Masterpiece: Physical Interaction and 3D Content-Based Search in VR Applications

Konstantinos Moustakas and Michael G. Strintzis
Aristotle University of Thessaloniki

Dimitrios Tzovaras
Informatics and Telematics Institute

Sebastien Carbini, Olivier Bernier, and Jean Emmanuel Viallet
France Telecom R&D

Stephan Raidt
INP-Grenoble

Matei Mancas
Faculty of Engineering, Mons

Mariella Dimiccoli
Technical University of Catalonia

Enver Yagci and Serdar Balci
Bogazici University

Eloisa Ibanez Leon
Technical University of Madrid

Virtual reality interfaces can immerse users into virtual environments from an impressive array of application fields, including entertainment, education, design, and navigation. However, history teaches us that no matter how rich the content is from these applications, it remains out of reach for users without a physical way to interact with it. Multimodal interfaces give users a way to interact with the virtual environment (VE) using more than one complementary modality.

Masterpiece (which is short for Multimodal Authoring Tool with SIMILAR Technologies from European Research utilizing a Physical Interface in an Enhanced Collaborative Environment) is a platform for a multimodal natural interface. We integrated Masterpiece into a new authoring tool for designers and engineers that uses 3D search capabilities to access original database content, supporting natural human–computer interaction.

Masterpiece increases the user's immersion with a physical interface that's easier to use than a traditional mouse and keyboard. The user can generate and manipulate simple 3D objects with a sketch-based approach that integrates a multimodal gesture–speech interface. They can then assemble their 3D parts into more complex objects. Moreover, the user can access a database's original 3D content using a 3D search engine.[1–3] Using the rough sketch they created, users can search for similar 3D content in the database.

Application framework

A prospective Masterpiece user might want to generate a 3D virtual environment by accessing models in a database using a sketch-based 3D content-based search. Using Masterpiece, the user would be able to

- manipulate 3D objects in a 3D environment—including translation, rotation, scaling, deformation, and so forth;
- assemble mechanical objects from their spare parts;
- import 3D primitive objects using sketches;
- manipulate and deform the primitive objects to generate more complex structures; and
- perform content-based 3D search using as a query either a scene object or a model the user designed by combining primitive objects.

Unfortunately, it would be difficult to operate such a system with standard keyboard-mouse input devices. The problem is that it's not easy to reproduce 3D actions using 2D input devices. (See the "Multimodal Interface Research" sidebar for details on other related work in this area.)

With Masterpiece, we let designers physically interact with the application to overcome the need to transition between the 2D input devices and the 3D VE. In particular, our multimodal interface consists of the following modules:

- speech recognition for specific commands,
- gesture recognition for efficiently handling 3D objects using 3D hand motions,
- recognition of 2D sketches,
- primitive model import and manipulation using gestures, and
- deformation of objects using gestures.

Speech recognition helps Masterpiece recognize

1070-986X/06/$20.00 © 2006 IEEE

Published by the IEEE Computer Society

Multimodal Interface Research

Since Richard Bolt's pioneering "Put-That-There" system,[1] researchers have known that among the gestures humans naturally use to communicate, pointing gestures associated with speech recognition lead to more powerful and natural human–machine interfaces. In a related "Wizard of Oz" experiment,[2] subjects were asked to manipulate 3D objects in a virtual game environment. This work shows that, if given the opportunity, 60 percent of subjects would use multimodal interaction more than 60 percent of the time to interact with the game.

Ed Kaiser et al.[3] showed that a user can manipulate and rotate objects with one-hand gestures and speech using magnetic sensors. Latoschik's multimodal system[4] allowed bi-manual manipulation of virtual objects thanks to two data gloves. Still, using instruments constrains the interaction and often tethers the user to the machine.

In Nils Krahnstoever's et al.'s research,[5] the user is free to use more natural gestures thanks to a vision-based recognition system. One-hand gestures are recognized using hidden Markov models (HMM). However, due to the statistical method they use for continuous recognition, the speech- and gesture-recognition systems typically provide results only after a one-second delay. Moreover, other experiments[6] have shown that speakers often use both hands during a descriptive monologue. For example, in Krahnstoever's experiement,[5] the use of a single camera leads to simple assumptions on background, and the cursor's displacement is linked to the hand's 2D position only.

Our system allows unconstrained and natural 3D gestures of both hands by using stereo camera-based gesture recognition. We make no assumptions about the background, which can contain other moving persons without disturbing the tracker. Speech recognition delivers results with an acceptable lag of 240 milliseconds. We also use iconic bimanual gestures for rotation and resizing as well as deictic gestures for selecting and moving objects to create an intuitive interface that users can pick up without any previous training.

References

1. R.A. Bolt, "Put-That-There: Voice and Gesture at the Graphics Interface," *Proc. 7th Ann. Conf. Computer Graphics and Interactive Techniques*, ACM Press, 1980, pp. 262-270.
2. A. Corradini and P.R. Cohen, "On the Relationships Among Speech, Gestures, and Object Manipulation in Virtual Environments: Initial Evidence," *Proc. Int'l CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*, 2002, pp. 52-61.
3. E. Kaiser et al., "Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality," *Proc. Int'l Conf. Multimodal Interfaces* (ICMI), IEEE CS Press, 2003, pp. 12-19.
4. M.E. Latoschik, "A User Interface Framework for Multimodal VR Interactions," *Proc. Int'l Conf. Multimodal Interfaces* (ICMI), IEEE CS Press, 2005, pp. 76-83.
5. N. Krahnstoever et al., "A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays," *Proc. Int'l Conf. Multimodal Interfaces* (ICMI), IEEE CS Press, 2002, pp. 349-354.
6. J. Eisenstein and R. Davis, "Natural Gesture in Descriptive Monologues," *Proc. ACM Symp. User Interface Software and Technology* (UIST 2003), ACM Press, 2003, pp. 69-70.

Table 1. Speech- and gesture-controlled actions.

Speech Commands	Actions Performed	Gesture Commands	Actions Performed
Search	Use the selected object as query and search for similar content	Pointer control	The 3D pointer follows the user's hand motion
Select group	Initiate grouping the primitives and call the selection command for each primitive	Selection	Point at the object to be selected
Retrieve	Retrieve the objects from the database starting with the most similar	Translation	Move the hand until the object reaches the target 3D position
Next	Retrieve the next most similar object	Rotation	Rotate the hands like grabbing and rotating a sphere
Delete	Delete selected object	Scaling	Increase/decrease the distance between the two hands
Clone	Clone selected object		
Sketch	Start freehand sketching		
Stop action	Stop currently performed action		

specific commands that signify the beginning of an action. Typical commands include "rotate," "move," "scale," "search for similar content," or "clone object." After defining the action, we use gesture recognition to perform the task—for example, users rotate an object by rotating their hands, the system moves an object to the 3D area the user points to, and so on. Finally, a user terminates a performed action giving the command "stop action." Table 1 illustrates actions controlled by gesture and speech.

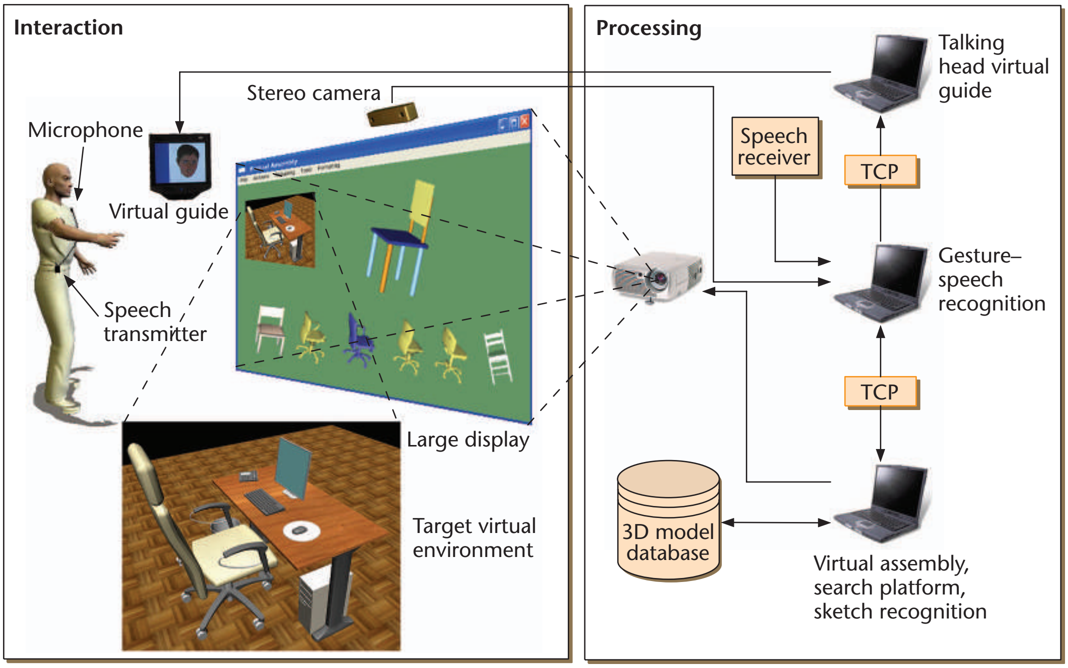
Interaction
Processing
Stereo camera
Microphone
Virtual guide
Speech transmitter
Large display
Target virtual environment
Talking head virtual guide
Speech receiver
TCP
Gesture–speech recognition
TCP
3D model database
Virtual assembly, search platform, sketch recognition

Figure 1. The Masterpiece platform's architecture. The interaction module includes mainly the interaction hardware, including the large display, the stereoscopic camera, and microphone. The processing part is responsible for handling all events generated by the user's actions such as speech commands and gestures. (TCP stands for Transmission Control Protocol.)

Figure 1 illustrates the platform's architecture. The system consists of interaction and processing parts, which are distinct but interrelated. The interaction hardware components include a stereoscopic camera, microphone and speech transmitter/receiver, and a large display.

Masterpiece modules

Figure 2 illustrates the architecture of the Masterpiece physical interface. The two input modalities—gesture and speech—drive the system, while the distributed processing module performs all the processing. The system projects output onto the large display and provides the user with feedback through a virtual guide. This article describes the interface's most important features.

Authoring tool

Masterpiece's core application is a physical-interaction-based VR tool for authoring VEs. Besides the gesture–speech interface, the core application supports interaction with haptic devices and stereoscopic displays. Moreover, it uses several smart modules, such as the snap and collision agent, to carry out the corresponding complex tasks. The authoring tool integrates a

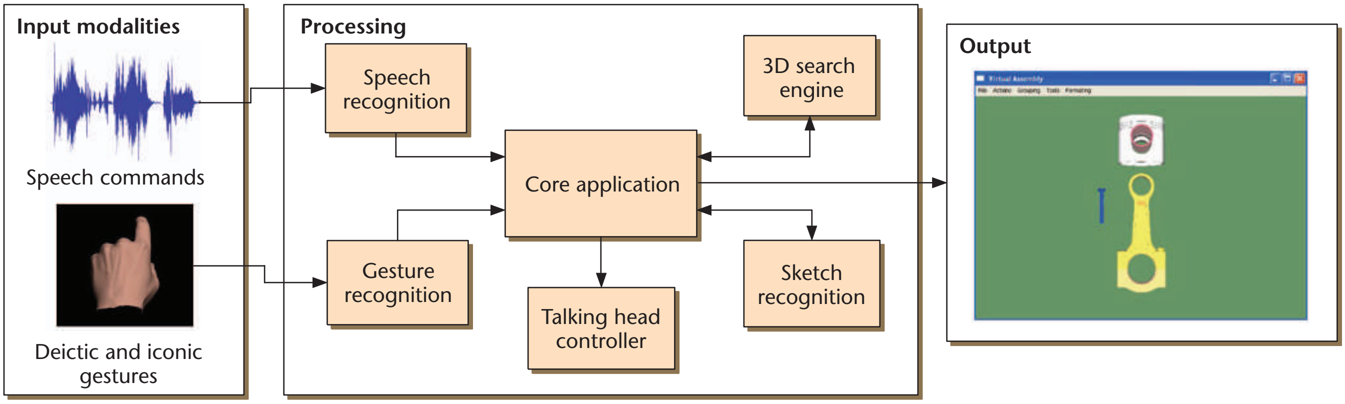

Figure 2. The Masterpiece processing modular architecture. The user's input to the system includes speech commands and deictic and iconic gestures. The core application recognizes the input actions and provides verification feedback to the user through a virtual guide. The sketch-recognition and 3D search modules are used to design simplified versions of a target object and to search for similar content, respectively. At every time step, the system displays the result of the user's actions on the large screen.

3D Content-Based Search

Three-dimensional content-based search and retrieval is a challenging research area with numerous application branches such as recognition in computer vision and mechanical engineering as well as content-based search in e-commerce and edutainment applications. These application fields will expand in the near future because the 3D-model databases are growing rapidly due to recently improved scanning hardware and modeling software. Furthermore, the increased processing power of the latest graphic cards enables fast processing and visualization of complex 3D shape representations, even on standard desktop computers.

State-of-the-art approaches usually extract geometric characteristics (descriptors) of the 3D objects and subsequently compare these descriptors to measure the similarity between objects.[1] The descriptors can be coefficients of 3D Fourier-based expansions or more complicated and sophisticated measures with specific attributes such as rotation invariance, independence from the sampling density, and robustness to noise. The 3D search engine we integrated into Masterpiece produces rotation-invariant descriptors using the spherical trace transform.[1]

Most approaches need a query model based on objects' similarity. This requirement is restrictive because of cases when users know what they want to search for but no query object is available.

The following scenario illustrates a realistic use case: The user of a virtual assembly application is trying to assemble an engine with spare parts. He inserts some rigid parts into the virtual scene and places them in the correct position. At one point, he needs to find a piston. Normally, he would have to manually search in the database to find the piston. However, it would be faster and easier if the user could sketch[2] an object similar to a piston and search for similar content in the database.

References

1. P. Daras et al., "3D Model Search and Retrieval Based on the Spherical Trace Transform," *Proc. IEEE Int'l Workshop on Multimedia Signal Processing* (MMSP 04), IEEE Press, 2004, pp. 335-338.
2. M. Oliveira et al., "Modeling Solids and Surfaces with Sketches: An Empirical Evaluation," *Proc. Winter School of Computer Graphics* (WSCG 2001), 2001, pp. 28-31.

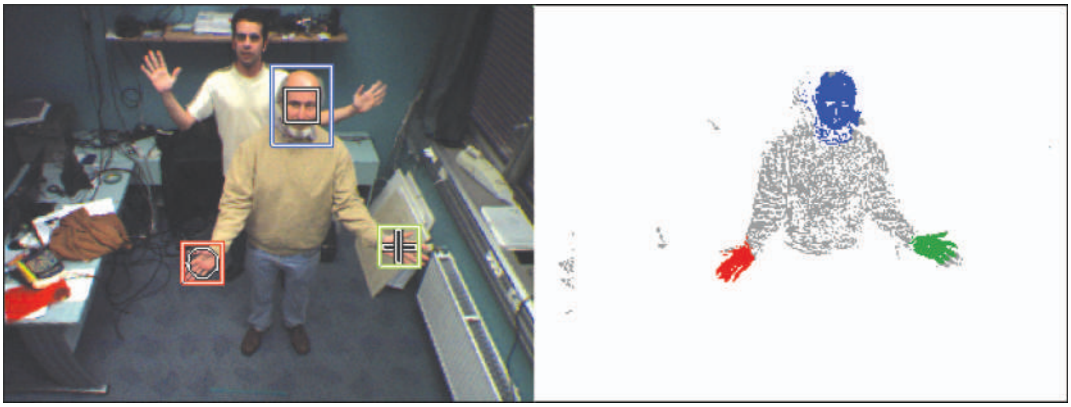

Figure 3. (a) In the camera image, the blue rectangle indicates the head, the red circle indicates hand one, and the green cross indicates hand two. (b) The system assigns observations to one of the four models depending on their probabilities. In this image, the blue is the head, the red blob is hand one, the green blob is hand two, the gray area is discarded, and the white pixels are ignored in Expectation Maximization.

3D search and retrieval engine, which can search for 3D objects in a database using another 3D object or a model the user designed as a query. Based on that query, the tool retrieves the most similar 3D content in terms of a distance metric between their descriptor vectors. (See the "3D Content-Based Search" sidebar for more details about this application area.)

Finally, to make the interface more natural, we included a talking-head agent to provide feedback about the task status. It provides audio–visual information about the recognized voice commands and the head and hand tracking. The virtual guide we integrated into Masterpiece greets new users. When the user is lost or localized again by the system, the clone provides the appropriate feedback.

Gesture recognition

Most people instinctively use the eye-tip of the finger line to point at a target. We use this convention in the Masterpiece framework to estimate the user's pointing direction. More precisely, we estimate the pointing area by projecting the head–hand axis to the screen.

The system detects and tracks head and hands[4] (see Figure 3a) using the 3D data that a stereo camera provides. The user's face is automatically detected by a neural network—see related work[5] for a full description. Hands are

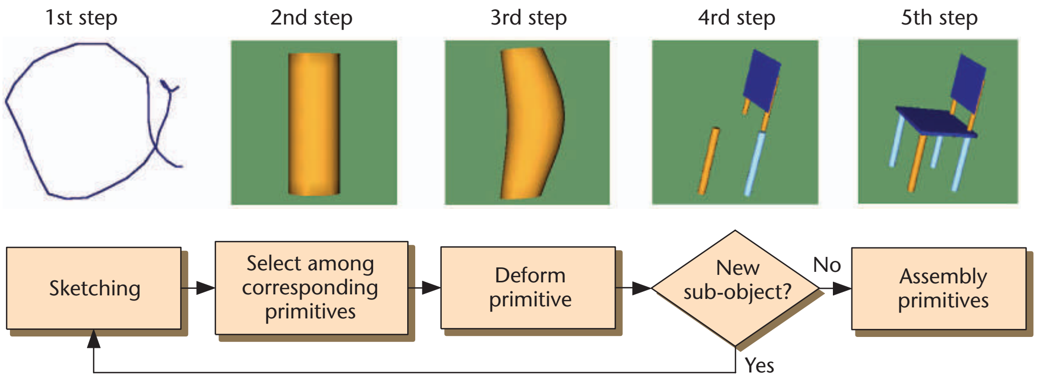

Figure 4. The query model generation procedure. Initially, the users sketch the 2D contour of the desired primitive. Next, they choose among the corresponding primitives (for example, a circle corresponds to both a sphere and a cylinder) using speech. In the third step, the primitive is deformed. If an extra primitive is necessary, the procedure is repeated; otherwise, the primitives are assembled to form the query object.

detected as a skin-colored moving area in front of the user.

The first detected hand is tagged as the "pointing hand" and the second as the "control hand." Hence, the system works for both right- and left-handed users. It doesn't explicitly differentiate the right hand from the left because the predominant hand is generally used to point.

Once detected, we track the head and hands simultaneously until tracking failure, at which point the algorithm automatically retriggers detection for the lost part. The tracking process represents each new image with a statistical model using the Expectation Maximization algorithm.[4] The statistical model consists of a color histogram and a 3D spatial Gaussian function for each tracked body part (see Figure 3b).

During runtime, speech commands trigger gesture recognition. For example, the "scale" command lets users resize objects proportionally to the distance between their hands. Table 1 lists the speech- and gesture-controlled actions.

Speech recognition

To recognize speech commands, we linearly sample the speech signal at 8 kHz in 16 bits. Next, we compute Mel frequency cepstrum coefficients (MFCC). The recognition system uses the frame energy, 8 cepstral coefficients, and an estimation of the speech signal's first- and second-order derivatives.

In the decoding system, we use hidden Markov models and a grammar to describe the sentence syntax the system recognizes. The system recognizes a 50-word vocabulary. Depending on the context, each word is obtained by phonetic unit concatenations that are allophones.[6] The system finally outputs the *n*-best results.

Query model generation, sketch recognition

This module helps users efficiently design an approximation of an object they want to search for in the database. The 3D primitive objects vocabulary that Masterpiece supports consists of seven objects: sphere, cylinder, cone, parallelepiped, pyramid (with parallelepiped base), pyramid (with triangular base), and prisma. The vocabulary might initially seem limited. However, in the tests we performed, we found that users used cylinders, spheres, and parallelepiped almost exclusively for all the objects they built.

The user must follow five steps to build an object:

1. Sketch the 2D contour of the desired primitive object.

2. Choose among the corresponding 3D shapes using speech (for example, for a circle, choose a sphere, cylinder, or cone) and define its height using gesture. The size of a cylinder's base is acquired from the sketch but not its height.

3. Deform the surface, if desired.

4. If a new primitive is needed to form a more complex shape, go to step 1. Otherwise, proceed to step 5.

5. Assemble primitives to form the final shape.

Figure 4 illustrates these steps.

To accomplish 2D sketch recognition, we must preprocess the acquired trajectories. They aren't uniformly sampled because it's difficult to keep a constant sketching speed and they exhibit points that don't belong to the intended shape at the beginning and end.

Therefore, we apply a filter before recognition, which resamples the points in the contour uniformly to keep the distance between two succes-

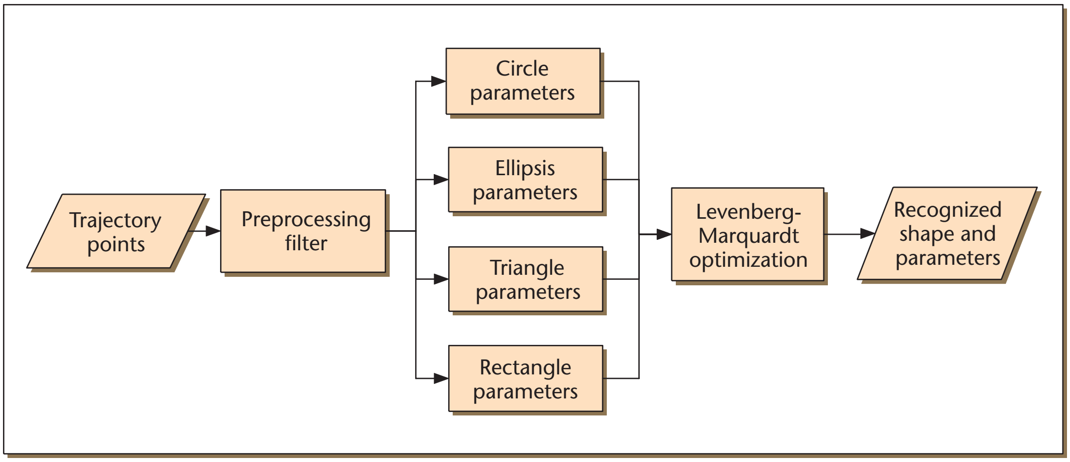

Figure 5. Flow chart of the sketch-recognition algorithm. The trajectory points of the user's sketch comprise the system input. They are initially filtered to provide equally spaced trajectory points. Next, they are fitted to four primitive shapes and, finally, the trajectory is classified to the shape that produces the minimum mean squared error.

sive points constant. This prevents the undesired effect of the point clustering, which we see mainly at shape corners. Moreover, the filter discards the undesired points at the contour's beginning and end by detecting the possible cross section and keeping only the points that form a closed trajectory.

To recognize the contours' 2D shape after filtering, we initially fit the three possible parameterized geometrical shapes (such as circle, ellipsis, triangle, or rectangle) to the obtained trajectory using the Levenberg-Marquardt method for unconstrained nonlinear least-squares minimization. This is an iterative technique that finds a local minimum of a multivariate function that's expressed as the sum of squares of nonlinear functions. The shapes' parameter space includes parameters for object position, rotation, scaling, and so on. We estimate initial values from the data's statistical properties—for example, the data points' mean, maximum, and minimum. After finding the optimal solution, we compare the least-squares errors (or the sum of the squared distances of each point from the shape) and classify the object to the category with the minimum least-squares error. Figure 5 illustrates the algorithm's flow chart.

Figure 6 illustrates four cases of shape recognition, where the recognized shape is drawn in a red dashed line. To address shape irregularities and gesture-tracking noise, the algorithm resamples the gesture trajectories and discards the noise points at the beginning and end of the sketch.

After recognizing the sketch's 2D shape, we have already defined the object's projection to a plane and a sample object appears on the screen. Until the user utters the "OK" command, the object scales nonuniformly alongside the perpendicular direction to the sketch plane by following the user's hand—that is, it becomes larger or smaller when the user moves her hand up or down, respectively.

In the next step, the user can deform the generated primitive object. We included deformation so the user could interactively affect the 3D object's triangulated model.

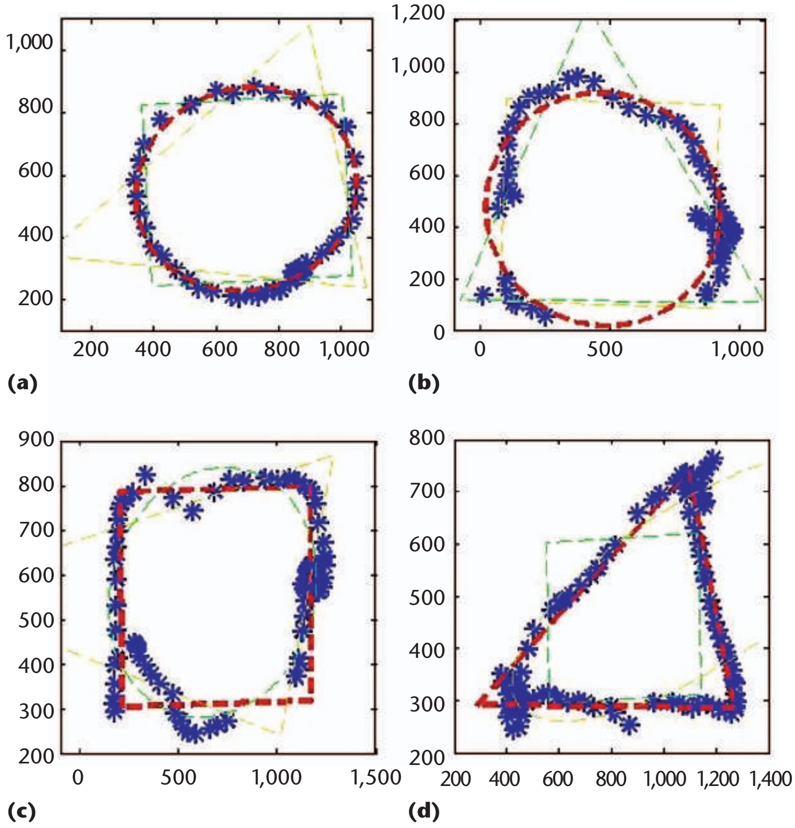

Figure 6. The user draws the gesture trajectory with blue star markers, and the red dashed line shows the detected shape: (a) a well-designed circular contour, (b) an incomplete circular contour, (c) a noisy rectangular shape, and (d) a triangle with tails at the beginning and end. (The graph measurements are in millimeters.)

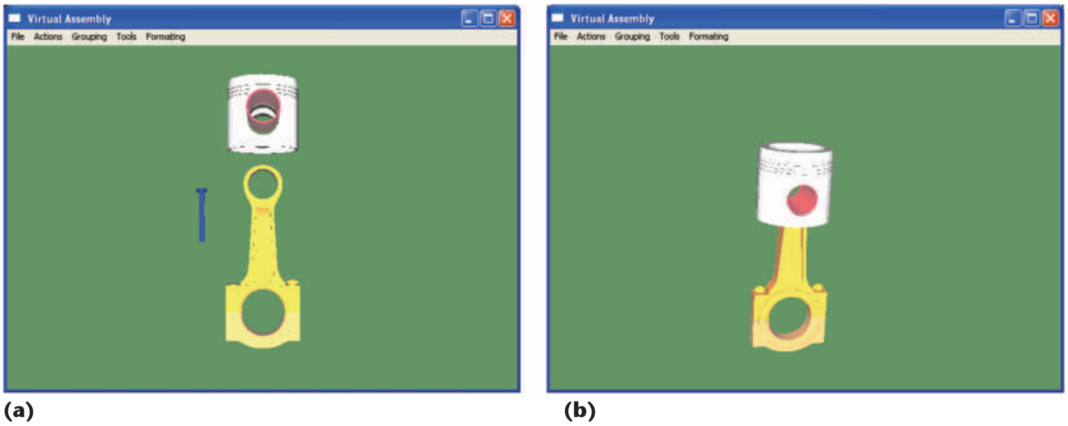
Virtual Assembly
File Actions Grouping Tools Formatting
(a)
Virtual Assembly
File Actions Grouping Tools Formatting
(b)

Figure 7. Gesture–speech-based assembly of a piston (a) during and (b) after the assembly procedure.

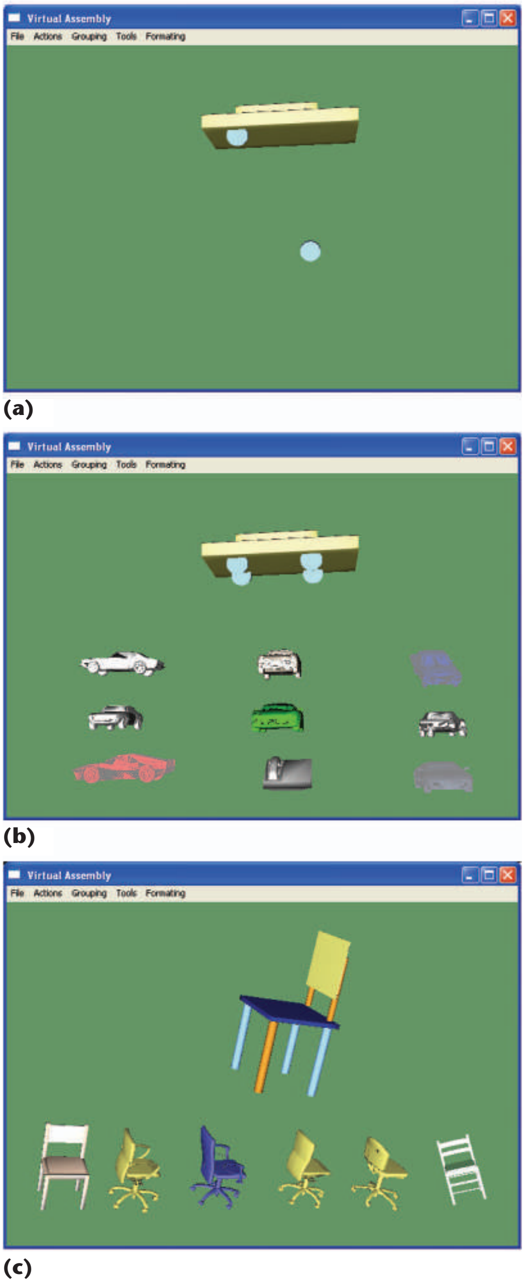
Virtual Assembly
File Actions Grouping Tools Formatting
(a)
Virtual Assembly
File Actions Grouping Tools Formatting
(b)
Virtual Assembly
File Actions Grouping Tools Formatting
(c)

Figure 8. With Masterpiece, the user attempts to (a) sketch a car and (b) retrieve similar objects using the sketched car query. (c) The user can also retrieve similar objects using a sketched chair as a query.

Initially, a number of control points are automatically defined on the object's surface. To deform the mesh, the user moves the control points, and the translation is propagated to the object elements (such as vertices or triangles) in a decreasing manner with respect to their geodesic distance from the control point. The points that are far away from the control point are translated less than closer points. Unlike the Euclidean distance, the geodesic distance is the minimum distance between two points on a surface—that is, the minimum length of the surface curve that connects them.

After the primitive objects are imported into the screen and processed accordingly (scaled, rotated, and/or deformed), they are translated to the desired position so that the user can build the targeting object.

Application demos

We've used two scenarios to test Masterpiece: assembling a piston and designing VEs using content-based 3D search and sketch-based query models.

In the first scenario, the user had to assemble a piston using the developed gesture–speech interface. Specific speech commands we described earlier were used to select an action and then the objects were manipulated using gestures. Figure 7 illustrates the assembly procedure's two phases.

The aim of the second scenario was for users to design 3D VEs. Initially, users had to draw primitive objects using sketches and then assemble them into a more complex sketch. They then had to query the 3D content-based search engine for similar content.

Figures 8a and 8b show two snapshots using this procedure to design a car and retrieve similar content. Only one of the retrieved objects in Figure 8b isn't a car, but all the objects have a similar geometrical shape. Figure 8c shows the objects the system retrieved when the user attempted to sketch a chair. Finally, Figure 9 illustrates the result of designing a living room and an office using Masterpiece's sketching utility.

Evaluation

We used many scenarios to evaluate our sketch-based 3D search platform in addition to the two we've described here. We compared all our results with a VR interface that uses the CyberGrasp haptic glove for interaction and a simple interface that uses a 2D mouse that can be operated in the 3D space using a gyroscope (see Figure 10). Our aim was to test our approach against different interfaces with respect to several parameters, as Table 2 illustrates.

The Masterpiece interface is superior in terms of user immersion, usability, 3D manipulation efficiency, and device intrusiveness. The haptic glove interface seems to be more robust, but the

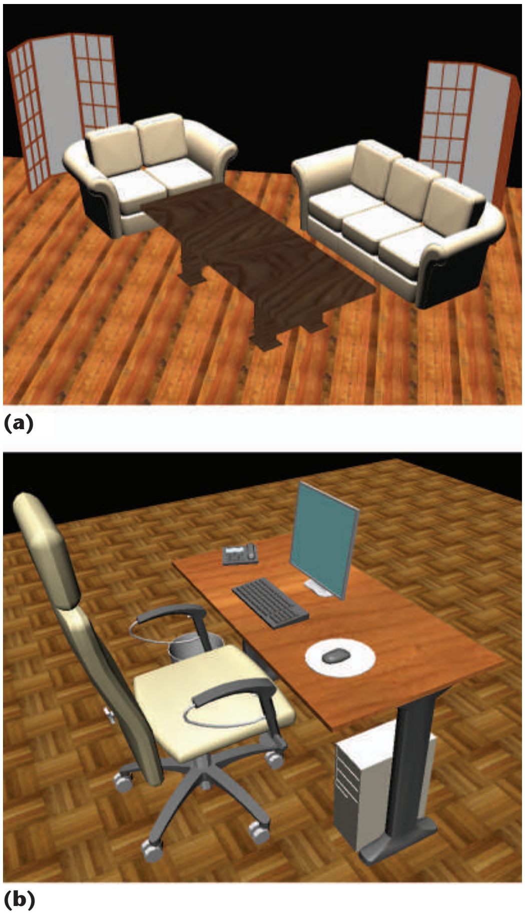

Figure 9. Users can design more complex sketches such as (a) a living room or (b) an office using Masterpiece.

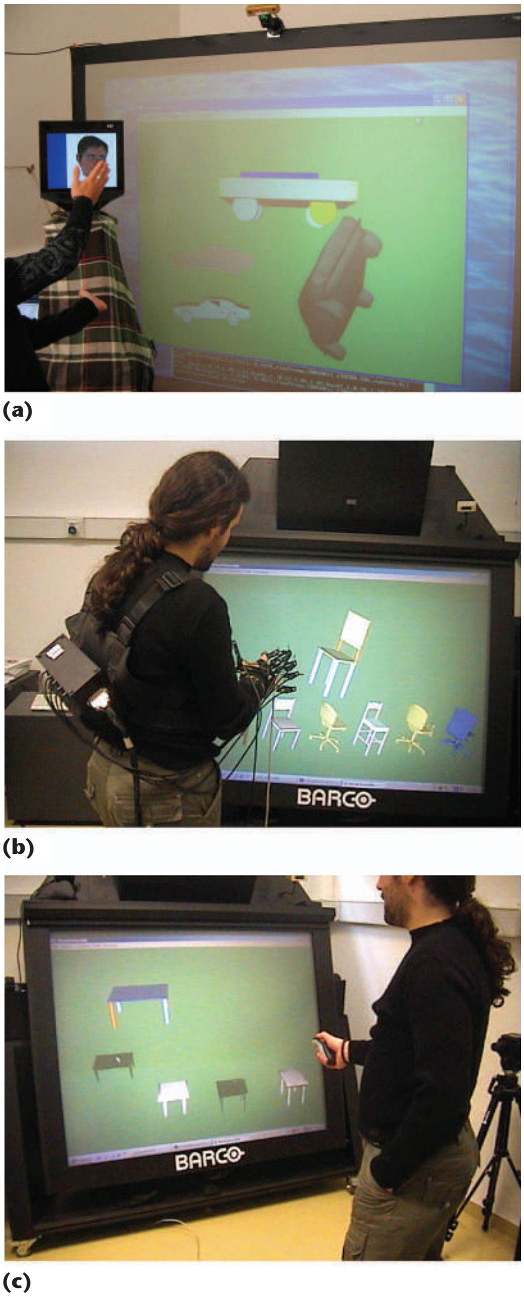

Figure 10. User attempting to perform similar operations using (a) the Masterpiece interface, (b) a haptic glove interface, and (c) an air mouse interface.

Table 2. Interface comparison.

Interface Attributes	Masterpiece	Haptic VR	Air Mouse
User immersion	Very high	High	Very low
Usability	Very high	Very high	Moderate
3D manipulation efficiency	Very high	Very high	Very low
Mobility	Very low	Very low	Very high
Robustness	High	Very high	Very high
Computational efficiency	Moderate	Moderate	Very high
Device intrusiveness	Very low	High	Low
Cost	Moderate	High	Very low

user must wear relatively heavy equipment, and the gear requires some initialization time prior to starting the interaction. Finally, the air mouse interface is a cheap solution, but it's inferior to the others, especially in terms of easily manipulating objects in a 3D environment.

Our system evaluation also illustrates that the gesture–speech interface is a feature that's nice to have, but it isn't absolutely necessary. On the contrary, users consider the sketch-based query-generation module important because it significantly reduces the time needed to perform 3D content-based search if no query model is available.

Conclusion

Our experiments and tests show that Masterpiece is user friendly and dramatically increases the user's immersion in the application. In the near future, we plan to extend it by adding a 3D surface sketching capability so users can more effectively design 3D objects. Moreover, we will explore the possibility of navigating in game-like VR environments using Masterpiece's interface augmented with feet recognition and tracking.

We believe realistic physical interfaces are the near future in the realm of human–computer interaction. The time when VR applications will be totally guided using unobtrusive interfaces isn't far away. **MM**

Acknowledgments

We conducted most of this work during the Enterface 2005 Summer Workshop on Multimodal Interfaces (http://www.enterface.net),[7] which is supported by the EU-funded SIMILAR Network of Excellence.

References

1. P. Daras et al., "3D Model Search and Retrieval Based on the Spherical Trace Transform," *Proc. IEEE Int'l Workshop on Multimedia Signal Processing* (MMSP 04), IEEE Press, 2004, pp. 335-338.
2. I. Kolonias et al., "Fast Content-Based Search of VRML Models Based on Shape Descriptors," *IEEE Trans. Multimedia*, vol. 7, no. 1, Feb. 2005, pp. 114-126.
3. D.V. Vranic and D. Saupe, "Description of 3D-Shape Using a Complex Function on the Sphere," *Proc. IEEE Int'l Conf. Multimedia and Expo* (ICME), IEEE CS Press, 2002, pp. 177-180.
4. S. Carbini, J.E. Viallet and L. Delphin-Poulat, "Context Dependent Interpretation of Multimodal Speech-Pointing Gesture Interface," *Proc. Int'l Conf. Multimodal Interfaces* (ICMI), IEEE CS Press, 2005, pp. 1-4.
5. R. Feraud et al., "A Fast and Accurate Face Detector Based on Neural Networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, Jan. 2001, pp. 42-53.
6. K. Bartkova and D. Jouvet, "Modelization of Allophones in a Speech Recognition System," *Proc. Int'l Congress of Phonetic Science* (ICPhS), 1991, pp. 474-477.
7. M.E. Latoschik, "A User Interface Framework for Multimodal VR Interactions," *Proc. Int'l Conf. Multimodal Interfaces* (ICMI), IEEE CS Press, 2005, pp. 76-83.

Readers may contact Konstantinos Moustakas at moustak@iti.gr.

Contact Multimedia at Work department editor Qibin Sun at qibin@i2ra.a-star.edu.sg.