

B. BAISAKOV, A. AKSHABAYEV, L. NAIZABAYEVA

MANAGING A VIRTUAL OBJECT USING A 3D CAMERA DISTANCE INFORMATION

Kazakh-British Technical University, Almaty

This paper describes our experiment with the video camera which can generate RGB color output and depth map in the one time. In this work used information generated by this camera for drawing and manipulating an augmented cube defined by real marker. For visualizing 3D cube augmented reality toolkit is used. Also we invented new algorithm for finding distance between camera and virtual object using special camera depth sensors.

1. Introduction. First, we found the translation of centre point of the marker in the video frame coordinate. By this information we calculate the depth centre point of the marker, e.g. we found the distance between the camera and marker centre. This distance is used as a third – Z coordinate of the centre point. Also, we found the centre of the real hand in the video frame. Finally, by calculating according Z coordinate to this point, we conclude that it's possible to manipulate the augmented cube by real hand in 3D environment.

To take a skin area, convert RGB input image to YCbCr color model. And we can get candidates of the hand. Calculate the center of gravity. And cut out fingers using the circle which has center at the center of gravity. To initialize hand region we count under blobs. Hand rectangle will have 6 to 7 blobs. At this time we should correspond depth map pixels to RGB map. This paper includes KINECT calibration method. We can update hand region by set the threshold in the depth.

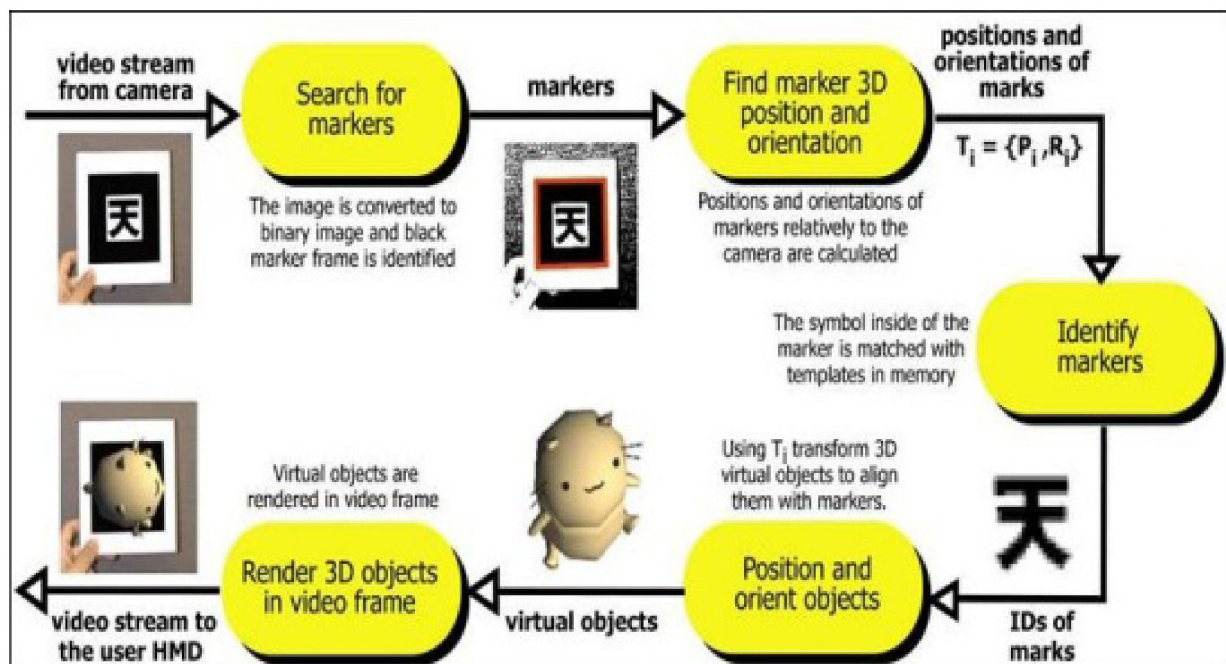


Fig. 1. TheARToolKit tracking steps taken from [2]

2. AR Objects Depth Detection

A. Reading data from sensors device. In our work we used active-sensing depth camera which has structured light, and a color camera. It generates images of 640x480 pixels 30 times a second. For accessing the generated data OpenNI (Open Natural Interaction) framework is used. The term Natural Interaction (NI) refers to a concept where Human-device interaction is based on human senses, mostly focused on hearing and vision. OpenNI is a multi-language, cross-platform framework that defines APIs

for writing applications utilizing Natural Interaction. OpenNI APIs are composed of a set of interfaces for writing NI applications. OpenNI defines Production Nodes, which are a set of components that have a productive role in the data creation process required for Natural Interaction based applications. Each production node encapsulates the functionality that relates to the generation of the specific data type [1].

B. Marker detection in the video frame. ARToolKit is a popular planar marker system for Augmented Reality and Human Computer Interaction (HCI) systems due to its available source code. The bitonal markers consist of a square black border and a pattern in the interior. Color video stream is generated by kinect and transferred to the application layer by the OpenNI Image generator [1]. The image is converted to binary image and black marker frame is identified. After successful identification of marker, the position of the camera relative to the black square is calculated. Area, centre position, lines and vertexes of the detected marker is used to drawing AR object on top of the video of the real world [2].



Fig. 2. Sample marker in frame without virtual object

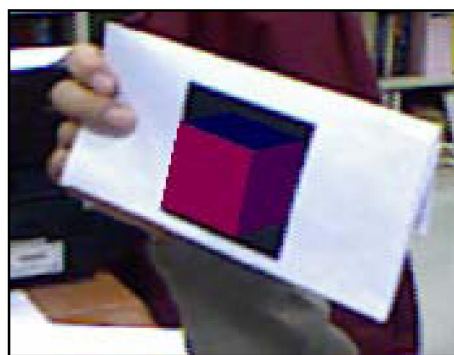


Fig. 3. Sample marker in frame with virtual cube

B. Video frame depth map generating. A depth generator is a production node that is implemented by a sensor, which takes raw sensory data from a depth sensor and outputs a depth map [1]. This data is generated by hardware devices that capture the visual and audio elements of the scene and stored in the 16-bit value. We can consider it as a one dimensional array.

C. Virtual object depth evaluating from the depth map. Let's consider detected marker centre coordinates in the video frame as point P with the coordinates px and py . For finding the correspondence element to P from the depth map we have to evaluate corresponding index. We can use next equation:

$$index = py' \cdot w + px',$$

where w – width of the video frame. Usually, py and px are given by floating numbers. Index is integer value, we can't use py and px directly, that is why we will use only integer part of this numbers. For reducing error value in the type casting, we have to convert to integer each of number separately, we have:

$$px' = \text{int}(px), \quad py' = \text{int}(py).$$

D. Checking the result. For testing algorithm which described above, we use next activity diagram:

- 1) Hardware and software initializing
- 2) Hand centre depth (HCD) evaluating.

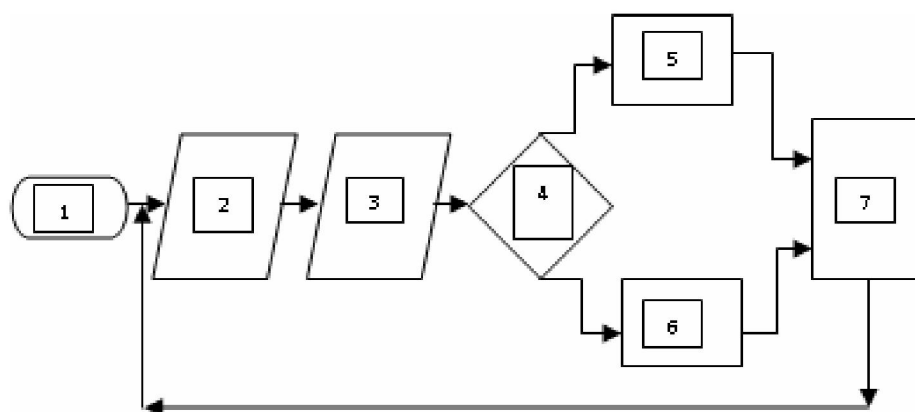


Fig. 4. Basic steps for depth evaluating

- 3) Virtual object depth (VOD) evaluating.
- 4) Check if the HCD is equal to the VOD. If it is true then go to 5th step, else go 6th step.
- 5) Set the virtual object color to blue.
- 6) Set the virtual object color to red.
- 7) Draw virtual object on the video frame and return to 2nd step.



Fig. 5. Depth of the cube and hand is not same



Fig. 6. Depth of the cube and hand is same

E. Further work. Depth map errors often lead to noticeable artifacts in 3D video and significantly decrease the resultant quality. Errors often occur in the occlusion areas. In the [3] purposed a method of filtering depth maps provided by Kinect depth camera. Filter uses output of the conventional Kinect camera along with the depth sensor to improve the temporal stability of the depth map and fill occlusion areas. To filter input depth map, the algorithm uses the information about motion and color of objects from the video. The proposed method can be applied as a preprocessing stage before using Kinect output data.

REFERENCES

1. OpenNI documentation. http://www.openni.org/images/stories/pdf/OpenNI_UserGuide_v4.pdf ARToolKit documentation.
2. <http://www.hitl.washington.edu/artoolkit/documentation/userarwork.htm>
3. Matyunin S., Vatin D., Berdnikov Y., Smirnov M. Temporal filtering for depth maps generated by Kinect depth camera 3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV).

Б. Байсақов, А. Ақшабаев, Л. Найзабаева

3D КАМЕРАСЫНЫҢ ЖӘНЕ ҚАШЫҚТЫҚ ТУРАЛЫ АҚПАРАТТЫҢ КӨМЕГІМЕН АУАНИ НЫСАНДАРДЫ БАСҚАРУ

Бұл жұмыста бір уақытта RGB түсін беретін және нысандарға дейінгі қашықтықты анықтай алатын бейне камераны қолдану түрі қарастырылған. Берілген ақпаратты қолдана отырып, кубты салуға және оны толықтырылған шынайылықта басқаруға мүмкіндік беретін бағдарлама жазылды. Бейне камерадан нысанға дейінгі қашықтықты анықтау үшін арнайы сенсор камералары қолданылды. Толықтырылған шынайылықпен жұмыс істеуге мүмкіндік беретін 3D кубын көрсету мақсатында арнайы библиотека қолданды.

Б. Б. Байсаков, А. К. Ақшабаев, Л. Найзабаева

УПРАВЛЕНИЕ ВИРТУАЛЬНЫМ ОБЪЕКТОМ С ПОМОЩЬЮ ИНФОРМАЦИИ О РАССТОЯНИИ И 3D КАМЕРЫ

Рассмотрен способ использования видеокамеры, которая одновременно на выходе генерирует RGB цвет и информацию о расстоянии до объектов. Применяя данную информацию, написана программа, с помощью которой можно рисовать и управлять кубом в дополненной реальности. Для определения расстояния от камеры до объектов был использован специальный сенсор камеры. Для визуализации 3D куба была использована специальная библиотека, которая позволяет работать с дополненной реальностью.