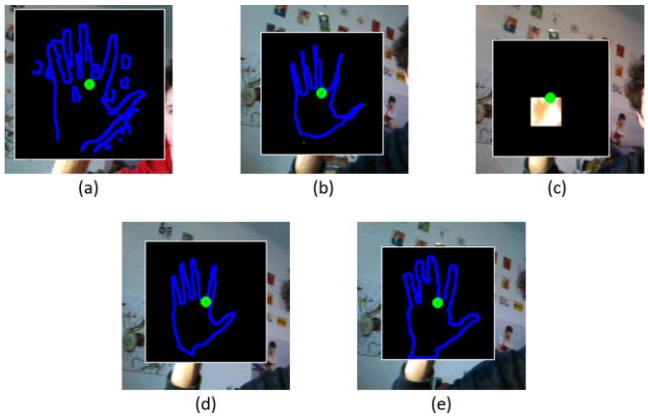

Figure 5: Hand contour with A2

In A2-C, A2-H and A2-T, the convex hull (Figure 6a) and convexity defects are computed for fingertip detection.

Convexity defects describe the deviations of the contour to the convex hull. The convexity defects are fingertip candidates and have to meet certain criteria to be classified as fingertips, i.e. a minimum distance to the hand's midpoint (Figure 6b). Depending on the amount of detected fingertips (Figure 6c), the hand's postures are determined as four to five fingertips define *Posture 1* and one to two fingertips define *Posture 2*.

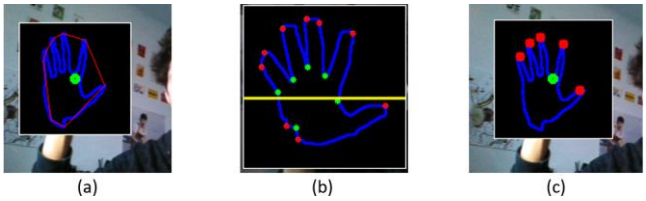

Figure 6: Fingertip detection

The hand's 3D position is calculated with relative depth estimation, as described in Section III.2. The data flow at runtime to determine the hand's 3D position using A2-C, A2-H and A2-T is shown in Figure 7.

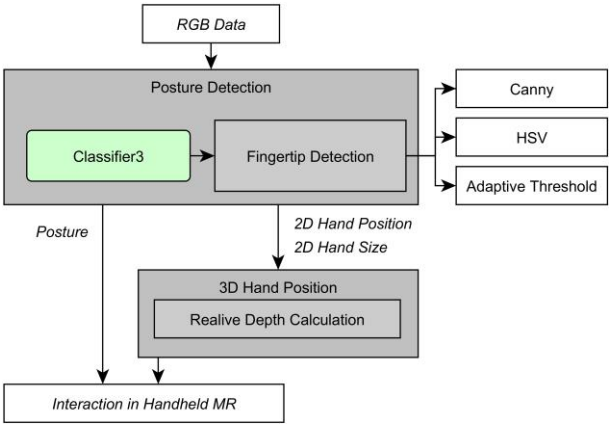

Figure 7: Application flow of A2 with relative depth estimation

In A2-D, the hand's palm is detected using the cascaded classifier C3. Next, RGB and depth imaging data are fused, as described in Section III.2.2, to obtain more robust fingertip candidates and to calculate the absolute depth position of the hand for 3D interaction. Only those RGB pixels are considered that lie onto the same depth plane as the detection window, obtained by classifier C3. Subsequently, all other pixels are removed from the RGB data, resulting in a pixel mask that only contains hand pixels. Therefore, the hand can easily be segmented and its contour used for fingertip detection, as described before. With this approach, we overcome the limitation of [15] that requires the hand to be the closest object to the camera for correct hand segmentation. The workflow to calculate the hand's 3D position for later interaction is depicted in Figure 8.

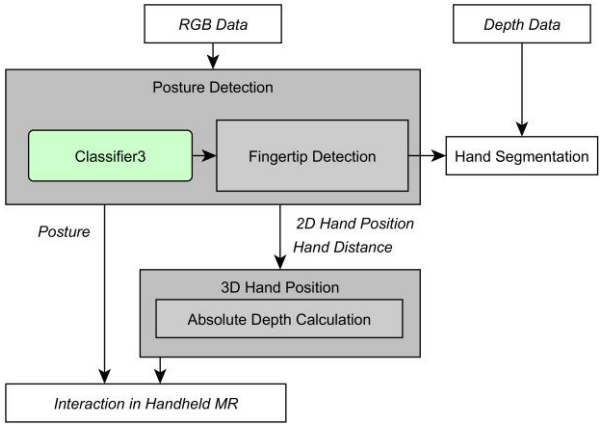

Figure 8: Application flow of A2 with absolute depth estimation

III.2 3D Position Estimation

The 2D position of the hand is obtained using the proposed approaches from III.1. It is provided by the classifier upon posture recognition by using the calculated center point of the detection window as x- and y-coordinates, as illustrated in Figure 5. To obtain the hand's z-coordinate to enable 3D interaction, relative and absolute depth estimation methods are applied, as described in the following.

III.2.1 Relative Depth Estimation

Additionally to the hand's 2D position, the classifier also delivers the **hand size** that is given by the detection window size. It can be subsequently used for relative depth estimation by exploiting the correlation between detected hand size and distance. The bigger the hand appears in the image, the closer it is to the camera in physical space. Before tracking, the minimal and maximal values for hand size and corresponding distance need to be empirically determined once by segmenting the hand at minimal and maximal distance to the camera. The minimal and maximal distance is defined by the boundaries from and up to the hand that can be detected by the algorithm within the RGB imaging data. Thereby, the relation between detected hand size and physical distance is established and subsequently used to estimate the z-coordinate of the hand during interaction.

Alternatively to the hand size, the maximum **gray value** of the detected hand region is calculated by analyzing the gray scale pixel values. Upon determination, this value can be employed for relative depth estimation. We made the assumption the higher the gray value the closer the hand is to the camera, resulting in the corresponding z-coordinate of the hand. At the beginning of the interaction, the minimum and maximum gray values are obtained by positioning the hand at minimum and maximum distance to the camera. Again, these values are the minimum and maximum boundaries of the interaction volume. As this approach highly depends on the illumination, the system must be able to cope with changes of the environmental

light situation during runtime. Therefore, the user can adjust the minimum and maximum gray values during tracking leading to a constant input for interaction and a reliable depth estimation.

III.2.2 Absolute Depth Estimation

The **depth data** provided by the depth imaging device is used to calculate the absolute depth of the hand's 3D position. Therefore, RGB and depth data need to be fused. This is performed by intrinsically and extrinsically calibrating both cameras in an offline process using [7, 8]. Next, the z-coordinate of the calculated hand's x/y-coordinate (in RGB camera coordinate system) can be determined by transforming the corresponding depth pixel into the RGB camera coordinate system by employing the obtained camera projection matrices.

IV. FRAMEWORK

In our developed framework, the handheld device and the depth sensor are rigidly coupled and calibrated to provide correctly registered RGB+D data. For intrinsic and extrinsic camera calibration, pictures of a 7×4 checkerboard pattern were simultaneously taken with the built-in mobile RGB camera and the externally connected depth imaging device. All data was then processed with the *MIP* tool [7] to estimate both internal and external camera parameters. Based on the calibration, depth data is mapped onto the RGB image, resulting in RGB+D data [4].

All computations – capturing and fusing of RGB and depth data, detection of hand and posture – are performed on an *Asus Eee Pad Transformer Prime (TF201)* tablet. It runs Android 4.1.1 and is equipped with a quad-core 1.4GHz processor, 1GB main memory, 10.1 inch display with 1280 x 800 pixel, USB 2.0 port and a 1.2-megapixel front-facing camera that provides the RGB data. A *Kinect for Windows* is used as depth sensor.

The proposed setup uses RGB+D data with a resolution of 320x240 pixels. Via USB, the Kinect is directly connected to the tablet [9] and is interfaced with the hand posture detection application using *OpenNI* [10]. The drivers with OpenNI support, provided by *Avin2's SensorKinect* [12], had to be re-compiled with *Android NDK* [1] in order to use the Kinect with Android. Unfortunately, this driver does not support the *Near Mode* feature of the Kinect for Windows. Thus, the proposed setup requires a minimum distance of 48cm for reliable and robust hand depth data.

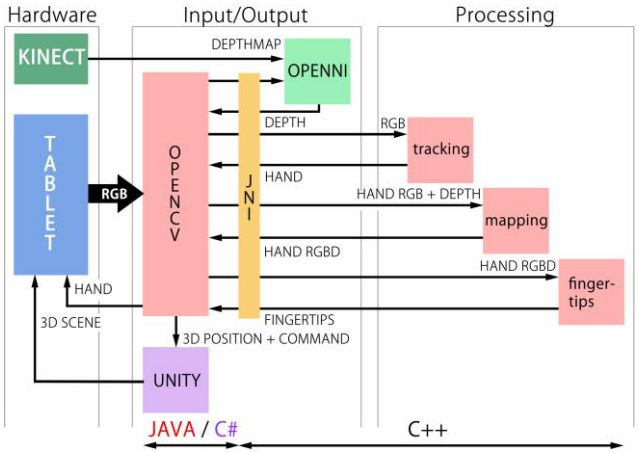

Figure 9: Software Processing Pipeline

In Figure 9, the software processing pipeline is depicted. We use *OpenCV* [8] for hand and fingertip detection in RGB data. To minimize tracking latency, Kinect integration as well as image and depth data processing is entirely implemented in C++, using *Android NDK* [1] and the *Java Native Interface (JNI)*. Rendering of the virtual scene as well as 3D interaction is handled by the game engine *Unity3D* [13], which is integrated into the Android application.

V. EVALUATION OF POSTURE DETECTION

With our framework, we tested the proposed RGB classifier and RGB(+D) segmentation techniques for posture recognition by evaluating five different test sets with varying hand positions, background and illumination conditions.

V.1 Test Sets & Ground Truth

We established five test sets to simulate various realistic VR/AR 3D interaction scenarios. Each test set consists of equally weighted (50:50 ratio) positive (with hand posture) and negative (without posture) images.

Set1: *Posture 1*; constant light, white background (104 images)
Set2: *Posture 1*; varying light, white background (110 images)
Set3: *Posture 1*; constant light, cluttered background (148 images)
Set4: *Posture 2*; varying light, white background (138 images)
Set5: *Posture 1&2*; varying light, white background (220 images)

All images were captured with our application using the front-facing camera (RGB) and Kinect (depth map). Next, we annotated all images (center point of the hand) to formulate a well-defined and robust ground truth.

V.2 Results

The hand posture detection was evaluated with *f-score*, as the combination of precision and recall as their weighted average, and *accuracy*, as the proportion of true results. For performance evaluation, we measured the achieved application frame rate.

Table 1: Performance evaluation

Mean (σ)	A1	A2-C	A2-H	A2-T	A2-D
Frame rate in fps	13.86 (1.91)	16.6 (1.76)	16.62 (1.88)	16.79 (1.79)	8.13 (1.51)

The mean frame rate of each approach with standard deviation σ is listed in Table 1. Since A1 as well as the A2-D require complex computations, their processing is slower, resulting in significantly less frame rates than A2-C, A2-H, A2-T. Before evaluating f-score and accuracy for each (compound) approach, we first tested separately the performance of the three Haar-like feature classifiers C1, C2, C3. Therefore, we calculated the mean of f-score and accuracy for each classifier over all test sets. As listed in Table 2, the results indicate similar performances of classifier C1 to detect *Posture 1* in A1 and classifier C3 to detect the hand's palm in A2. The results of classifier C2 to detect *Posture 2* in A1 were found less accurate than C1 and C3.

Table 2: Hand detection - Classifier evaluation

Mean (σ)	C1	C2	C3
F-Score in %	86.03 (12.12)	60.61 (5.79)	82.22 (9.89)
Accuracy in %	90.13 (7.06)	72.65 (13.22)	81.26 (8.89)

Next, all test sets were independently processed with each hand posture detection approach. Then, the obtained results were compared with the ground truth to determine true and false detections. We defined detection as true positive if the distance between the center point of the detected hand and the center point of the ground truth is less than 15 pixels.

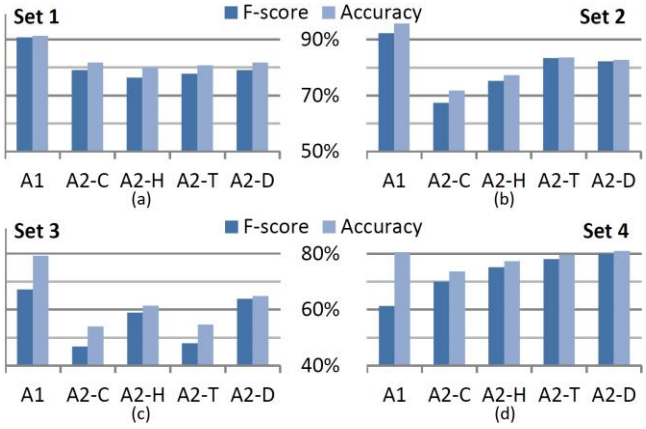

Figure 10: Evaluation of Set 1, Set 2, Set 3 and Set 4

As illustrated in Figure 10, the detection of *Posture 1* in front of white backgrounds shows good results for all approaches with constant illumination (Set1, Figure 10a) but poorer results for A2-C and A2-H for varying lighting conditions (Set2, Figure 10b). When evaluating *Posture 1* in front of cluttered background (Set3), considerable differences between A2-C and A2-T could be found, as illustrated in Figure 10c. Compared to Figure 10b, the results for *Posture 2* (Set4) exposed almost similar performance of A2-C, A2-H, A2-T, A2-D and a large deviation for A1, as depicted in Figure 10d.

After separately evaluating the detection of *Posture 1* and *2*, we tested their detection in the combined Set5. As depicted in Figure 11a, A1, A2-T and A2-D perform considerably better than A2-H and especially A2-C.

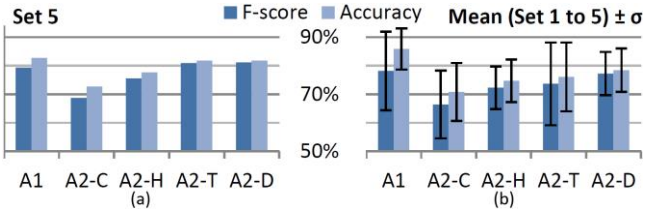

Figure 11: Evaluation of Set 5 and Mean of Set 1 to 5

Finally, we calculated the mean f-score and mean accuracy over all test sets for each approach. As illustrated in Figure 11b, the best results are achieved with A1 and A2-D, while A2-C performed worst.

V.3 Discussion

The results of the performance evaluation indicate that all approaches achieve interactive frame rates to provide intuitive 3D interaction. However, A1 and A2-D perform slower caused by the use of two computationally expensive classifiers in A1 and the gathering of depth data and mapping in A2-D. The Haar-like feature classifiers C1 and C3 delivered good result even under varying lighting conditions and with cluttered backgrounds. Results for classifier C2 have found lower f-score and accuracy. That might be caused by the shape of an outstretched single finger, which is small and therefore provides fewer features for detection and differentiation from the background. This problem can be mitigated with more intensive training of the classifier.

Analyzing the results for posture recognition, A1 was found to work well with cluttered background and showed overall good results for *Posture 1*. The differences for *Posture 2* reflect the results of the classifier evaluation of C2. A2-C performed poorly with cluttered backgrounds and was overall ranked last in this evaluation. Since the Canny Edge detection cannot differentiate between hand contour and other edges, this approach is not recommended in a minor to non-constrained handheld NUI environment. A2-H achieved good overall results and outperformed A2-C as well as A2-T, particularly with cluttered backgrounds. To be able to react to varying lighting conditions, user can manually adapt the threshold in A2-T.

For those situations, it shows good results but performs worse in situations with cluttered background. The depth data in A2-D is used to remove the background; this results in good performance with cluttered backgrounds, varying lighting conditions and the best overall performance amongst the A2 approaches. The standard deviations in Table 2 and Figure 11b are mostly influenced by cluttered backgrounds and C2, as discussed above.

Based on the given results and the test sets, we derived guidelines to support users to choose the most suitable hand recognition technique for a certain handheld VR/AR 3D interaction scenario:

- **A1** shows very good results if the underlying classifier is well trained, but provides slower frame rate and requires one classifier for each posture.
- **A2-H** is overall the best choice if only RGB data is available and fast tracking is required.
- **A2-T** provides high frame rates, can be quickly implemented and is suitable if 3D interaction is performed in front of a white background. For different illumination situations, manual user adaption is required to react to light changes.
- **A2-D** is the most robust approach for all 3D interaction situations, but provides smaller update rates.

VI. EVALUATION OF 3D INTERACTION

As hand posture recognition is the essential foundation to enable 3D interaction, we used the framework and the integrated interaction techniques to conduct an experimental evaluation with one participant of the two canonical 3D interaction tasks selection and positioning.

VI.1 Test Scenario

For task evaluation, a virtual scene was created comprising an empty cubic volume. A cube was added to the scene and set to its initial position in the front left corner. The rear right corner was defined as target area. In this virtual scene, the user had to perform the following compound interaction task:

Task 1: Select the cube using *Posture 1*
Task 2: Translate the cube into the target area using *Posture 2*

Using the recognition approaches A1 and A2, as described in Section III, the postures are detected and the estimated 3D position of the user's hand is mapped to a virtual interaction object, the so called virtual hand.

VI.2 Results

For each posture detection approach, the compound interaction task was performed once with constant illumination and in front of a cluttered background. To evaluate the reliability of detection, the obtained postures were recorded and plotted for the time the interaction occurred.

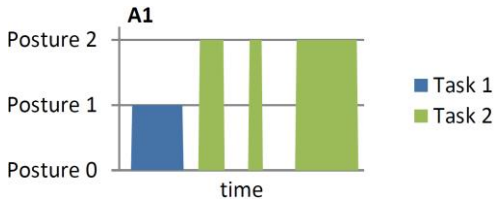

Figure 12: Posture detection with A1

Figure 12 depicts good results of A1 for Task 1 "Selection" that requires the detection of *Posture 1*. During Task 2 "Positioning", the detection of *Posture 2* resulted twice in false negative identifications.

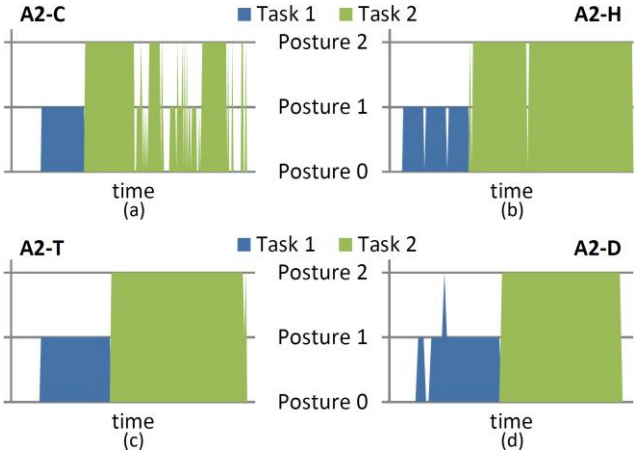

Figure 13: Posture detection with A2-C, A2-H, A2-T & A2-D

The evaluation of posture detections for all sub-types of A2 is depicted in Figure 13. As illustrated in Figure 13a, A2-C indicates stable detection results during Task 1 but a considerable amount of false negative identifications while performing Task 2. For A2-H, Figure 13b depicts short interruptions of posture detections during Task 1 and Task 2. Using A2-T, reliable detection during both interaction tasks is observed, as illustrated in Figure 13c. As shown in Figure 13d, A2-D detects *Posture 1* reliably, except two false detections, and indicates robust posture identification during Task 2.

After evaluating the robustness as well as reliability of posture detection over the entire compound interaction task, the estimated 3D positions of the virtual hand are examined. Therefore, the hand's 3D position was recorded during the interaction and subsequently plotted.

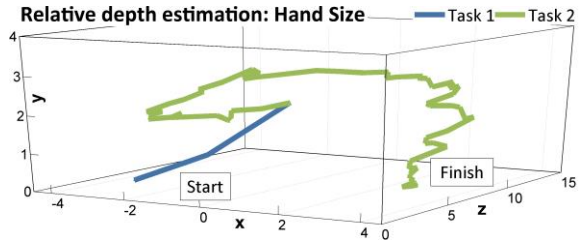

Figure 14: 3D position of Virtual Hand using Hand Size

In Figure 14, the virtual hand's 3D position was obtained using the hand size. The graph indicates a straightforward task completion that requires no manual initial step by the user; however, considerable deviations of the z-coordinate estimate are observed. They are caused by the varying size of the classifiers detection window.

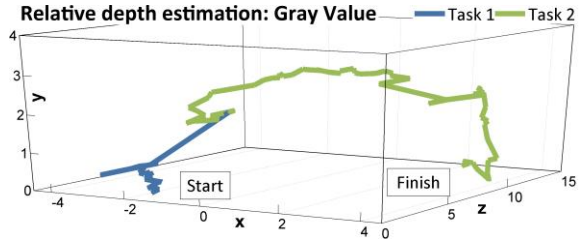

Figure 15: 3D position of Virtual Hand using Gray Value

Figure 15 shows a fair performance to accomplish the compound interaction task by analyzing the maximum gray value of the hand to estimate its z-coordinate. However, it requires a training phase at the beginning of the interaction indicated by the back and forth movements at the beginning of Task 1. The deviations in the z-coordinate during Task 2 are decreased compared to Figure 14. The remaining deviations are caused by varying hand illumination and thus varying absolute gray value calculation for similar distances due to changes of the hand's orientation in relation to the light.

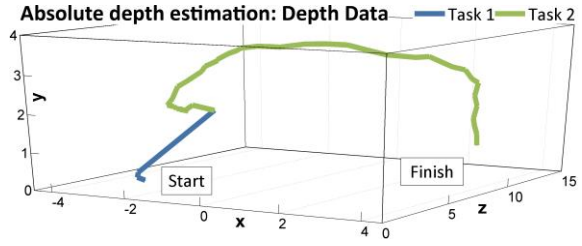

Figure 16: 3D position of Virtual Hand using Depth Data

In Figure 16, the estimated absolute 3D position of A2-D is shown. The graph depicts a continuous z-coordinate estimate, indicating a robust frame-to-frame 3D position over the entire task time.

VI.3 Discussion

Analyzing the results of posture detection over the time of 3D interaction, strong correlations were found to the results of the detection evaluation from Section V.2. C2 performs unstable to identify *Posture 2* for Approach A1, leading to false negative detections during the positioning task. Using the Canny edge detector for fingertip segmentation was again found to be most unstable, reflected by the large amount of false negative detections during the positioning tasks. The results of A2-T, A2-H and A2-D achieved stable overall results with just a minor amount of false negative detections during the compound interaction tasks. However, this was found to not significantly affect the interaction experience during the qualitative evaluation.

When evaluating the results of the 3D hand position determination, taking the hand size as a measure for depth estimation provides fair results. No manual initialization phase was required allowing the user to interact in 3D upon application start. However, the deviation of the z-coordinate is caused and influenced in magnitude by the performance of the classifiers. In the presence of false positive detections or incorrect estimations of the hand size, the z-coordinate estimate is jittery, decreasing the 3D interaction experience. The depth estimation using the hand's maximum gray value was found to perform well in this test scenario as well, requiring constant lighting and a well-lit hand. After a brief familiarization phase to the light-dependent relation between gray and depth value, the user was able to complete the 3D interaction task without problems. The range of accepted gray values has to be manually adapted by the user according to the scene's current illumination conditions. The most robust, stable and thus reliable 3D positions were estimated by processing the depth imaging data. No initial adaption phase was required and no deviations of the z-coordinate occur, leading to a natural 3D interaction experience.

The results of the posture recognition during the compound interaction task correlates with the results of the detection evaluation from Section V and emphasize the importance of a robust detection to provide natural interaction experience. Taking the results and findings of the 3D position estimation into account, the assets and drawbacks of the presented methods are:

- **Hand size** is a reliable feature for relative depth estimation but only if classifiers are well trained. It provides a straightforward connection between hand size and 3D position without the need of an initial setup phase.
- **Gray value** of the hand can act as a reliable input parameter for relative depth estimation, but only if constant illumination is given and the hand is fairly illuminated. In addition, the correlation between gray value and depth estimation requires an adaption phase and is not self-explanatory.
- **Depth data** provides an absolute distance estimate that can be seamlessly integrated to enable a natural 3D interaction experience.

VII. CONCLUSION & FUTURE WORK

We have presented a software framework to provide natural 3D interaction for handheld VR/AR environments using RGB and RGB+D data. Based on a systematic study, we were able to derive useful guidelines to support developers to choose an appropriate technique amongst the large variety of approaches. The experimental 3D interaction evaluation supported the results of the detection evaluation and demonstrated the different characteristic of the depth estimation methods. A comprehensive user study is planned to further examine 3D interaction in a handheld mixed reality setup. Future work will also focus on dynamic gesture recognition as well as integration of smaller depth sensing devices to provide a handheld NUI with enhanced usability.

VIII. REFERENCES

[1] Android NDK, [Software] Revision 8 http://developer.android.com/tools/sdk/ndk/index.html. Accessed: 2014-07-02.

[2] Baldauf, M., Zambanini, S., Fröhlich, P. and Reichl, P. 2011. Markerless visual fingertip detection for natural mobile device interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)* (2011), 539–544.

[3] Bilal, S., Akmeliawati, R., Salami, M.J. El, Shafie, A. a. and Bouhabba, E.M. 2010. A hybrid method using haar-like and skin-color algorithm for hand posture detection, recognition and tracking. In *Proceedings of International Conference on Mechatronics and Automation (ICMA)* (2010), 934–939.

[4] Kinect Calibration, [Online] http://nicolas.burrus.name/index.php/Research/KinectCalibration. Accessed: 2014-07-02.

[5] Lee, T. and Höllerer, T. 2007. Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers* (2007), 83–90.

[6] Lienhart, R. and Maydt, J. 2002. An extended set of haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing (ICIP)* (2002), I–900 – I–903.

[7] MIP - MultiCameraCalibration, [Software] Version 1.0.0 http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Calibration. Accessed: 2014-07-02.

[8] OpenCV library, [Software] Version 2.4.6 http://opencv.org/. Accessed: 2014-07-02.

[9] OpenNI and Kinect for Android Tutorial, [Online] http://pointclouds.org/blog/nvcs/raymondlo84/index.php. Accessed: 2014-07-02.

[10] OpenNI, [Software] Version 1.5.4.0 unstable https://github.com/OpenNI/OpenNI/tree/unstable. Accessed: 2014-07-02.

[11] Rodriguez, S., Picon, A. and Villodas, A. 2010. Robust vision-based hand tracking using single camera for ubiquitous 3D gesture interaction. In *Proceedings of the 2010 IEEE Symposium on 3D User Interfaces (3DUI)* (2010), 135–136.

[12] SensorKinect, [Software] Version 0.93 https://github.com/avin2/SensorKinect. Accessed: 2014-07-02.

[13] Unity3D, [Software] Version 3.5.1 http://unity3d.com/. Accessed: 2014-07-02.

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

[15] Yeo, H., Lee, B. and Lim, H. 2013. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*. (2013), 1–29.

Daniel Fritz is a graduate student at the Vienna University of Technology, Austria. He graduated at a higher technical college of engineering for electronics and technical informatics. He is studying Media Informatics at Vienna University of Technology and currently writes his master thesis at the Interactive Media Systems group about hand posture and head detection using RGB+D data for natural 3D interaction with a virtual scene on a handheld device.

Annette Mossel is a research assistant and PhD candidate at the Interactive Media Systems group at Vienna University of Technology, Austria. She studied Computer Science and Media with emphasis on computer graphics at the University of Applied Sciences Wiesbaden, Germany and wrote her master thesis in collaboration with the Fraunhofer Institute for Computer Graphics (IGD), Darmstadt, Germany. She was involved in various national as well as international research projects (e.g. EU FP7 project I2Mine), working in the fields of virtual & augmented reality and vision-based wide area tracking technologies. Recent work also focuses on autonomous flight of unmanned Arial vehicles and 3D interaction in mixed reality scenarios. Currently, she is working as a visiting researcher at the MIT Media Lab.

Hannes Kaufmann is associate professor at the Interactive Media Systems Group at Vienna University of Technology and heading the VR&AR group with currently 8 PhD students. After completing his PhD thesis in 2004 on "Geometry Education with Augmented Reality" he did postdoc research in the FP5 EU-ICT project Lab@Future. His Habilitation (2010) was on "Applications of Mixed Reality" with a major focus on educational mixed reality applications. He was project manager of national research projects and participated in projects in the fields of virtual and augmented reality, spatial abilities, geometry, educational mathematics software and published more than 70 scientific articles up to date. His research interests include virtual and augmented reality, optical tracking technologies and applications, motion capture, mobile applications, display technologies, medical applications of VR/AR, psychological topics (Cyberpsychology), education in mixed reality, 3D user interface design, AR/VR and CAD Integration