International Journal of Image and Graphics

SEMI-AUTOMATIC SURFACE SCANNER FOR MEDICAL TANGIBLE USER INTERFACES

ANDREI SHERSTYUK

Avatar Reality, Inc
Honolulu, Hawaii, United States
andrei@avatar-reality.com

ANTON TRESKUNOV

Institute for Creative Technologies,
University of Southern California,
Los Angeles, California, United States
treskunov@gmail.edu

BENJAMIN BERG

Telehealth Research Institute, University of Hawaii,
Honolulu, Hawaii, United States
bwberg@hawaii.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Mixing real and virtual components into one consistent environment often involves creating geometry models of physical objects. Traditional approaches include manual modeling by 3D artists or use of dedicated devices. Both approaches require special skills or special hardware and may be costly.

We propose a new method for fast semi-automatic 3D geometry acquisition, based upon unconventional use of motion tracking equipment. The proposed method is intended for quick surface prototyping for Virtual, Augmented and Mixed reality applications (VR/AR/MR), where quality of visualization of scanned objects is not required or is of low priority.

Keywords: Motion Tracking; Tangible User Interface; Surface Scanner.

1. Introduction

Geometry acquisition of physical objects is one of the most common tasks for building virtual and mixed reality applications. Objects that require 3D replication range in size and detail from millimeters (such models are used in surgical simulators), to meters and tens of meters in open-space applications with urban scenes and landscape elements. Accommodating this range of objects requires a variety of surface capturing methods: from CT scans, popular in the medical community, to vehiclemounted video scanners. The use of specialized scanning equipment is expensive and may not provide a cost-effective solution in many cases.

In this work, we explore a new approach to fast geometry acquisition, which does not require additional devices. It is based upon a simple idea that motion tracking equipment, a common element in many VR/AR/MR systems, may be used not only to navigate in and interact with an environment, but also create 3D content for interaction.

This paper presents full detail of our method which was previously published as an extended abstract [1]. It is organized as follows. In section 2, we review existing methods of surface scanning, both commercial and academic. Section 3 explains our motivation and goals, in the context of an existing medical MR system. The surface scanning algorithm is presented in sections 4 to 6. Discussion of possible extensions and applications conclude the paper.

2. Existing methods and related work

Capturing 3D shape of physical objects is a well established discipline with dozens commercial solutions and a large body of academic work. Typical applications are reverse engineering, realistic rendering by means of computer graphics and Augmented Reality.

Available off-the-shelf systems can be classified as contact and contactless. Contact devices measure coordinates of points of contacts between the device and the object surface. Coordinates are obtained either by mechanical means or via non-contact tracking of the measuring tip. Mechanical devices are called touch probes or coordinate measuring machines (CMM). A typical example is a Microscribe series from Immersion. Commercially available tracked devices are represented by laser trackers, that measure coordinates of mirrored spherical probe (e.g., FARO product line), or magnetic trackers (e.g. Polhemus Liberty with measuring Stylus). Our solution belongs to this category as well. However, unlike Liberty Stylus, our method uses the same tracking sensor both for geometry acquisition and for the MR application which interacts with the geometry. This approach eliminates the need for additional equipment and calibration.

It is not always feasible or desirable to use physical contact with the measured surface. Methods that avoid contact include laser scanners based on triangulation and radar laser scanners, based on measuring time-of-flight. Commercial examples are: NextEngine desktop scanner, the most affordable device in the first category; DeltaSphere 3D Laser Scanner is a typical representative of the second group [2]. Many techniques fall into the structured light category, in which the surface of the object is illuminated by single or sequential light patterns. This method allows to establish correspondence between rays from the light source and rays from illuminated points on the surface to camera pixels and then solve for 3D coordinates using triangulation. The light source may be as primitive as a desk lamp with a manually waived stick [3]. Other systems employ moving laser stripes [4] and slide projectors [5], replaced nowdays by video projectors. A previously published survey [6] provides a comprehensive review of coded structured light techniques. The ray correspondence required for triangulation, can also be established by means of stereo vision [7], defocusing [8] or use of silhouettes [9]. When triangulation is not possible, Conoscopic Holography [10] may be a viable alternative. Some computer vision based systems measure only relative depth, as shape from shading [11], from texture [12] or from specularity methods [13].

A large amount of research is devoted to outdoor geometry acquisition using combination of computer vision, inertial tracking and GPS position detection. Recently near real-time performance was achieved in the conduct of the DARPA UrbanScape project in 3D modeling of urban scenes from video [14].

All outlined methods of surface scanning require special equipment. In addition, desktop scanners require sewing to capture human size objects. Acquired 3D geometry must be registered with the world coordinate system before it can be used with a runtime tracking device. The novel method proposed in this paper avoids these complications.

3. Motivation: augmenting human manikins

The surface scanning method presented in this paper was developed for an on-going project to augment human medical simulation manikins with a 3D touch interface. A medical manikin is a life size plastic replica of the human body (or one of its parts), equipped with electrical, mechanical and pneumatic sensors and actuators, that allow to students and instructors to interact with the manikin and simulate various physiological functions of a human organism. Manikins are widely used in medical education, as they provide realistic hands-on experience for students, and can be programmed to simulate many common and uncommon conditions.

Because manikin's interactive capabilities are implemented in hardware and proprietary software, they can not be extended beyond mechanical design limitations and software interfaces. For example, students can palpate the pulse at the wrist of a full body SimMan manikin from Laerdal Medical Corporation [15], and the student action is detected and recorded, because a pressure-sensitive sensor element is located at the left wrist. Most other locations on his surface are "touch-blind". Adding new sensors for educational purposes on-site is impractical because it requires re-engineering of the manikin, and voids all manufacturer warranty and service agreements.

Using tangible user interface (TUI) techniques, we enhance manikin functionality by adding touch-sensitivity to arbitrary locations of the manikin. The manikin itself becomes a single interface object, providing realistic tactile feedback when students palpate in relevant anatomic locations. Student hands are tracked and checked for collisions with the 3D model of the manikin. Simultaneously, the system analyzes hand movements and determines which action is being performed: tapping, percussion, deep/shallow palpating, etc. When a meaningful action-location pair

Fig. 1. Human manikin, augmented with a touch interface. The system components: the Anne Torso manikin from Laerdal Medical Corporation, Flock of Birds tracking system from Ascension, a laptop PC (1.8 GHz CPU, 1 GB RAM, Linux OS). Sports gloves are used to support motion sensors. Bottom: manikin in use. A medical student is performing physical exam by applying percussion on the manikin's abdomen.

is detected, the system triggers an appropriate response function. For example, the manikin groans with pain, when palpated or percussed at a designated tender location. Figure 1 shows the *Anne Torso* manikin, touch-augmented for teaching palpation techniques. The first prototype of our augmented medical manikin was presented at Medical Simulation Workshop in Singapore earlier this year [18], full detail on this system were published recently [19].

Anne is a successor of our first augmented manikin, *SimMan*, shown in Figure 2. Both SimMan and Anne were programmed to simulate pain on palpation due to appendicitis. Students were asked to find the location of abdominal tenderness using percussion technique, and an audible feedback indicator. The surface of the abdomen was approximated by a number of spheres; one of the spheres was assigned to trigger an audible groaning sound, when percussed correctly. The abdominal 3D model for SimMan (Figure 2, bottom right) was constructed using the following steps:

1. measure the extent of the abdomen module;
2. approximate its surface by spheres, using Maya 3D authoring tool;
3. export the spheres from Maya into the simulator;
4. run the simulator, detecting hand-spheres collisions;
5. repeat 2-4 until complete;

Steps 2-4 were repeated, varying the number, locations and sizes of the spheres, until reported collisions matched the physical contacts between user hands and the abdomen module closely. The whole process didn't take a very long time. However, when we had to go through the same routine for our second manikin, it became clear that there must be a better way of doing this.

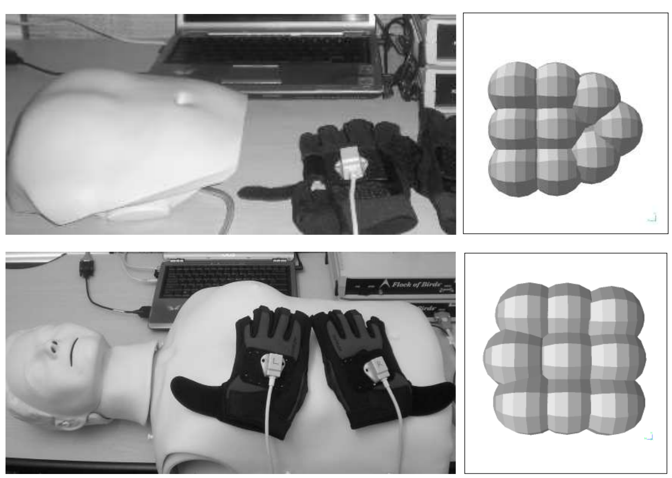

Fig. 2. 3D surface models of SimMan and Anne manikins, approximated by spheres.

4. Surface scanning algorithm outline

The main idea behind our method is to use the motion tracking equipment, already integrated into the simulator, to create 3D models of objects of interest, exemplified by the abdomen and torso modules. Both objects can be represented by a height-field on a plane; both are smooth and lack the necessity for representation of geometric detail less than 20mm. Thus, they can be conveniently built by sweeping a positional sensor along the surfaces and snapping vertices of some underlying grid to the sensor contact locations.

This surface scanner was implemented as a satellite utility within our augmented manikin simulator [19]. To improve the positional sensitivity of the sensor, which has a roughly cubical shape of 25 x 25 x 20 mm, we placed it at the center of a Ping-Pong ball, as shown in Figure 3. This new enclosure allows tracing of surface features smaller than the original footprint of the sensor. The scanning process is described below step by step.

(1) *Grid generation.* A regular grid of quadrilaterals is created in Maya and is used as a template to build the height-field. The dimensions and density of the grid are adjusted to match the size of the object to be scanned.
(2) *Sensor-grid alignment.* A sensor is placed on top of the object (Figure 3, left); the system captures the sensor location and shifts its rest position to and above the center of the grid. At this moment, scanning is initiated.
(3) *Surface scanning.* The user moves the sensor along the surface of the object in sweeping motions and the system updates the height-field interactively, at the frequency of the graphics loop. On each cycle, the closest vertex is found and snapped vertically to the current sensor elevation. The process continues, until all vertices are elevated and the height-field is complete. Progress is monitored visually, as shown on Figure 3.
(4) *Run-time operations on the mesh.* During scanning, the mesh can be saved, reset and convolved with a low-pass filter. Camera parameters (i.e, viewing angle, zoom) and wireframe/solid rendering modes can also be adjusted at run-time.

Fig. 3. Scanning SimMan's abdomen module, left to right: sensor alignment, scanning in progress, completed mesh. Grid size 40 x 40 cm, 20 x 20 points. Scanning time: less than 2 minutes.

5. Implementation detail

The following pseudo-code summarizes the surface scanning algorithm. Procedure `move_vertex()` is an elementary surface building step, which moves a given vertex to the point of contact between the motion sensor and the physical object. The motion sensor is expected to be in constant contact with the surface. Procedure `update_mesh()` is executed at the frequency of the graphics loop, which is 25 times a second in our system, to ensure high interactivity and responsiveness of the scanning process. Normal vectors of the surface are computed less frequently, because this operation is rather costly. We perform this step using a local timer signal. Half-second timeout for normal vectors re-calculations yielded visually acceptable update rates.

```
procedure update_mesh()
begin
    static object mesh;   // surface under construction
    vector P;             // 3D position of the motion sensor
    vector V;             // vertex in the mesh object

    // read the motion tracker data
    P := capture_sensor_position();
    V := find_closest_vertex(mesh, P);

    // update the surface height-field
    move_vertex(V, P);
end

procedure update_normal_vectors()
begin
    if(timer expired)
        for each vertex Vi in the mesh:
            // compute and summarize normal vectors
            // from all polygons Pj that include Vi,
            // save the result as vertex normal Ni:
            Ni := sum(make_polygon_normal(Pj));
        end
        restart timer;
    fi
end
```

6. Algorithm extensions: multi-resolution scanning

The second round of tests of our surface scanner was conducted using the *Anne Torso* manikin, shown in Figure 4. The density of the grid was doubled and an enhanced plastic easter egg sensor enclosure (5 x 3 cm) was used. The eggshell shaped sensor enclosure naturally conforms to high, medium and low precision scanning modes illustrated in Figure 5. Users switch between resolution modes simply by touching the surface with the appropriate side of the egg. Changes in orientation of the eggshell enclosure are captured by the motion tracking system, and modes are toggled as directed. The multi-resolution version of the surface scanning algorithm is outlined below.

```
procedure update_mesh2()
begin
    static object mesh;   // surface under construction
    vector P;             // 3D position of the motion sensor
    vector V;             // vertex in the mesh object
    int    resolution;    // high, medium, low (see Fig. 5 A,B,C)

    // read the motion tracker data
    P := capture_sensor_position();
    V := find_closest_vertex(mesh, P);
    resolution := capture_sensor_orientation();

    // update the surface height-field
    if (resolution = fine)
        // move a single point
        move_vertex(V, P);
    fi

    if (resolution = medium)
        // iterate on vertex neighbors once:
        // 1+4 points, for a regular grid
        move_vertex(V, P);
        for all V's neighbors Ni:  move_vertex(Ni, P);
    fi

    if (resolution = low)
        // iterate on vertex neighbors twice:
        // 1+4+8 points, for a regular grid
        move_vertex(V, P);
        for all V's neighbors Ni:  move_vertex(Ni, P);
            for all Ni's neighbors Nj:  move_vertex(Nj, P);
    fi
end
```

Multi-resolution scanning permits dynamic adjustment of both speed and accuracy to match the local geometric detail of the surface. This method requires each vertex in the grid to have a list of its neighbors. These lists are commonly created during initialization of the grid. If all neighbors of a certain vertex are snapped to the same height, as shown in Figure 5, scanning speed can be further improved by bypassing recently snapped neighbors in the main search loop. In our implementation, we did not use this optimization, as the search time was not an issue. The system was able to update grids of sizes 20x20, 40x40 and 80x80 points at 25 frames per second, which was adequate to our purposes.

The system was running on a consumer level PC laptop with 1 GB RAM and 1.8 GHz CPU, with Linux OS installed. For visualization and 3D scene management, we used *Flatland* open source engine from Homunculus Project [20].

7. Discussion and Extensions

The surface scanning method presented in this paper has several attractive features.

First, it required no additional hardware devices, because the system utilizes motion tracking equipment that is already in place. Therefore, surface scanning cost is limited to the cost of adding one page of code to an existing VR/AR/MR system.

Second, this system eliminates the need for special skills to create 3D content.

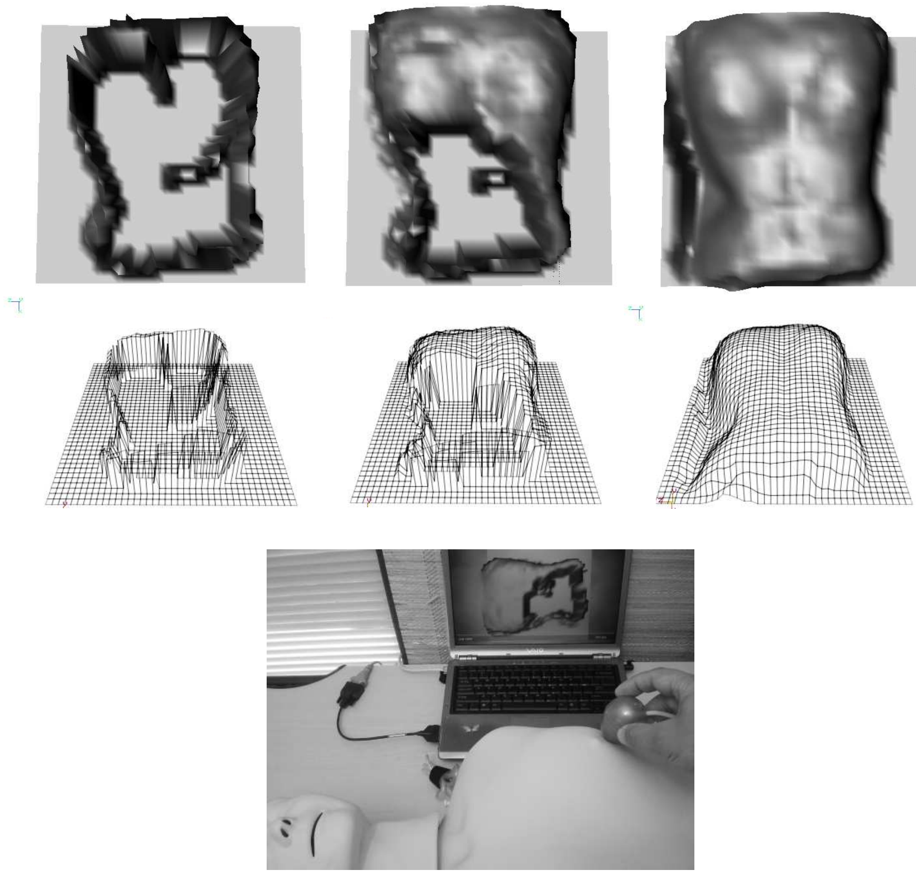

Fig. 4. 3D geometry snapshots of the surface scanning process: initial contour (left), intermediate shape (middle), and final mesh (right). The 3D snapshots are shown "as is", without retouching (see also Figure 6). Grid size 40 x 40 cm, 40 x 40 points. Scanning time: about 8 minutes.

Scanning can be performed by any person, after minimal user-interface control training.

Finally, the scanning process is fast. The 3D models presented in this paper, were all created in minutes and exceeded the quality and fidelity required for their intended use (i.e., collision detection for palpation training). Anne manikin has several 'siblings', manufactured by the same company: *Little Anne*, *Little Junior*, *Baby Anne* – all of them can be scanned in minutes, using our method.

In addition, the proposed technique naturally addresses the long-standing problem of distortions in the magnetic environment, that plagues most magnetic tracking system. During scanning, distortions are imprinted into vertex positions of the

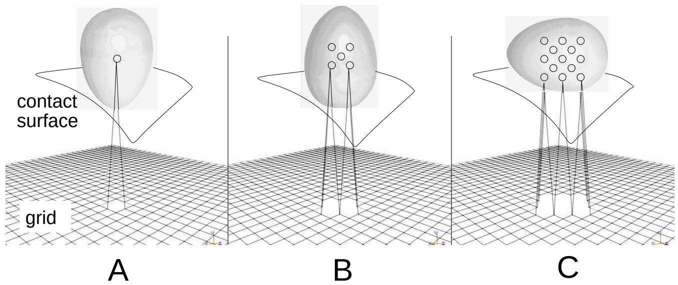

Fig. 5. A plastic eggshell provides three intuitive scanning positions: (A) contact with the sharp end moves one vertex; (B) the obtuse end moves the vertex and its closest neighbors (5 vertices per move); (C) contact with the elongated area makes the system to iterate on vertex neighbors twice (13 vertices per move).

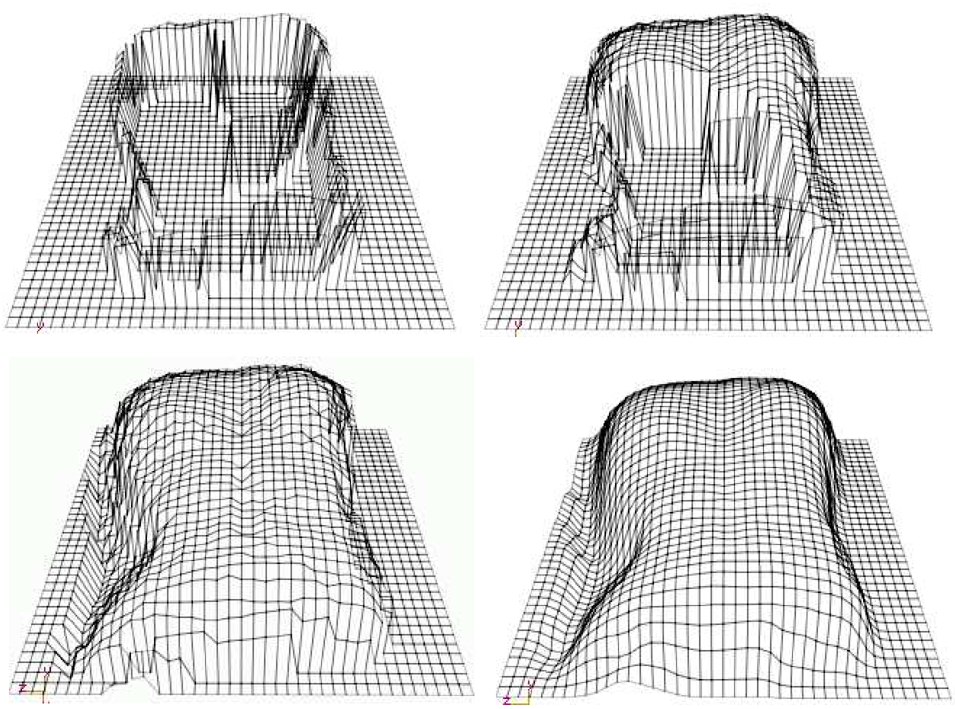

Fig. 6. Wireframe views of the progressive surface scans. The bottom-right shape is smoothed with a low-pass filter.

surface. Normally, these tracking artifacts are not welcome and may render the resulting mesh useless, if used elsewhere (i.e., under different magnetic environment).

However, if the shape is used with the same simulator at the same location, the distorted shape is perfectly preprocessed for correct hand/surface collision detections. The shape produced with our scanner is a reflection of the true surface of the physical object (a manikin) in the magnetic environment of the lab and its own inner 'organs'. A perfect surface model would require a complicated tracker calibration, to take magnetic distortion into account. Utilizing the same equipment for surface acquisition effectively mitigates this problem. This feature is illustrated in Figure 7.

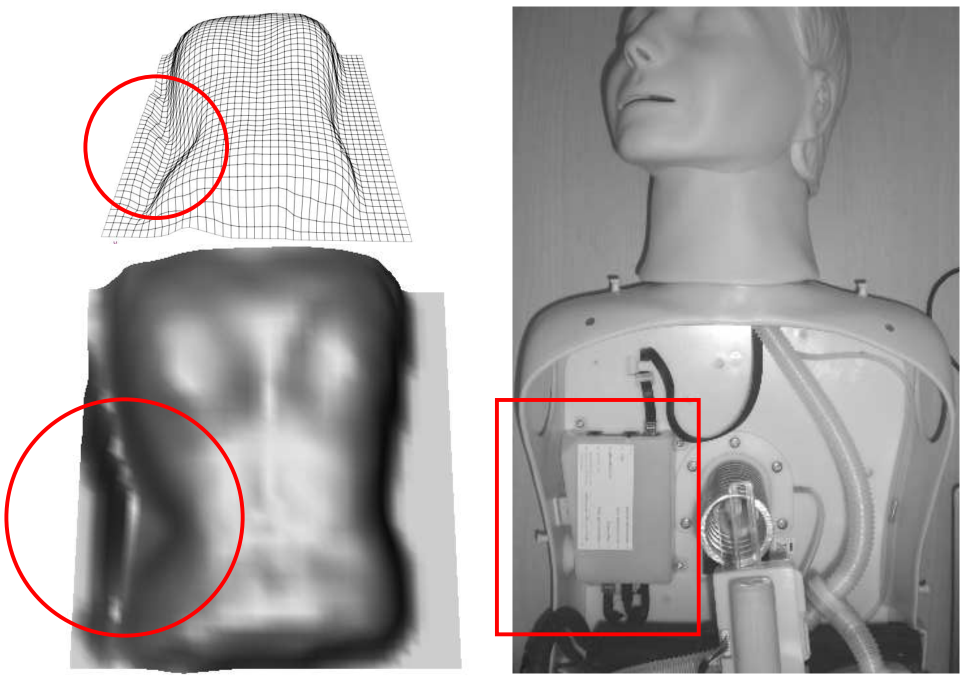

Fig. 7. Distorted area in the scanned surface, denoted by a circle. A 'postmortem' examination revealed that Anne was engineered asymmetrically: a large metal switchbox installed in her upper right abdominal area, caused distortions in the magnetic field. As described in the text, embedding these distortions into the model effectively cancels their effect during model use.

The proposed method has limitations. It can only produce shapes that are either a height-field on a plane, as in the case of abdomen simulators described above, or a height-field on a sphere or a cylinder, which are more suitable for limbs. Also, surface tracking does not capture normal vectors of the surface. Thus, the resulting shape will always appear smooth. These limitations may render surface tracking unsuitable for many applications. However, in the field of medical simulation and training, when the human body parts must be captured for TUI purposes, these issues do not impact on operational utility. Scanned anatomical shapes are expected to be smooth, without holes and sharp features.

Surface tracking offers a number of interesting extensions and applications, that are briefly described below.

7.1. *Multi-sensor scanning*

For magnetic tracking, the form factor of the motion sensors defines how many sensors a user can handle simultaneously. In our experiments, we used only one sensor, which was sufficient for the task. However, it seems reasonable to suggest that using several sensors, attached to user's fingertips, could reduce scanning time. For example, sensors of Mini Birds 800 system from Ascension measure 10 mm x 10 mm x 5 mm and weigh 1 gram only. These sensors are thus good candidates for multi-finger version of surface tracking.

7.2. *Vertex painting*

In addition to surface scanning, motion tracking can be used for creating educational and training scenarios. One example includes instructor selection of an examination technique (e.g., percussion), a system response on a contact event (e.g, audible "ouch") and confirmation of the predesignated tender area by simply touching the manikin's surface. The instructor can effectively "draw" the tender areas on the manikin's 3D surface, by tracing his or her tracked hand on the real manikin object. The level of pain can be stored and displayed as vertex colors, for example, red. Note that such use of tracking technology allows programming of teaching scenarios without writing a single line of code, and the ability to change it on-the-fly during the exercise.

7.3. *Manual surface sculpting*

The prototype version of the surface tracking system assumes that the sensor (or sensors) closely follows the surface of the source object. However, an exact replica of the object may not be the ultimate objective, but rather a first draft of the desired shape. For example, the user may scan his or her own face to acquire a basic canvas for creating a caricature (e.g., a Pinocchio mask). Similar artistic extensions are discussed by creators of *AR-Jig* scanning device [17], although their initial surface acquisition method is very different from ours.

Use of 3D brushes, pushing and pulling operations and various filtering will enable a user to further sculpt and refine the model. The job division between hands can be arranged naturally, by parenting the grid as a solid object to a non-dominant hand (which must be also tracked), and using the dominant hand to manipulate vertices. Tracking can be switched on and off (e.g., by flipping the object upside down), so the user can change or adjust sculpting tools, take surface snapshots, or simply rest his or her fingers. A pinch glove and a Head Mounted Display could make interactions with the model and the UI controls more comfortable.

7.4. *Ball tracing*

Ball tracing is a mental experiment, loosely related to our method by the same concept of using tracking equipment for surface sampling. As the name implies, ball tracing involves visual tracking of a large number of Ping-Pong balls bouncing in a large closed space, such as a living room with furniture. The tracking system must detect locations where balls reverse their trajectories. Every hit gives a position, a normal vector and color information for the point-based representation of the environment. Given enough time (and balls), the cloud of sample points will become dense enough to allow polygonal reconstruction of the scene. The scene can also be rendered directly with a point-based rendering algorithm, such as QSplat [21]. To make the scanning process completely automatic, a special machine may be used to shoot balls in random directions continuously. Such machines are commonly used in table-tennis classes. This addition would make ball tracing resemble ray-tracing more closely, although the task here is reversed.

The ball tracing idea predates the surface tracking method, described in this work, but it has not been implemented. Thus, the question whether it is practical remains open.

8. Conclusion

The novel surface tracking technique, presented in this paper, opens new and interesting directions in the field of fast content creation for VR and MR systems. Surface tracking can be used off-line, during application development. Also, it may be applied in real-time, for creating 3D models of physical objects on demand, that can be used, stored, and even shared between local and remote VR/AR/MR collaborators on-line.

Acknowledgments

We wish to thank Mark Fiala for lively and productive discussions on surface tracking techniques.

References

1. A. Sherstyuk, A. Treskunov, B. Berg. Fast Geometry Acquisition for Mixed Reality Applications Using Motion Tracking. *Proc. International Symposium on Mixed and Augmented Reality, ISMAR'08*, Cambridge, UK, September (2008).
2. A. Cracknell and L. Hayes. *Introduction to Remote Sensing*, (2 ed.), London: Taylor and Francis, (2007).
3. J-Y. Bouguet and P. Perona. 3D Photography on Your Desk. *Proc. International Conference on Computer Vision*, Bombay, India, January (1998).
4. A. Gruss, S. Tada, T. Kanade. A VLSI Smart Sensor for Fast Range Imaging. *Proc. Conference on Intelligent Robots and Systems*, (1992).
5. M. Proesmans, L. Van Gool, F. Defoort. Reading between the lines-a method for extracting dynamic 3D withtexture. *Proc. Sixth International Conference on Computer Vision*, 4-7, January (1998).

6. J. Salvi, J. Pages, J. Batlle. Pattern Codification Strategies in Structured Light Systems. *Pattern Recognition* 37(4), pp. 827-849, April (2004).
7. T. Kanade, A. Yoshida, K. Oda, H. Kano, M. Tanaka. A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications. *Proc. Computer Vision and Pattern Recognition Conference*, (1996).
8. A. Pentland, T. Darrell, M. Turk, W. Huang. A Simple, Real-Tie Range Camera. *Proc. Computer Vision and Pattern Recognition Conference*, (1998).
9. W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. Mcmillan. Image-Based Visual Hulls. Proc. ACM SIGGRAPH, (2000).
10. G. Sirat, and D. Psaltis. Conoscopic Holography. *Optics Letters*, 10(1), pp. 4-6, (1985).
11. B. Horn and M. Brooks (Eds.). *Shape from Shading*. Cambridge: MIT Press, (1989).
12. A. Blake, and C. Marinos. Shape from Texture: Estimation, Isotropy and Moments. *Artificial Intelligence*, 45(3), October (1990).
13. G. Brelstaff. Inferring Surface Shape from Specular Reflections. *Ph.D. Dissertation*, University of Edinburgh, (1989).
14. P. Mordohai, J.-M. Frahm, A. Akbarzadeh, B. Clipp, C. Engels, D. Gallup, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewnius, H. Towles, G. Welch, R. Yang, M. Pollefeys and D. Nistr. Real-Time Video-Based Reconstruction of Urban Environments. *3D-ARCH International Workshop: 3D Virtual Reconstruction and Visualization of Complex Architectures.* Zurich, Switzerland, July (2007).
15. Laerdal Medical Corporation, http://www.laerdal.com
16. Ascension Technology Corporation, http://www.ascension-tech.com
17. M. Anabuki and H. Ishii. AR-Jig: A Handheld Tangible User Interface for Modification of 3D Digital Form via 2D Physical Curve. *Proc. ISMAR Symposium*, Nara, Japan, pp. 55-64, (2007).
18. Simulation Workshop at the 18th Annual Asia Pacific Military Medicine Conference, Singapore, http://www.apmmc.org/, April (2008).
19. A. Sherstyuk, D. Vincent, B. Berg. Creating Mixed Reality Manikins for Medical Education. *Proc. of the 18th International Conference on Artificial Reality and Telepresence (ICAT)*, Jokohama, Japan, December (2008).
20. The Homunculus Project, http://www.hpc.unm.edu/homunculus.
21. S. Rusinkiewicz, M. Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes, *Proc. ACM SIGGRAPH*, pp. 343-352, (2000).

Photo and Bibliography

Andrei Sherstyuk earned his first higher education degree in physics from Novosibirsk State University, Soviet Union, in 1989. After working several years as a consultant for AT&T, he started an M.S. program in Computer Science at Caltech and graduated in 1994. He defended his Ph.D. dissertation "Convolution surfaces in computer graphics" at Monash University, Australia. In 1999, Andrei Sherstyuk moved to Hawaii to work on the world's first photo-realistic CG film Final Fantasy, The Spirits Within. In 2002, Andrei Sherstyuk joined Telehealth Research Institute, where he directed research and development on Virtual Reality systems for medical education and training. Now he works on the next-generation online virtual world *Blue Mars*, at

Avatar Reality. His recent publications are in the area of Virtual/Augmented/Mixed realities and user interface design.

Anton Treskunov graduated from the mathematics department at Moscow University. Dr. Treskunov completed his Ph.D. in Computer Science at the Keldysh Institute of the Russian Academy of Science. His Ph.D. work focused upon issues related to the application of machine vision techniques for automated industrial inspection. In 1999, he moved to United States, where he worked as a multimedia software developer at Phantom Reality. In 2003, Dr. Treskunov joined ICT as the lead researcher. Among his many responsibilities, Dr. Treskunov oversees software engineering decisions related to the development of the group's various mixed reality projects. His research at ICT focuses upon computer vision based techniques for enhancing tracking and rendering in mixed reality systems.

Benjamin Berg, Clinical Professor of Medicine, was born in London UK, and received his M.D. degree from the Tufts University School of Medicine. Dr. Berg retired from the US Army in 2005. He served as the Internal Medicine Residency Director, Chairman of the Institutional Review Board, and as the Director of Graduate Medical Education at Tripler Army Medical Center, Honolulu HI. Dr. Berg has conducted multiple clinical telemedicine and distance learning programs and now holds an appointment as the Director of Simulation at the Telehealth Research Institute, University of Hawaii, John A. Burns School of Medicine.

His primary research interests include active work in advanced telemedicine for intensive care unit patients, simulation based medical education, and distance learning using simulation and advanced technologies.