A Portable, Low-Cost 3D Body Scanning System

Christoph KOPF*[a], Christoph HEINDL[a], Martijn ROOKER[a], Harald BAUER[a], Andreas PICHLER[a]
[a]PROFACTOR GmbH, Steyr (Upper Austria), Austria

http://dx.doi.org/10.15221/13.417

Abstract

Current 3D body scanning setups often requires extensive amount of hardware which needs to be accurately calibrated. Calibrating such systems can be a time consuming and error-prone task. Nevertheless, a well calibrated system generates accurate and fast 3D body scanning results. As a drawback, these setups are often statically built and therefore it is difficult to use them at different locations. New sensor technologies which have been established during the last years offer new possibilities for 3D body scanning tasks. This paper presents a portable, intuitively usable, scale-able and real-time capable 3D body scanner, which is affordable for everyone. Portable, since the system only requires a PC, one or more 3D depth sensors (e.g. PrimeSense Carmine 1.09 [15]). Usable, since the system sensor is hand guided by a user. Scale-able, since any 3D depth sensor with different ranges can be integrated with the system and the scan area can be adapted to the sensor. Real-time, because every new frame from the camera changes the global model in real-time and gives the user feedback of the current status of the global model. To scan a person, the user takes the camera in his hand, starts capturing depth data from all required views by walking around the person to be scanned. As a result a polygon model can be exported and post processed. Due to the flexibility of the system, multiple sensors can be used at the same time. This enables the data of all sensors to be unified in the global model without the need of a complex calibration routine from the auto calibration algorithms of the system.

Keywords: 3D, Surface Reconstruction, Depth Sensor, Tracking

1. Introduction

Common body scanning solutions are often expensive and hardly portable. This paper presents a low cost, portable and scalable solution for fast 3D body scanning which is intuitively usable. Making such a system affordable is mainly achieved by replacing the most expensive hardware parts (e.g. laser scanner, structured light scanner) by recently developed and publicly available low-cost active sensor technology (e.g. PrimeSense Carmine 1.09 [15]) and processing on decent standard computer hardware which is designed for accelerated processing (e.g. GPGPU). Therefore, the proposed system consists of a standard personal computer and a low-cost sensor. The user just takes the low-cost sensor into his hand and freely films the target person from as many views as possible. In the background, a global model is created containing the unified information of the low-cost sensor. User feedback is provided by status information about the current global model. This is important, since the user can identify holes in the global model and easily fill them by filming the person from the missing point of view. Thanks to the flexibility of the system, it is possible to capture data with more than one sensor at the same time without the need of a complex extrinsic calibration process. This paper presents a quality analysis of the generated surface and shows how a setup with more than one sensor can be easily created.

2. Related Work

To provide personalized products with the human body as source, a surface representation (e.g. 3D surface model) is required. Common sensors use techniques like laser triangulation, structured light [1, 2, 9, 10, 11] and time of flight to capture range data. A single sensor is only able to capture data from its point of view. To get data from different views as well, either the senor or object has to be moved [1, 8, 9, 10] or setups with multiple sensors are used [14]. Setups with multiple sensors also leads to increased hardware amount which makes transportation harder and requires enhances calibration routines.

* christoph.kopf@profactor.at; www.profactor.at

The work in this paper focuses on an approach using a setup which is easily transportable, low-cost and multi sensor capable. Therefore, RGB-D sensors such as the PrimeSense Carmine 1.09 are used to capture range data [2, 11, 15]. Early approaches unified aligned point cloud data in a global reference coordinate system [9]. Since RGB-D sensors generate a huge amount of data, this could lead to memory issues. Thus, other approaches often use a single volumetric representation based on implicit surfaces [1, 6, 13]. This approach provides constant memory consumption and the implicit surface consists of averaged data.

Since low-cost RGB-D sensors do not provide tracking facility by default, the sensor position has to be tracked by the generated data. Therefore, the well known ICP algorithm can be used [1]. Since the standard method is not compatible to the real-time requirements, advanced methods with constraints increases the performance dramatically [12]. Due to these constraints, tracking failures could occur. To recover such failures, object recognition methods could help to recover tracking [5].

3. Method

3.1. Environment

The approach in this paper is based on the work of [1] and optimized regarding portability, flexibility, usability and real-time data processing. The system environment consists of four main parts (see Figure 1). These parts are a global model, a 3D sensor, a user who leads the sensor and a computer for data processing and visualization. The captured data is unified in the global model which in this case is described by an implicit function.

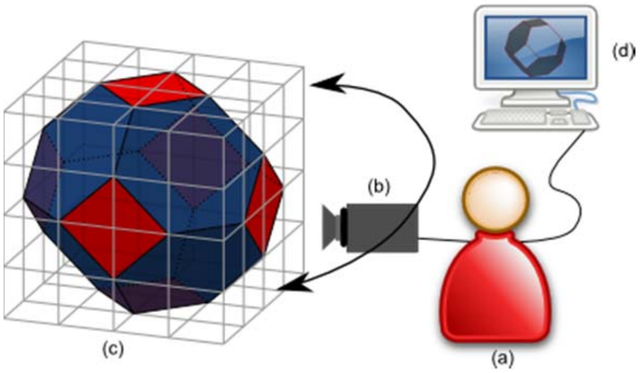

Fig. 1. System components. (a) User, (b) Sensor, (c) Global model, (d) Computer / real-time visualization.

The user just has to take the sensor in his hand and freely move it around the person to scan. Current sensors like the PrimeSense Carmine 1.09 [15], Asus Xtion Pro Live or Microsoft Kinect produces frame rates at about 30 frames per second. Each frame consists of unique information and should be integrated in the global model. Thus, the system must be able to process data in real-time. A big advantage of real-time data processing is that the user immediately gets feedback about the current scan status. Thus, the user is able to identify holes in the global model and to move the sensor in the required position to fill the hole with data [1, 9].

3.2. Sensor Technology

Current sensors like the PrimeSense Carmine 1.09 [15] use a structured light approach. Therefore, an infrared projector emits a light pattern which is detected by an infrared light camera. Since the projector and the camera are calibrated to each other, depth information can be generated. These sensors provide depth information as depth images. Most sensors provide images at a frame rate of 30 frames per second. An analysis of the generated depth data of the sensors showed that there is an error [2, 11]. This error occurs due to an erroneous calibration of the infrared projector to the infrared camera, the lightning conditions and shiny surfaces. The error also increases by distance. Thus, depth data beyond 3 meter should not be used.

Each depth frame contains an error, but gets integrated into the global model. Assuming that averaging over multiple depth images decreases at the errors of lightning conditions and shiny surfaces, the global model consists of averaged and less erroneous data.

3.3. Surface Representation

Transforming each depth image into a 3D point cloud and keep those in a world coordinate system would continuously increase the amount of required memory. To avoid breaking memory constraints, this approach uses a discrete volume representation which provides constant memory consumption.

The discrete volume uses truncated signed distance functions (TSDF) to describe the surface. When the position of the sensor relative to a world coordinate system is known, the TSDF values of the rays can be stored at the corners of the discrete volume (see Figure 2). Further, these values are averaged over multiple views to get rid of the sensor noise. Applying a TSDF from rays of a known sensor position is called data integration. The interested reader might use [1, 2, 13] for further instructions to TSDF.

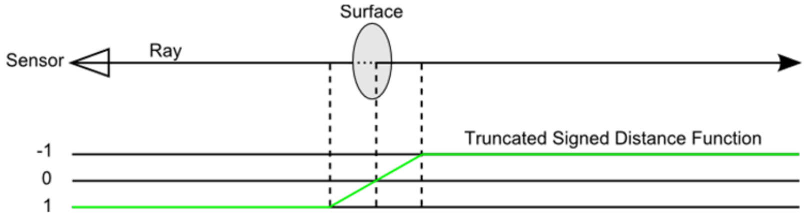

Fig. 2.Truncated signed distance function for a sensor ray.

After scanning, the global model contains unified global 3D information of the surface. The marching cubes algorithm is a well known algorithm to extract a triangulated mesh representation of the discrete volume [7]. This mesh representation further can be exported to common 3D file formats like .PLY, .STL, .OBJ or .3DS.

3.4. Data Processing and Workflow

Decent graphic cards (GPU) provide a massive parallel execution model and two programming interfaces are currently available. These interfaces are nVidia's CUDA [3] on the one hand and on the other hand the OpenCL standard defined by the Khronos Group [4]. While CUDA is only available for nVidia hardware, the OpenCL standard is implemented by many popular vendors like AMD, nVidia and Intel. The work described in this paper uses the OpenCL interface to achieve a better portability.

Since the global model is represented by a discrete volume grid, the required memory size of the global model is constant. This is important when storing the global model on the GPU, because memory is a limited resource on these devices. Figure 3 shows the general workflow the low-cost system.

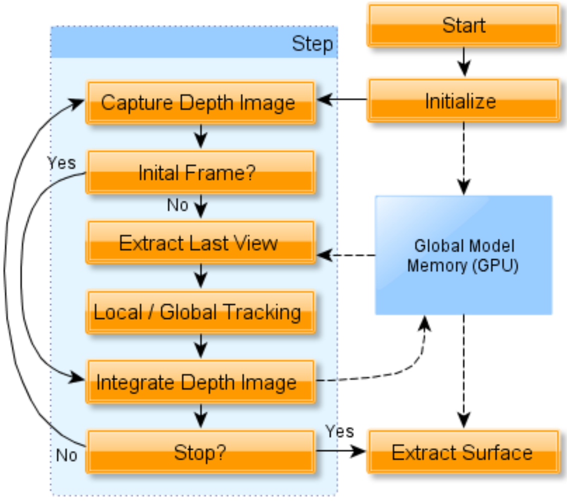

Fig. 3. Workflow of the scanning system.

First, the global model and the sensor position become initialized with their default values. After initialization, the main scan loop is accessed. A single iteration of the main loop can be determined as step. The first block in a step is to capture a depth image from the sensor which can be transformed to a 3D point cloud, defined in the of the sensor coordinate system. If it is the first depth image to integrate, it can be integrated to the global model from the initial sensor position and the next iteration starts. Again, a depth image is captured. Since the sensor position might have changed, the current sensor position has to be estimated by local or global tracking (see Subsection 3.5). Since the global model consists of averaged data, the point cloud of the last sensor view is extracted from the discrete volume instead of using the noisy data from the sensor. The iterative step is processed as long as the user decides to stop. When the user stops the scanning process, a triangle mesh can be extracted from the global model and saved.

3.5. Coordinate Systems and Sensor Tracking

The low-cost system defines the global model in the world coordinate system and the global model is defined within a discrete volume. Since the volume is axis aligned to the world coordinate system, it can be defined by two 3D points. These points are the minimum corner and the maximum corner, both described in the world coordinate system (see Figure 4). Assuming a setup with one sensor, the sensor moves within the world coordinate system and the sensor position has to be tracked.

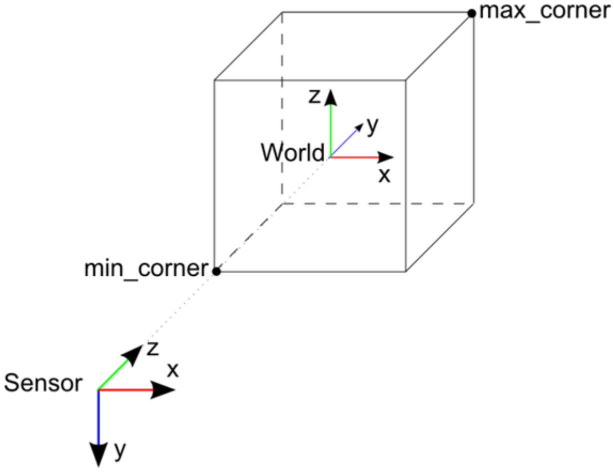

Fig. 4.Coordinate Systems.

Figure 4 shows the initial sensor position in the world coordinate system. It is aligned and points across the y-axis of the world coordinate system and is positioned 400 millimeters in front of the volume. Thus the initial sensor position $t_{initial}$ is a translation along the y-axis, depending on the volume definition. When the user starts the scanning process, the world coordinate system is fixed and the sensor now has to estimate its position within the world coordinate system. The sensor tracking is based on an efficient variant of the ICP algorithm [12]. For the ICP registration at iteration i, the last sensor point cloud $\mathbb{F}_{i-1}$ described in the sensor coordinate system and the current sensor point cloud $\mathbb{F}_i$ defined in the sensor coordinate system as well, gets aligned and a rigid transformation t_i can be calculated. Thus, the sensor position in iteration i can be defined as

$$T = \left(\prod_i^1 t_i\right) t_{initial} \quad (1)$$

This kind of tracking is defined as local tracking. See Figure 5 for a recorded sensor path.

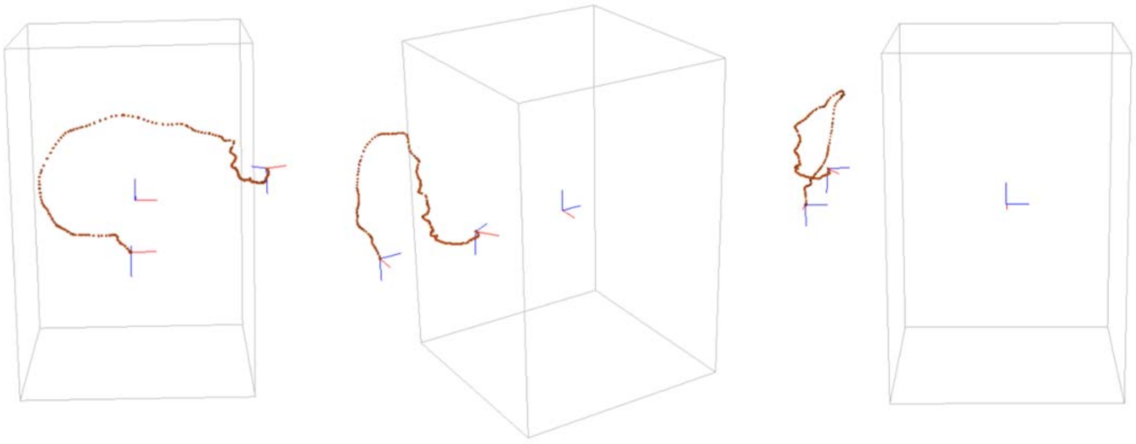

Fig.5. A recorded sensor path from different views, tracked by the ICP algorithm.

This approach uses an optimized ICP algorithm which assumes small sensor movements between the iterations. If the movement is too big, the ICP algorithm produces erroneous results and the sensor tracking is lost. The system is able to detect such tracking failures. In case of tracking failure detection, global tracking comes into account.

3.5.1. Global Tracking

Global tracking helps to recover the camera tracking. If the sensor tracking get lost, the position has to be recovered against a trustful reference, called recovery position $t_{recovery}$. The system continuously stores a recovery position. Global tracking aligns the current point cloud in the sensor coordinate system and the point cloud taken at $t_{recovery}$. The ICP algorithm cannot be used, since it assumes coarse aligned point clouds. Instead, an object recognition approach is used to align the point clouds and thus estimate a relative transformation from $t_{recovery}$ to the searched sensor position. If global tracking is able to estimate a transformation, the sensor position is recovered. In this approach, Candelor [5] is used for global tracking. Candelor is a 3D object recognition library. It allows finding a model in a scene. The same model can be found multiple times in the scene by Candelor. In this case, only one model (current sensor point cloud) has to be found in the scene (reference point cloud).

3.5.2. Using Multiple Sensors

When using multiple sensors, each sensor has to know its position in the world coordinate system. Thus, each sensor has to be tracked. For each sensor the same local tracking procedure is applied. When the tracking gets lost for a sensor, it can use global tracking to estimate its position again. The main problem when using more than one sensor is to find the initial sensor position $t_{initial}$ for each sensor. Two approaches can be used.

If the extrinsic calibration for each sensor is known, the tracking can start from the very first iteration since $t_{initial}$ is known for all sensors. In case, that the extrinsic calibration for the sensor positions is unknown, a “master sensor” can be chosen to define the world coordinate system. Assuming that the initial position of all remaining “slave sensors” is close to the master sensor, global tracking is able to estimate the initial position for the slave sensors as well (see Figure 9). Afterwards, the local tracking will resume tracking of the sensor positions for the master sensor and slave sensors.

Since the position has to be tracked for each sensor, the risk of losing track increases This is due to the fact that the time between the tracking iterations increases when the number of sensors increases, since the workflow has to be executed for each sensor (see Subsection 3.4). Thus, when using more than one sensor it is recommended to move the sensors slowly to ensure that local tracking succeeds.

4. Tests

This section is about quality measurements of the low-cost system compared to a high quality laser scan. Therefore, two setups were used. First, a scan with a hand guided sensor was taken. The second setup used two sensors with an extrinsic calibration and a turntable.

4.1. System Quality

This subsection introduces the quality measure method, the reference scan and the measured quality results of the low-cost system scan compared to the reference scan. As reference object, a dummy was used since it is a rigid object. Thus, both scanners generate data from the same source which is required for the quality measurement.

4.1.1. Method

To measure the quality a reference scan and a test scan are needed. The reference scan is taken by a laser scanner and can be defined as

$$\mathbb{P} = \{p_i\}_{i=0}^{N_p} \text{ where } p_i \in \mathbb{R}^3. \ (2)$$

The test scan is taken by the low-cost system and can be defined as

$$\mathbb{T} = \{t_i\}_{i=0}^{N_t} \text{ where } t_i \in \mathbb{R}^3. (3)$$

Since both scans are described in different coordinate systems, one of the two scans has to be rigidly transformed in $\mathbb{R}^3$ to align the scans. Thus, a transformation matrix m has to be estimated which properly aligns the test scan $\mathbb{T}$ to the reference scan $\mathbb{P}$. Therefore, two steps were used. First, a rough rigid transformation m_{global} is estimated by using the tool Candelor [5]. Second, a local minimum approach can be chosen for a proper alignment. In this case, the local minim approach is the well-known iterative closest point (ICP) algorithm which is robust when two data sets are already roughly aligned. The ICP estimates a rigid transformation m_{local} and the total transformation m can be finally defined as

$$m = m_{local} \times m_{global} \ (4)$$

When m is a known transformation which aligns the test scan well to the reference scan, the aligned test scan can be defined as

$$\mathbb{Q} = \{m \times t_i\}_{i=0}^{N_t} \text{ where } m \times t_i \in \mathbb{R}^3. \text{ (5)}$$

Having two properly aligned scans, nearest neighbor correspondences $\mathbb{C}$ can be created between the reference scan $\mathbb{P}$ and the test scan $\mathbb{Q}$

$$\mathbb{C} = \{(p,q) | p \in \mathbb{P}, q \in \mathbb{Q} \text{ such that } \|p-q\| = \min\|p-q_i\|, q_i \in \mathbb{Q}\} \text{ (6)}$$

For each correspondence $(p,q) \in \mathbb{C}$, the Euclidean distance can be defined as

$$d\big((p,q)\big) = \|p-q\| \text{ (7)}$$

The quality measurement in this context is about extracting statistic values from the Euclidean distances of all nearest neighbor correspondence. Thus we next define the set of distances $\mathbb{D}$ which is

$$\mathbb{D} = \{d(c_i)\}_{i=0}^{N_c} \text{ where } c_i \in \mathbb{C} \text{ . (8)}$$

Finally, all relevant quality values like median, mean and standard deviation can be determined based on $\mathbb{D}$. These statistic values provide an interpretation of the generated data, compared to a high quality scan from a laser scanner.

4.1.2. Reference Object and Scan

The human body is a non-rigid object. Thus, the quality cannot be measured against scans of a real person. Instead, a rigid model of the human body was taken. In this scenario, a dummy was used as reference object (see Figure 6) which is usually used in fashion stores and about 130 centimeter tall.

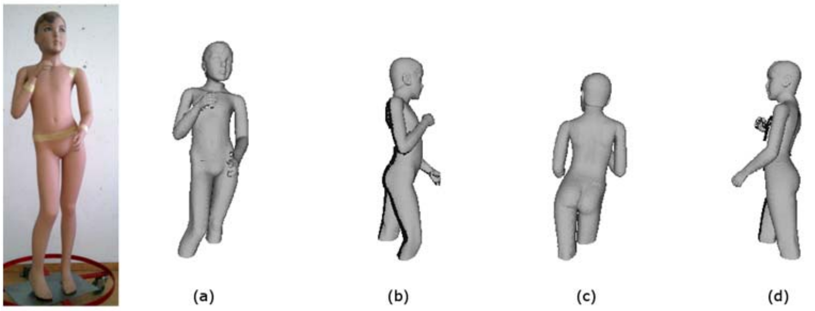

Fig. 6.(a) Scan from top view (b) Scan from left side (c) Scan from back (d) Scan from right.

An accurate system is required to produce a meaningful reference scan of the reference object. In this case, the reference scan was taken by a Sick IVP E1200 triangulation sensor moving on a linear axis. Due to the documentation of the sensor, it generates data at a height resolution of 0.4 millimeter. Figure 6 indicates the reference laser scans.

4.1.3. Quality Measurements

First, a test scan was taken with the low-cost system using a Microsoft Kinect sensor and scanning the reference object from as many views as possible. The test scan was taken on a Dell Alienware M17X, and the data was processed on a nVidia GTX 560M GPU. Figure 7 shows the test scan from different views.

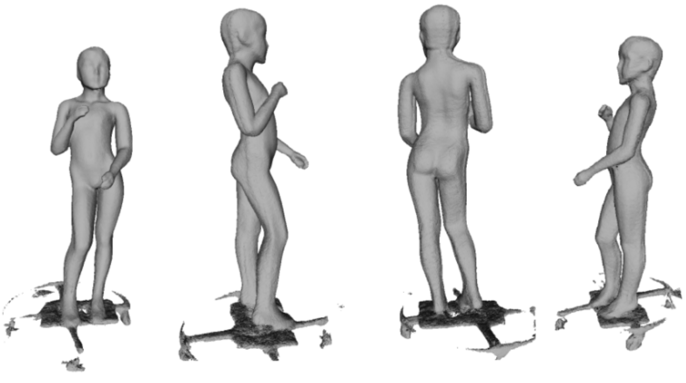

Fig. 1. Test scan rendered from different views

The following settings were applied to the discrete TSDF volume:

- Width: 947 millimeter at a resolution of 512
- Height: 1667 millimeter at a resolution of 512
- Depth: 928 millimeter at a resolution of 256

For each reference scan, the quality measurement method was applied. Figure 8 shows the color coded measurements.

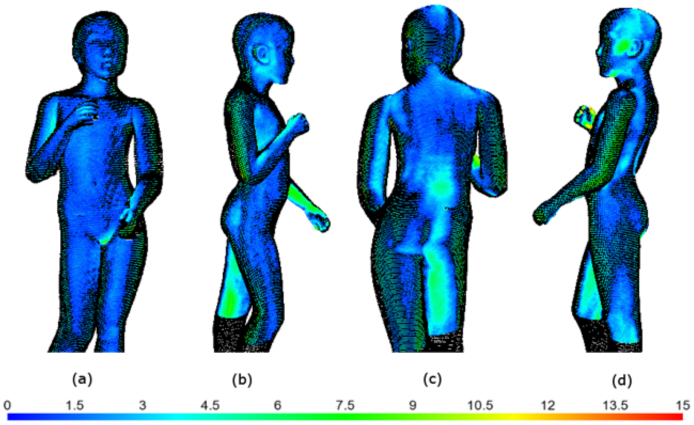

Fig. 8. Distance measurements between the reference scans and test data in millimeter.

As Figure 8 shows, the captured surface of the test scan is close to the reference scans. The biggest differences are visible on the limbs. Especially the legs and arms have some deviation to the reference scan as visible in (b), (c) and (d). The surface of the torso is very close to the reference scan as illustrated in (a) and (b). Since the distances are known, statistic values can be extracted. Table shows the extracted statistic values, for each quality measurement.

Table 1. Extracted statistic values from distance measurements. Corresponds to Figure 8.

	Mean [mm]	Median [mm]	Standard deviation [mm]
(a)	1.87903	1.53814	1.26670
(b)	2.56304	1.98054	1.77960
(c)	2.52908	2.04929	1.62750
(d)	2.90497	2.43542	1.75246

As Table 1 indicates, the mean distance is about 2.5 millimeter. Since the standard deviation is about 1.55 millimeter, the test scans are mainly close to the reference scans.

4.4. System Flexibility

This section shows how two sensors at the same time can be used to scan and how an extrinsic calibration can be easily created. Both sensors unify their data in the same global model.

4.4.1. Scan Using Multiple Cameras

When using more than one senor, each sensor position has to be tracked. The initial position can be either fixed due to a known extrinsic calibration, or estimated due to global tracking of the slave sensor (see Sub-subsection 3.5.2). In this scenario, two sensors were used. Goal was to find an extrinsic calibration for the sensors and then use a turntable to scan the reference object.

4.4.2. Extrinsic Calibration

Initially, the position of the slave sensor is unknown, whereas the initial position of the master sensor is the default position. The default position is defined 400 millimeters in front of the volume along the y-Axis of the world coordinate system, see Subsection 3.5). Thus, global tracking comes into account to estimate the initial position of the slave sensor. Figure 9 shows the sensor paths of the setup, where (1) is the initial position of the master sensor, and (3) is the initial estimated position of the slave sensor. To enable global tracking of the slave sensor, its initial position should be close the master sensor since registration is only possible when both sensors film overlapping regions.

When both sensors know their positions, they can be tracked independently. Now, the sensors can be moved to their final positions. The master sensors final position is shown in Figure 9 (2) and the slave's final position is (4). The final positions of both sensors get saved and can be used as start positions when performing the turntable scan. To ensure a well extrinsic calibration, a rigid object is recommended for this process.

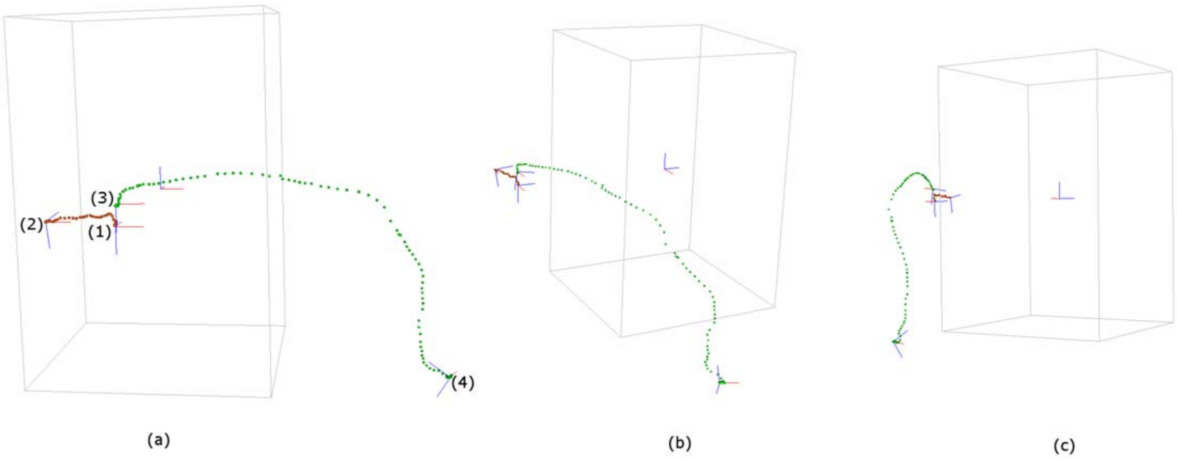

Fig.9. Multi sensor extrinsic calibration and sensor paths form different views.

4.4.3. Turntable Scan

When the initial extrinsic calibration is known, both sensors immediately are able to start the scanning process and their positions can be independently tracked. Thus, after loading the extrinsic calibration the scan can start. In this case, a turntable was used to perform the scan and the sensors were not moved. Again, the reference object was scanned, which allows quality measurements against the reference scan.

4.4.4. Quality Measurements

The method for quality measurements can be applied to the multi sensor scan as well (see Sub-subsection 4.1.1). Figure 10 shows the closest distances from the reference scan to the turntable scan, color coded in millimeter.

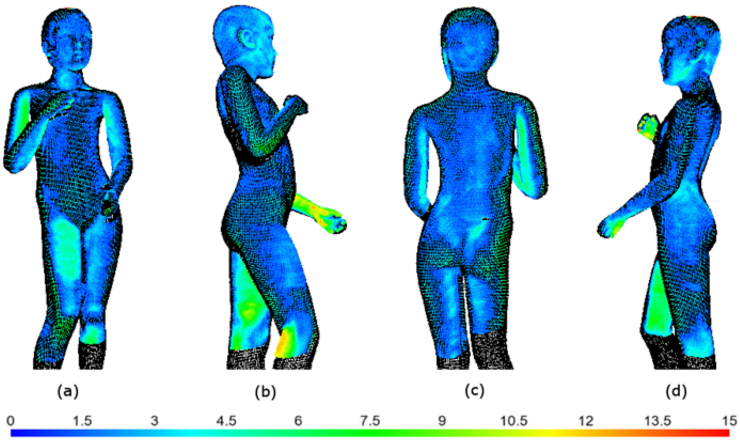

Fig. 10. Quality measurements in millimeter of a scan, taken in a multi sensor setup.

Again, the torso and head is very close to the reference scan. The biggest difference is visible on the limbs at (b) and (c). Statistic values can be extracted as well. They are listed in Table 2.

Table 2. Statistic values of quality measurements corresponding to Figure 10

	Mean [mm]	Median [mm]	Standard deviation [mm]
(a)	2.40313	2.08697	1.32874
(b)	2.89839	2.29675	2.02711
(c)	2.23083	1.96103	1.20602
(d)	2.47158	2.07378	1.55825

The mean is about 2.5 millimeter and the standard deviation about 1.51 millimeters. The turntable scan is close to the reference scan and the statistic values indicate that there is no quality loss of a scan with two sensors compared to a scan with one sensor.

5. Conclusion and Future Work

This paper introduced a scanning system using low-cost sensor technology. Besides a low-cost sensor, a computer or laptop is required to process the global model on. Since the system only requires a sensor and a computer, it is easily portable and quickly set up. The system allows scanning with one or more sensors at the same time. When using more than one sensor, extrinsic calibration is optional. Nevertheless, a setup with an extrinsic calibration can be quickly and intuitively set up. Scanning is performed by moving the sensor or using a turntable to turn the person to scan. But, even a combination of both is allowed. Finally, the system generates a triangle mesh which can be exported to common CAD file formats.

Comparing the result of the system to an accurate laser scan, the system provides an average accuracy of 2.5 millimeters and a standard deviation of about 1.5 millimeters.

In the future, different sensors designed for closer ranges will be tested. Such sensors could be used for high quality head and limb scanning. Another future topic is to enhance the portability. Although the current solution is already easily portable, research on a mobile solution based on tablets (which is capable for real-time scanning) will be done as well. Finally, one of the next key topics is to add accurate color information to the 3D models. This means, generating a texture based on high quality images of digital single-lens reflex cameras.

In summary, the introduced system is easily portable, consists of low-cost components and provides flexible setups without the need of complex calibration routines.

References

1. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A., (2011): "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera.", In Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11), pp. 559-568.
2. Khoshelham, K., Elberink, S.O, (2012): "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications.", Sensors. 2012; 12(2):1437-1454.
3. Parallel Programming and Computing Platform, CUDA, NVIDIA (accessed 2013): http://www.nvidia.com/object/cuda_home_new.html
4. OpenCL - The standard for parallel computing on heterogeneous systems (accessed 2013): http://www.khronos.org/opencl/
5. Object Recognition System "Candelor - Understand 3D" (accessed 2013): http://candelor.com/
6. Curless, B. and Levoy, M, (1996): "A volumetric method for building complex models from range images.", In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96), pp. 303-312.
7. Lorensen, W.E. and Cline, H.E., (1987): "Marching cubes: A high resolution 3D surface construction algorithm.", In Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH '87), Maureen C. Stone (Ed.). pp. 163-169.
8. Newcombe, R.A., Lovegrove, S.J. and Davison, A.J, (2011): "DTAM: Dense tracking and mapping in real-time." In Proceedings of the 2011 International Conference on Computer Vision (ICCV '11), pp. 2320-2327.
9. Rusinkiewicz, S., Hall-Holt, O. and Levoy, M., (2002). Real-time 3D model acquisition. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02), pp. 438-446.
10. Weiss, A., Hirshberg. D. and Black, M.J., (2011): "Home 3D body scans from noisy image and range data." In Proceedings of the 2011 International Conference on Computer Vision(ICCV '11), pp. 1951-1958.
11. Andersen, M.R., Jensen, T., Lisouski, P., Mortensen, A.K., Hansen, M.K., Gregersen, T. and Ahrendt, P., (2012): "Kinect Depth Sensor Evaluation for Computer Vision Applications", Technical Report ECE-TR-6, Aarhus University
12. Rusinkiewicz, S. and Levoy, M. (2001): "Efficient Variants of the ICP Algorithm", In Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling, pp. 145--152 .
13. Osher, S. and R. Fedkiw (2002): "Level Set Methods and Dynamic Implicit Surfaces", Springer.
14. Kainz, B., Hauswiesner, S., Reitmayr, G., Steinberger, M., Grasset, R., Gruber, L., Veas, E.E., Kalkofen, D., Seichter, H. and Schmalstieg, D, (2012): "OmniKinect: real-time dense volumetric data acquisition and applications." Paper presented at the meeting of the VRST 2012.
15. RGB-D Sensor Carmine 1.09 (accessed 1013): http://www.primesense.com/get-your-sensor2/carmine109/