Capturing Close Interactions with Objects Using a Magnetic Motion Capture System and a RGBD Sensor

Peter Sandilands, Myung Geol Choi, and Taku Komura

Institute of Perception, Action and Behaviour
School of Informatics, University of Edinburgh

Abstract. Games and interactive virtual worlds increasingly rely on interactions with the environment, and require animations for displaying them. Manually synthesizing such animations is a daunting task due to the difficulty of handling the close interactions between a character's body and the object. Capturing such movements using optical motion capture systems, which are the most prevalent devices for motion capturing, is also not very straightforward due to occlusions happening between the body markers and the object or body itself. In this paper, we describe a scheme to capture such movements using a magnetic motion capture system. The advantage of using a magnetic motion capture system is that it can obtain the global data without any concern for the occlusion problem. This allows us to digitally recreate captured close interactions without significant artist work in creating the scene after capture. We show examples of capturing movements including opening a bottle, drawing on a paper, taking on / off pen caps, carrying objects and interacting with furniture. The captured data is currently published as a publicly available database.

Keywords: character animation, motion capture, environment interactions.

1 Introduction

Scenes of manipulating objects are often used in films and computer games. Such scenes include characters simply carrying objects, manipulating daily utilities such as pencils, screw drivers and hammers and using kitchen ware such as knives and bowls. Automatically synthesizing such movements of the body as well as the objects is a daunting task due to the complex coordination between the objects and the body. The body sometimes slides over the surface of the object, stays rigidly with it, or avoids collisions with the object while conducting complex interactions such as wrapping and winding movements. Although some techniques to automatically synthesize such movements have been implemented, they are still in the early stage of research and are not mature enough to be used in the production pipeline.

Another option to synthesize such animation is to capture the raw human motion. In fact, in most computer animation productions, the optical motion capture system is used to capture such human and object movements. In such cases, reflectors are attached to the fingers as well as the objects and the 3D movements are tracked by high resolution infra-red cameras. However, usually such capturing process requires a laborious post-processing, due to the significant amount of occlusions that happen between the body

M. Kallmann and K. Bekris (Eds.): MIG 2012, LNCS 7660, pp. 220–231, 2012.

and the object. For synthesizing a realistic animation of close interactions, not only the movements of the actor and objects, but also their geometry and the morphology of the actor must be captured precisely. Consider picking up a cup: With longer bone lengths, the fingers would likely penetrate into each other and the cup due to the joint angles being great and the distances being small. An incorrectly specified object could cause penetrations and unnatural motion.

In order to synthesize animations of close interactions between the body and the manipulated object, we propose a new framework to capture such scenes using a magnetic motion capture together with a Kinect tracking system. The main advantage of magnetic motion capture system is that it does not suffer from the occlusion problem. A secondary advantage is that these sensors record both translation and orientation, giving six recorded degrees-of-freedom in a single marker. This allows for fewer markers than would be required for an optical system to get the same level of information. Although there can be deformation of the space globally in a magnetic system, locally the relationships between points are accurate, which is most important when dealing with close interactions. The sensors are also small enough to be attached to the fingers. We attach the sensors both to the actor and to the object, and scan the object's geometry using a Kinect system. Using such an approach, we can calibrate the geometry data of the object and the magnetic sensor data. As the number of sensors are limited, the sensors are attached to the finger tips and the back of the hand and the rest of the body joint angles are computed by inverse kinematics.

We show examples of capturing various movements including opening a jar, drawing on a paper, taking on / off pen caps, picking up and carrying objects, and sitting on a chair. The captured data is currently published as a publicly available database [1].

2 Related Work

We first review about the magnetic motion capture system. Next, we review methods to capture humans manipulating objects.

2.1 Magnetic Motion Capture Systems

Magnetic motion capture systems have the advantage in capturing close interactions as they do not suffer from the occlusion problem. Previous AC field systems[1, 2] suffered from eddy currents which are induced by the surrounding metals. These currents produce their own smaller magnetic fields and thus the sensors receive a distorted magnetic field and the accuracy degrades. A common approach nowadays is to use a DC field. In the DC field approach [3, 4], a gap is produced between the timing that the transmitter produces the field and the sensors detect the field such that the eddy currents disappear. Fields are produced in three directions sequentially to obtain the 3D location of the sensors as well as their orientation in the environment. However, these DC sensors are less accurate and are negatively affected by the earth's magnetic field, power outlets and electric motors, which modern AC systems are resilient to. DC sensors also tend

[1] available at `http://www.ipab.inf.ed.ac.uk/cgvu/`

to be slightly larger and less accurate. For these reasons the system that we use in this research is an AC system: the LIBERTY system by Polhemus Inc.

2.2 Capturing Scenes of Object Manipulation

Recently, synthesizing movements of close interactions is attracting researchers [5–7]. Methods based on optimization and contact constraints have been proposed. Liu synthesizes motions such as opening the top of a bottle by specifying the contact points between the finger tips and the top [6]. The movements that can be synthesized by such a method is limited to those mainly involving the finger tips. The range of movements that can be synthesized are extended to those involving the palm in [8], in which the wrist position and orientation, alongside the object transformation are captured and then the finger movements are synthesized. Although automatic synthesis of the hand movements is a long time goal of computer animation, at the moment we still need to rely on captured motions for applications such as animation and computer games.

Methods to use depth sensors together with vision sensors have been implemented to capture close interactions such as manipulating a mobile phone [9]. Kry et al. propose a method for capturing the contact force between a hand and an object [5]. The system captured the joint positions using optical motion capture, and the joint compliance via torque sensors and fingertip pressure sensors. Using this, they were able to physically simulate the captured motions on objects with different properties. Vision-based methods suffer from occlusions between the body and the objects.

The occlusion problem can be coped with by using a magnetic motion capture system. Mitobe et al. [10] present a magnetic motion capture system for capturing a pianist's hand motion. Although the hand motion can be captured using this technique, the object geometry or the interaction between the hand and the object cannot be captured. Our aim in this paper is to introduce a framework that does not suffer from occlusions by combining the magnetic sensors with the geometry data of the object captured by depth-based sensors.

3 Method

3.1 Overview

The capturing session is composed of the following processes.

1. Place the magnetic markers in scene at landmark positions and record these sensor readings. Scan using the Kinect and also identify the landmark positions. We can then compute the transformation between the scene (scanned optically by the Kinect) and the magnetic marker space to align these two sensor recordings.
2. Obtain the geometry of the object via the Kinect sensor. The object must be placed in such a way that the surface is mostly visible. We can place a magnetic marker on each rigid section of the object and calculate their offsets using the transformation calculated in (1).
3. The additional magnetic markers can then be placed on the actor and the interaction can be performed, recording the sequence of transformations of the markers.

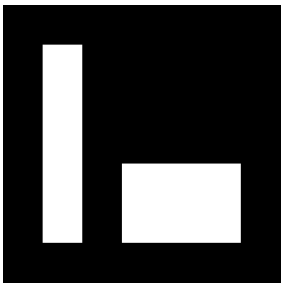


Fig. 1. An example visual marker

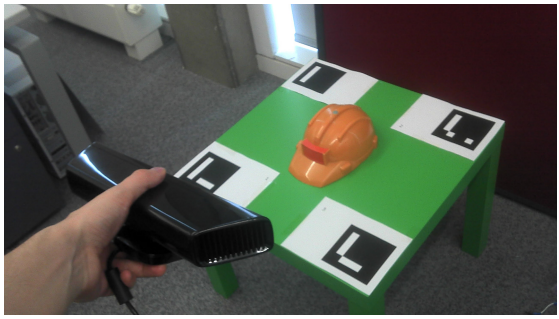


Fig. 2. The environment for scanning an object

4. In case of marker slip/changing location, in each new take we must recalculate the relative position of the magnetic marker to the object. To prevent having to reconstruct the geometry in each take, we can align the previously reconstructed geometry to the current take. This allows us to have starting configurations that obscure some of the important geometry or would make the geometry hard to reconstruct (such as having a bottle lying down on its side), whilst retaining an acceptable quality reconstruction of the object.
5. For motion playback, the actor and the object are fitted to the markers. The objects are fitted using the precomputed transformation in (4).

Each of these processes are described in the following subsections.

3.2 Aligning Magnetic Markers to the Scene

We capture the scene using two sensor systems: the Kinect, which gives us the geometry of the object, and the magnetic marker system, which gives us the movement of specified points in the scene. These two systems are unaligned and have different scaling so can be difficult to use together. In order to align the two sensor spaces, we must compute a transformation from the magnetic marker system to the scene scanned using the Kinect. This transformation stays consistent as long as the magnetic transmitter stays static and no new metal is brought into the environment.

Fig. 3. Examples of reconstructed 3D objects, captured by the Kinect sensor

To compute this transformation we first need to define a world coordinate system that is defined by the Kinect scan. Four visual markers, as shown in Figure 1, are embedded in the environment and the scene is scanned using the Kinect. By detecting at least two of the oriented visual markers in the scene using ArUco [11], we can estimate the Kinect's transformation relative to the visual markers and thus define a world coordinate system using the relative translation and rotation between the two markers. We then place two magnetic markers at known positions in the world coordinate system (such as the corners of the visual markers). Using the positions of the magnetic markers in the world coordinate system and the magnetic motion capture space, we can compute the transformation matrix to align the two environments:

$$X^w = \mathbf{M}_{w \leftarrow m} X^m \tag{1}$$

where X^m is an oriented point defined in the magnetic motion capture space, X^w is the corresponding oriented point in the global coordinates, and $\mathbf{M}_{w \leftarrow m}$ is the conversion matrix between the magnetic motion capture space and the global coordinate system.

We can compute the transform by taking the difference in length between these known markers in both spaces and scaling the magnetic space appropriately, applying the average difference in translation between the known markers to the magnetic space, and finally rotating by finding the axis parallel to the vectors between the two markers in both magnetic and Kinect space, rotating around this angle to align the vectors between the markers in both spaces.

3.3 Kinect Object Reconstruction

In order to acquire the geometry of the object we are interacting with, we use the Kinect sensor to capture surface information about the object [12]. We capture the object geometry using the Kinect sensor by placing the object into the environment with visual markers that was processed in Section 3.2 (see Figure 2). By scanning the object using the Kinect from multiple directions, we can build up a 3D coloured point-map of the scene as follows[2]. We first project each pixel to a 3D point using the depth data and the computed transformation of the Kinect for each captured frame. This reconstructed pointcloud has the colour information for each point embedded in it. We can then remove points belonging to the floor plane, which the visual markers are on, to leave only

[2] A technique adapted from Nicholas Burrus' RGBDemo:
`http://labs.manctl.com/rgbdemo/`

points related to the object. From this point representation of the object, we can recover the surface information as follows. First, downsampling the data using a voxel grid allows us to be able to process the data in far less time, without significant loss of detail. This downsampling approximates the pointcloud by replacing each voxel that contains at least one point with a single point at their centroid. We found that then performing Poisson Surface Reconstruction [13] gave us similar results to the original geometry. Images of some of the reconstructed 3D surfaces using this method are shown in Figure 3.

After the object is reconstructed, we attach a magnetic marker whilst keeping the object in the same position. We can then compute the initial offset of the marker by the difference between the object and marker transformations.

3.4 Marker-Object Transform Calculation (Dealing with Marker Slip)

To compute the relationship between the magnetic marker attached to the object and the object model, we need the aligned model and marker at some time in a take. We make the assumption that the marker does not change relative position significantly during a single take. However, we notice that over a number of takes, due to interaction with the object, the marker can move relative to the object. It is also possible that the marker may have to be moved between takes in order to best capture different interactions. This renders the initial transformation between object and marker invalid, and so must be recomputed. We do this by recomputing the transformation at the beginning of each take, using the initial marker reading and a Kinect scan of the scene. In order to not have to reconstruct the full geometry of the object for each take, we use a single model for each object, but align it to the initial scan of each take. This gives us a consistent model across motion takes, but also allows for the change in relative marker position in each take.

To assist the user in aligning the object, we use automatic object template alignment methods on the pointcloud representation of the object and the scene. We use the technique specified in [14] and briefly explain it here. Given that we already have a point-based representation of the object with estimated normals, we can compute the 'Fast Point Feature Histogram' descriptors. We call this the object template. We can also compute these descriptors for the target scene (with the ground plane removed) and call this the scene template. The FPFH descriptors encode relative normal and positional data locally for each point in the pointcloud. They do this by having a local 'sphere of influence' around each point p_i, and computing the relative orientation and orthogonal vectors for every point p_j in this sphere as compared to the initial point p_i. After this has been computed for the pointcloud, a second pass is performed, which then adds a weighted sum of the histogram of each neighbouring point based on its euclidean distance to that neighbour. To align the template to the target object, we use the Sample Consensus Initial Alignment method also outlined in [14]. Briefly, this selects a subset of the features on the template and tries to find similar points on the object pointcloud for each feature. Once these similar points are found, a correspondence is chosen for each initial sample point on the template randomly out of the list of similar points. The rigid transformation defined by these correspondences is then applied and a distance-based error metric is applied. After a set number of steps, the best alignment

(a) (b) (c)

Fig. 4. An alignment between a partial scan and the object is shown here in (c). (a) shows the partial scan of the object to be aligned: an upright bottle. (b) shows the target scan: a bottle lying on a table. The alignment is computed without regard to the points belonging to the floor, but (c) shows the whole scan in (a) transformed by the computed alignment transformation matrix.

is returned as a transformation from the template to the object. We can then calculate the inverse transformation to transform the object into the initial position for the motion capture session. An example of an object model fitted to the point cloud is shown in Figure 4.

As the Kinect scan is only done from one direction we found the automatic fitting does not perform well in all circumstances, especially when the object has a symmetric shape. It is also important to scan the object from the direction that the magnetic sensors are not visible as it strongly affects the fitting process. In the case that the automatic fitting does not converge well, we manually complete the fitting process.

Once this procedure is done, we know the transformation of the object and the object magnetic marker in the first frame of the motion take, and so can compute the offset by the difference between these transformations.

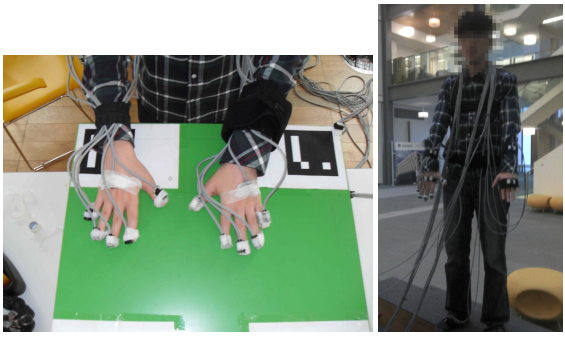


Fig. 5. Sensors attached to the hands (left) and body (right)

3.5 Capturing Motion via Magnetic Sensors

In order to capture the human motion, we use magnetic markers due to their accuracy and the desire for continuous data. The finger tips are often occluded by objects, the hand itself when grasping, or when moving. This led us to use a sensor that could not be occluded.

We have two separate marker sets for capturing the human motion, one for intricate motions that require two hands, and another set for capturing full-body motion. When we capture intricate motions, we place a sensor on each distal finger section, and a sensor on the back of the hand as shown in Figure 5, left. When capturing full-body motion, we place the sensors for the right hand as we do for the intricate motion configuration, but have a single sensor on the head, torso, left forearm, left hand, left foot and right foot as shown in Figure 5, right. This allows us to capture accurately a single hand interaction with many objects using a limited number of sensors (see Table 1 for sensor details).

Both of these marker sets require a total of 12 sensors, and so allows us 4 sensors for objects in the scene and for marking known locations. The object or objects require a magnetic sensor for each rigid body we wish to capture, to recover the orientation and position of each object. We export the data for a capture as a ASCII encoded sequence of positions and orientations, tagged by sensor ID. We can then parse these in our system, or can import them using a Python script for Autodesk Motionbuilder. As previously mentioned, some joints require inverse kinematics to compute the required angles. To ensure these angles are similar to the performed joint angles we use the obtained orientation data from the magnetic sensors for the fingertips to limit the solutions of the IK for the fingers, giving more accurate results than if we were to use the positional data alone.

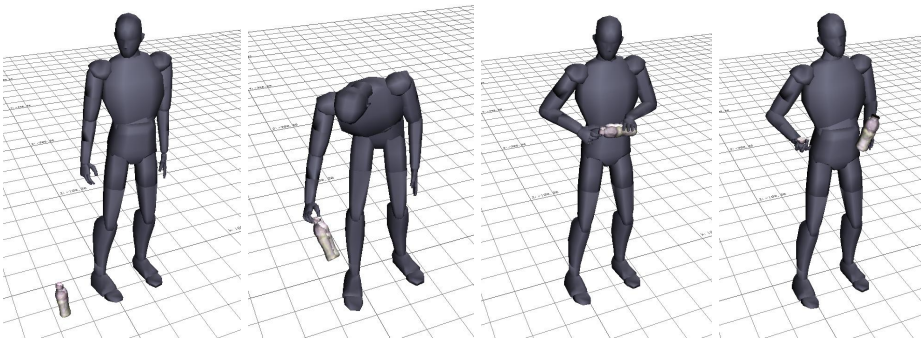
The object motion is reconstructed by parenting the relevant marker's position and orientation using the offset previously calculated. In the way, the object follows the change in position and orientation of the marker.

Table 1. Polhemus Liberty Properties

Update Rate	240 Hz
Optimal Positional Accuracy	0.71mm
Optimal Orientation Accuracy	0.15 deg RMS
Latency	3.5ms
Number of Sensors	16
Degrees of Freedom	6

4 Experimental Results

The capture-session is held in an open space with less interference from conductive material. In order to reduce the influence of the electric cables and conductive material underground, we setup a capture region at 1 meter above the ground using wooden platforms. Next, a calibration session to compensate the magnetic field distortion in our capture space is conducted. This is done by constructing a 'marker pole', which consists of five markers at regular intervals and known orientation. We position this at points on a 5x5 grid in the capture space, and sample the magnetic readings. These can be linearly interpolated to give a continuous space for the markers to exist in. It is likely that the deformation of the space is not linear, however this gives us a good approximation which is better than without calibration and is fast to implement and calculate.



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

Fig. 6. An example motion (bottle cap removal)

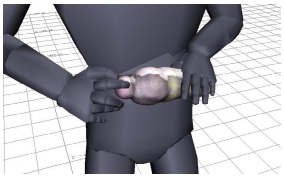
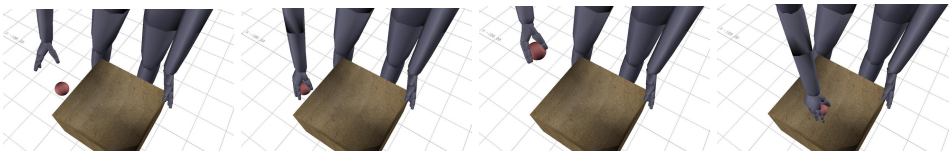
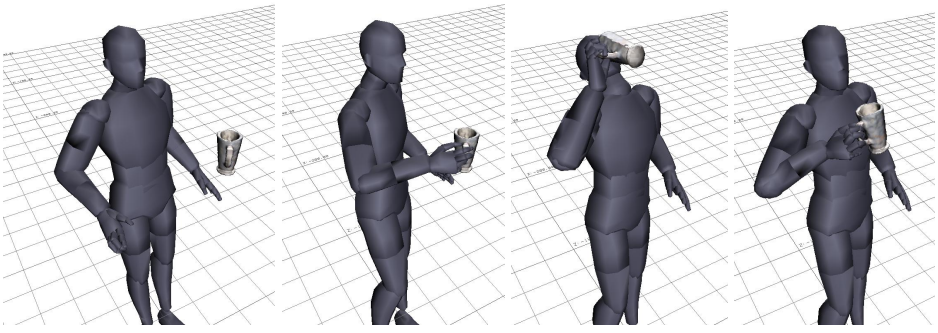


Fig. 7. Close-up of example motion. These types of close interaction must be accurately captured to prevent serious penetration issues.



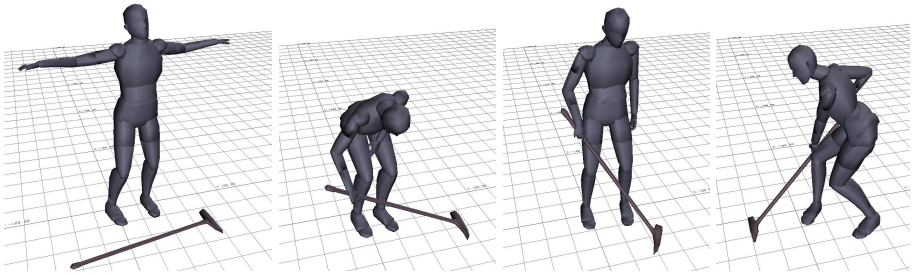
(a) (b) (c) (d)

Fig. 8. Captured motion of putting a ball into a box



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

Fig. 9. Reconstructed motion of drinking from a cup



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

Fig. 10. An example captured motion of sweeping the floor

We captured 50 different close interaction motions, with 12 different objects. These motions include using a screwdriver, picking up and putting on a hat, drawing using a pen, sweeping using a broom, drinking from different shapes of cups, and from a bottle after unscrewing the cap. We first manually estimated the body geometry and morphology using Motionbuilder, and fit the object and character to the markers, using the built-in IK for the character when necessary. These takes are available from our website in FBX format. Snapshots of some of the examples are shown in Figure 6-10. For example videos, please see the project website[3].

5 Discussions and Future Work

In this paper, we have proposed a framework to capture the human movements of manipulating objects by jointly using a magnetic motion capture system and a Kinect sensor. We have successfully captured movements such as manipulating daily objects. The data is publicly available from our website.

Comparison with Glove-Based Frameworks: We tested glove-based methods as an alternative to the magnetic capture system, but we found them to have drawbacks. The two glove-based motion capture systems we tested were resistance-based, in that the movement of the hand affected electrically conductive sections of the glove, changing the resistance which was then measured. This measurement infers some information about the joint state of the hand. This requires calibration on each use, as the location and size of the hand in the glove greatly affects how the sensors are affected. This method is in contrast to optical or magnetic sensors, as it gives you the local transformation of the hand joints, greatly relying on an accurate morphology and calibration to ensure the fingertips are in the correct place, rather than the absolute position and rotation you get from the magnetic sensors. As motion of interaction with the environment rely on the fingertips for interaction often, and incorrect positioning of them can cause penetrations, this is a serious problem for capturing close interactions. Not only this,

[3] `http://homepages.inf.ed.ac.uk/s0569500/InteractionDatabase/interactiondb.html`

but a further motion capture system would be required for the capture of the object's motion, and for computing the wrist's global transformation. We deemed the magnetic capture system superior in this case, as the markers give us the absolute position and orientation of each end effector and the objects, allowing us to reconstruct the scene.

Comparison with Optical-Based Frameworks: The magnetic system also appeared to be superior to an optical marker approach. We captured some simple grasping motions with optical markers, grabbing a cylinder for the side in an open environment. Even in this simple case, occlusions occurred. When markers passed close to each other, there was also the possibility of marker confusion, as the fingertips are too small to place a unique set of markers to define a rigid body. These two problems meant that many takes of captured data required interpolation between optical markers and manual cleanup of the data. The advantage the optical system has however, is that the markers are small and wireless, meaning motions such as throwing a ball could be captured successfully.

In the future, we plan to capture the movements of the body and the object not only by the magnetic trackers but also with the Kinect system. Such an approach will increase the precision of the object tracking. We also intend to rewrite the IK system in order to bias the solution towards more precise finger positions, at the expense of the wrist and other limbs precision, as the fingertips are the most important body part for the interactions we capture.

Acknowledgements. We thank the anonymous reviewers for their constructive comments. This work is partially supported by grants from EPSRC (EP/H012338/1) and EU FP7 TOMSY.

References

1. Burdea, G.: Virtual reality systems and applications (short course). In: Electro 1993 International Conference (1993)
2. Krieg, J.: Motion tracking: Polhemus technology. Virtual Reality Systems 1(1), 32–36 (1993)
3. Blood, E.: Device for quantitatively measuring the relative position and orientation of two bodies in the presence of metals utilizing direct current magnetic fields. U.S. Patent 4,849,692 (July 18, 1989)
4. Ascension: Flock of birds real-time motion tracker. Company Brochure, Ascension Technology Co., Burlington, VT (1998)
5. Kry, P.G., Pai, D.K.: Interaction capture and synthesis. ACM Trans. Graph. 25(3), 872–880 (2006)
6. Liu, C.K.: Dextrous manipulation from a grasping pose. ACM Trans. Graph. 28(3) (2009)
7. Ho, E.S.L., Komura, T., Tai, C.L.: Spatial relationship preserving character motion adaptation. ACM Trans. Graph. 29(4) (2010)
8. Ye, Y., Liu, C.K.: Synthesis of detailed hand manipulations using contact sampling. ACM Trans. Graph. (SIGGRAPH 2012) 31(4) (2012)
9. Hamer, H., Gall, J., Urtasun, R., van Gool, L.: Data-driven animation of hand-object interactions. In: IEEE Conference on Automatic Face and Gesture Recognition, pp. 360–367 (2011)

10. Mitobe, K., Kaiga, T., Yukawa, T., Miura, T., Tamamoto, H., Rodgers, A., Yoshimura, N.: Development of a motion capture system for a hand using a magnetic three dimensional position sensor. In: ACM SIGGRAPH 2006 Research Posters, SIGGRAPH 2006. ACM, New York (2006)
11. Munoz-Salinas, R.: Aruco: a minimal library for augmented reality applications based on opencv (2012), http://www.uco.es/investiga/grupos/ava/node/26
12. Burrus, N.: Rgbdemo (June 2012), http://labs.manctl.com/rgbdemo/
13. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Eurographics Symposium on Geometry Processing (2006)
14. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3212–3217 (May 2009)