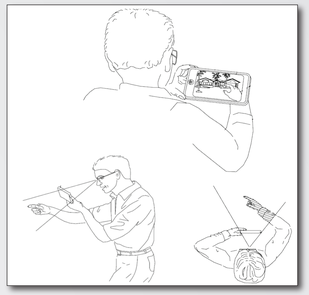
Range Camera for Simple Behind Display Interaction

DRIVE



DRIVE is an interaction method that allows a user to manipulate virtual content by reaching behind a display, unlike works that use front volume or front, side, and back surfaces. Together with face tracking, it creates an illusion that the user's hand is co-located with virtual volumetric content.

Range Camera for Interaction

New interaction methods make a human body the actual controller. Range cameras measure per-pixel distance and make it easier to locate and track body parts. However, one needs to deal with problems like self occlusions, limbs identification, discrimination between multiple people, as well as noise and low resolution of the sensor.

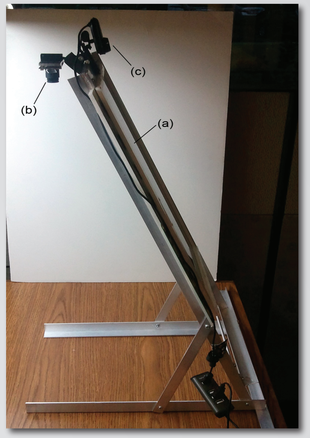
DRIVE Enabled Simplifications

DRIVE allows to streamline the implementation:

1. Instead of gesture recognition, an interaction happens via collision detection of CG content with low polygon aproximation of the interacting body part.

2. Since the camera sees only the manipulating hand of the user in front of a mostly static background, there is no need to identify and track body parts.

Hardware and Software Setup



Hardware platform:

a) color translucent LCD with 10% trasnparency;

b) time-of-flight range camera by PMDTech;

c) webcam for motion parallax depth cues.

Software:

a) 3D graphics engine OGRE;

b) face tracker faceAPI;

c) hand tracking algorithm.

d) physics engine for collision and gravity effects;

e) anaglyph stereo rendering, so that virtual objects appear behind the transparent display;

f) view-dependant rendering, based on face position for motion parallax depth cue.

Hand Localization Algorithm

1. Filter sensor data, get point cloud

2. Aproximate point cloud with oriented box

3. Construct a proxy object for physics engine

1. Filter Sensor Data

1.1. Weak - discard pixels with small amplitude ~ 70% of all pixels

1.2. Background - discard pixels failing z-test ~ 10% of all pixels

1.3. Outliers - discard pixels based on proximity ~ 3% pixels

1.3. Statistical Outlier Removal

1.3.1. Down-sample for performance reasons
1.3.2. Form point cloud
1.3.3. Find nearest neighbors for each point
1.3.4. Discard outliers based on average distance
1.3.5. Cluster points based on mutual distance
1.3.6. Keep the largest cluster

Implementation and Performance

Nearest neighbours search is done using KDD tree via ANN library. Orientation and size of an approximating box are calculated using PCA via Eigen library.

On Intel I7 CPU at 2.8 GHz with hand at 20cm from the sensor, hand detection and localization takes 22 ms, where 20ms are spent doing neighbouring search using KDD tree.

Possible Applications

- Gaming (direct interaction with game content, game characters, etc.);
- Virtual worlds (e.g., real person shaking hands with avatars);
- Telepresence (combining local and remote video and CG content, and manipulating remote content);
- CAD/CAM (direct manipulation of CG objects);
- Medical imaging (navigating and manipulating interactive 3D medical image content).

Reference

Statisstical filtering
Rusu, R. B. et al. **Towards 3D Point Cloud Based Object Maps for Household Environments**. Robotics and Autonomous Systems Journal, Special Issue on Semantic Knowledge, 2008.

See also

S.W. Kim, A. Treskunov, S. Marti. **DRIVE: Directly Reaching Into Virtual Environment With Bare Hand Manipulation Behind Mobile Display**. In: 3DUI 2011, 19-20 March, Singapore [Poster]

IAPR MVA2011

Anton Treskunov, Seung Wook Kim, Stefan Marti
Samsung Information Systems America (Samsung Electronics US R&D Center)

SAMSUNG