Improved Simulation of Holography Based on Stereoscopy and Face Tracking

Łukasz Dąbała(✉) and Przemysław Rokita

Warsaw University of Technology, Nowowiejska 15/19, 00-655 Warsaw, Poland
L.Dabala@mion.elka.pw.edu.pl, pro@ii.pw.edu.pl

Abstract. To meet the requirements of the market, people are improving communication with the virtual reality systems. We propose a method for simulating holography, where person can see the object from different points of view. To achieve such effect we used stereoscopy in combination with face tracking, what enables us to manipulate content on the screen. Despite heavy computation load we were able to maintain interactivity of the whole system.

1 Introduction

People always wanted to recreate real world in virtual environment. Such projection gives the possibility to manipulate elements from the surrounding without modifying reality. There are many ways to move the content in the virtual environment, but the simple manipulation of everything on screen with keyboard and mouse is not enough.

Solution for this are virtual reality systems. Each of them must consist of at least three elements. The first one is computer, heart of the system, which is responsible for every calculation. Secondly, the hardware for communicating with the user. This can be handled in many ways, for example with mouse and keyboard, camera, microphone. The third element is equipment, that presents user the result of his/her manipulation. There are many possibilities for showing results, just as it is for taking the input for the system, the simplest one is screen of the monitor. In this work we would like to create such system in combination with stereoscopy, as the enrichment of the experience for the user.

Stereoscopy will give the possibility to recreate depth, so user will be more involved into the virtual world. Stereo vision by itself, cannot be enough to immerse person in computer generated content, so we would like to simulate the holography. For this reason, the part of our system is a simple camera device, that will be used for tracking position of the user. That will help with involving user more into the manipulation and giving the impression, that everything can be seen from different points of view.

2 Background and Previous Work

In this section, there will be presented background for stereo perception as well as an overview of the previous solutions for view-dependent rendering.

M. Kryszkiewicz et al. (Eds.): PReMI 2015, LNCS 9124, pp. 193–201, 2015.
DOI: 10.1007/978-3-319-19941-2_19

2.1 Stereopsis

One of the most complicated systems in human organism is visual system. To achieve good perception it combines information from many different cues (for example occlusion or perspective). The most influential one is stereopsis [6,9]. It has a really strong influence on viewing experience. Human get information about the depth and 3d structure of objects from binocular vision. Because of the position of eyes, binocular vision results in two slightly different images, which are projected to the retinas of the eyes. Usually the difference between images is only in horizontal direction and it is called horizontal disparity. It can happen, that vertical disparities are also present (for example during watching content with reflections or refractions [11]). Excessive disparities can lead to visual discomfort, what results in poor user experience.

2.2 View-Dependent Systems

Previously there were some attempts of creating view-dependent systems. In [10] they tracked user position by using two cameras. The result of the tracking was used as an input for rendering. The system worked in real time, but it did not consider steroscopy. Interesting work has been showed in [5]. In developed system they used stereoscopic rendering and simple camera for face tracking. They focused on reducing distortions and saving as correct 3d perspective as possible. It was done by changing frustum from symmetric to assymetric. In the second work, they considered perception element. But in the end, they considered only movement in horizontal and vertical direction.

There were also some systems that used depth cameras. In one of them [3] authors used depth sensor from Microsoft KINECT to perform the head tracking. It was quite novel approach to this problem, because device they used solved most of the problems with detection of the head's orientation. The output of their algorithm can be projected on a flat surface, which then can be explored by a user. In another approach presented in [13] they improved depth perception by tracking position of the user and creating motion parallax for the rendered image. Their solution was confirmed by creating some interesting applications, that used pseudo 3d effects.

If it comes to the virtual reality systems, the most advanced is probably the one presented in [7]. They proposed a solution for changing a whole room into the augmented reality environment. By using the set of cameras and projectors, they cover whole room with the output pixels, and track user movement. Thanks to the tracking they are able to dynamically adapt the content of the room. They showed new way of creating and presenting augmented reality and in the same way, they created new area that can be explored.

3 Our Method

We are developing further the method presented in [2], so in this section we present short review of the previous method and what has been added to make the solution better.

3.1 Camera Arrangement

In our setup a camera was placed above the monitor, so like in every laptop. The device we used, was really simple, without any depth sensors. During using of our system, user can move his head freely, without any move restrictions.

3.2 Rendering

To simplify rendering part we used deferred shading. Because some objects (reflecting or refracting) can cause more problems [6, Ch. 12] than benefits, we considered only diffuse objects. In this case only thing, that we should be aware of accommodation and vergence conflict. What is connected also to diffuse rendering is limiting too large disparities, because otherwise it will be impossible to see the image in 3d. In our rendering we used image based lighting [4] in addition to the pre-computed ambient occlusion [14].

3.3 Face Tracking

Every view-dependent system uses some kind of tracking. Information, that comes from tracking movements, gives the possibility to manipulate the environment according to the gestures or position of the user. This is what we are aiming at, tracking technique that we used gives us the potentiality to simulate holography. Imitating the effect consists of showing user content on the screen from different perspectives.

The most common technique for showing 3d content is using glasses of different types. Such method has its drawbacks. Because of the glasses and the fact, that they are 'coloured', it is really hard to find position of the eyes and use them for tracking. Sometimes it is nearly impossible to find regions of interest, but to not handle such problem, we remained with face tracking. That gives us more robust solution, because it allows user for wearing glasses also during detection step.

For face detection we used Viola-Johns framework [12]. Because of the way it is working, lighting conditions should be really good. There should not be any over- and unsaturated regions. In the framework all features for detection are represented by combination of sequences of rectangles. Every such region consist of two kind of pixels: white and shaded ones. The value for feature is the difference of the sums of pixels that are inside white regions and shaded ones. By using a method called integral image, it is possible to calculate such feature in constant time, by only four table reads. Integral image can be precalculated by summing values in the grid, so the value in point (x, y) is a sum of all values that are above and to the left of (x, y). For selecting features and training classifiers AdaBoost was used. We used cascade classifiers, because every single part works only on the date from the previous one, so it rejects wrong solution very quickly.

After detection step, we need to perform tracking step. For this we used a method based on motion, called Lucas-Kanade optical flow [8]. The method assumes constant neighbourhood of the pixel and solves equation for it for the flow using least squares criterion.

3.4 View-Dependent Part

The origin of the camera in the scene is changed, to show objects from different perspecitve, so the user will have the impression of the holography. The scene is represented by a virtual sphere, where all objects are in inside the sphere and camera is moving on the surface of it (Fig. 1).

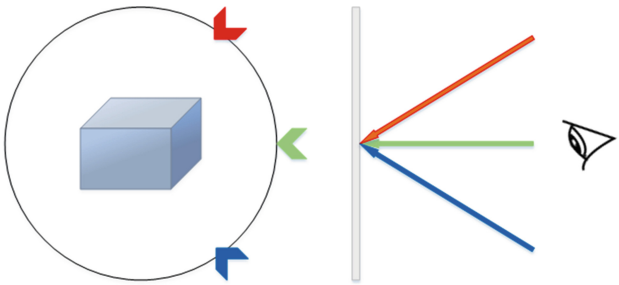

Fig. 1. Scene representation in our algorithm. Different colors of arrows are representing different positions of the viewer, that is watching the screen. Chevrons are representing corresponding camera positions in the scene.

Our alogrithm consists of three paths, like it is shown in Fig. 2. Green one is responsible for calculating movement in XY direction. The yellow path regulates the size of the radius of the virtual sphere. Thanks to this, we are able to make objects bigger or smaller, depending how far from the camera the user is. The last red part takes into account rotations of the user's head. That gives us the possibility to manipulate stereo matrices to add vertical disparity, in amount that depends on the angle.

Movement of the user in vertical or horizontal direction is simple to handle. All information, that is needed to change position of the camera comes directly from the optical flow. We calculate sum of differences in X and Y direction of the tracked points from current frame and previous one. Next, we directly use that values to change the position of the camera, and make it look at the center of the virtual sphere. To remove flickering from the presented content, small movements of the user are just ignored.

What can be done further is changing distance to the observed object. This can be done, by expanding or reducing the length of the radius of virtual sphere. Here, the optical flow, does not give enough information, because it is limited to vertical and horizontal direction. To do this, there is a need to calculate differences between consecutive frames. The problem that arises here is quality of input images, which depends on the video frames coming from the camera. To limit the error, that can happen during this step, consecutive frames are subtracted from each other. For the further calculation we have textures with differences between current frame and the previous one, and between previous one the penultimate. Too small values in such textures are filtered out. Next, the

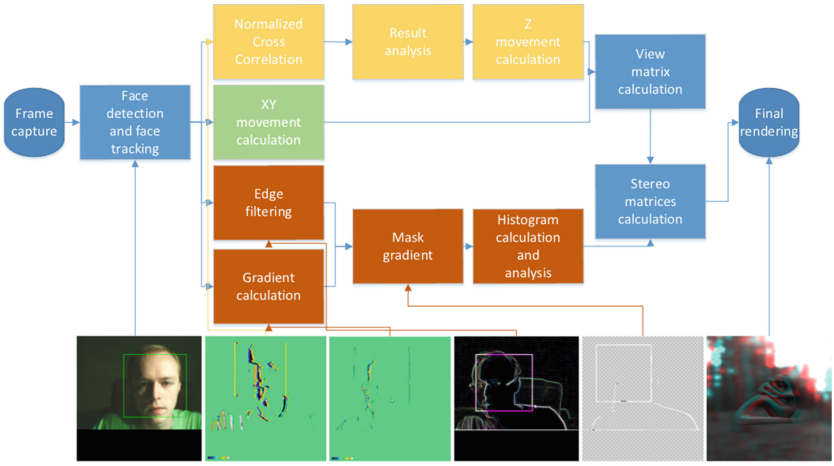

Fig. 2. Flow representing our algorithm.

cross correlation is done between these. The result of this step is an image with disparities in X direction, because vertical coefficient is not important. What needs to be done with the result is checking the sign of the sum of such texture. Moving towards the camera will result in negative values, in opposite direction in positive values. Negative values will shrink radius of the virtual sphere, positive ones will make it bigger.

Stereoscopic component of the application gives us the reason for handling new type of the user movement - head rotations. The result of the operation will be addition of small vertical disparity to the final image. Too much of such disparity can result in bad experiences during 3d watching, but a small can enhance it. What should be done in this step is calculating, how much user has turned his head in XY plane. Input for this step are images that are free of the camera capture errors. First, we calculate gradient in both directions. Such gradient will be calculated for a whole image, which will result in giving values that are not important. To remove them, we calculate the edge image for interesting region and improve it's quality by using erosion and dilation. Such mask is used to remove unnecessary regions. Next, we calculate histogram for such image. That gives us the possibility to predict rotation of head. The highest value in the histogram is the most probable angle of rotation. To prevent numeric errors, we remove values below some threshold, and use this value to add vertical disparity to the image.

4 Results and Discussion

We have created a virtual-reality system for simulating holography, by using stereoscopy and face tracking. We were aiming for creating view-dependent system with limited cost, so with the hardware that everybody can possibly have.

In our case, additional part was only a camera. It gives the possiblity to watch an object from different places and thanks to this, improve user's experience. In comparison to the previous system, which was used as a basic [2], we have improved movement in depth, made algorithm more stable and added stereoscopic component, which resulted in adding vertical disparity to the final image.

In Fig. 3 is shown movement in XY direction. As can be seen, by moving head, we are able to see the object from different angles.

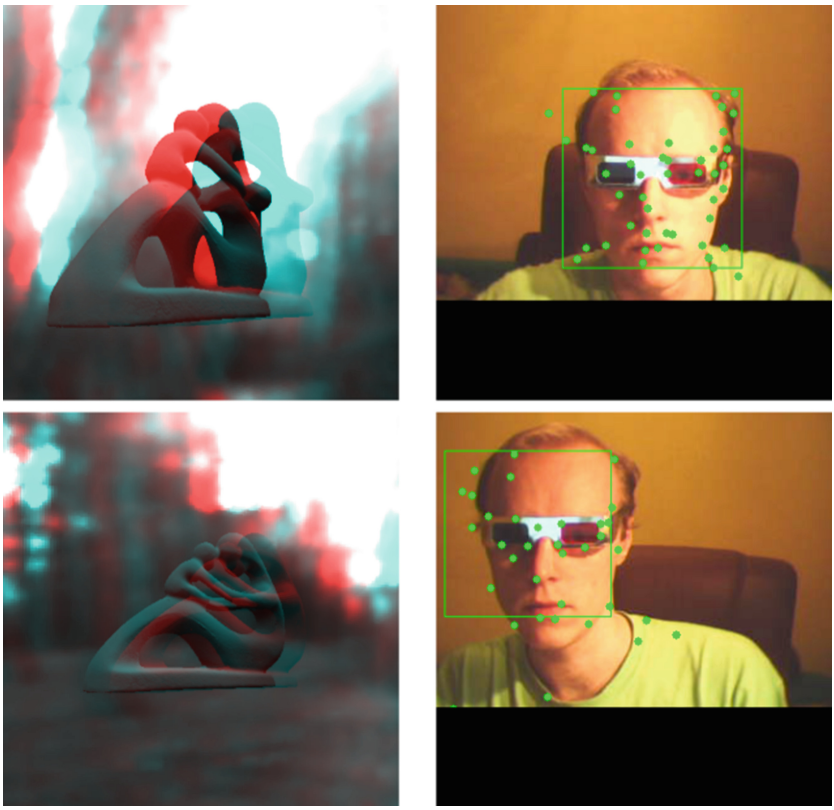

Fig. 3. XY movement - in the first row normal position, in the second row has changed.

Z direction movement can be seen in Fig. 4. By moving further from the camera it is possible to see the object as it is further from the user. The same applies to moving closer. The closer to the camera person will be, the closer the object is.

The most complicated transformation is shown in Fig. 5. By rotating head, small vertical disparity has been added, which can enhance experience in watching 3d content on the screen.

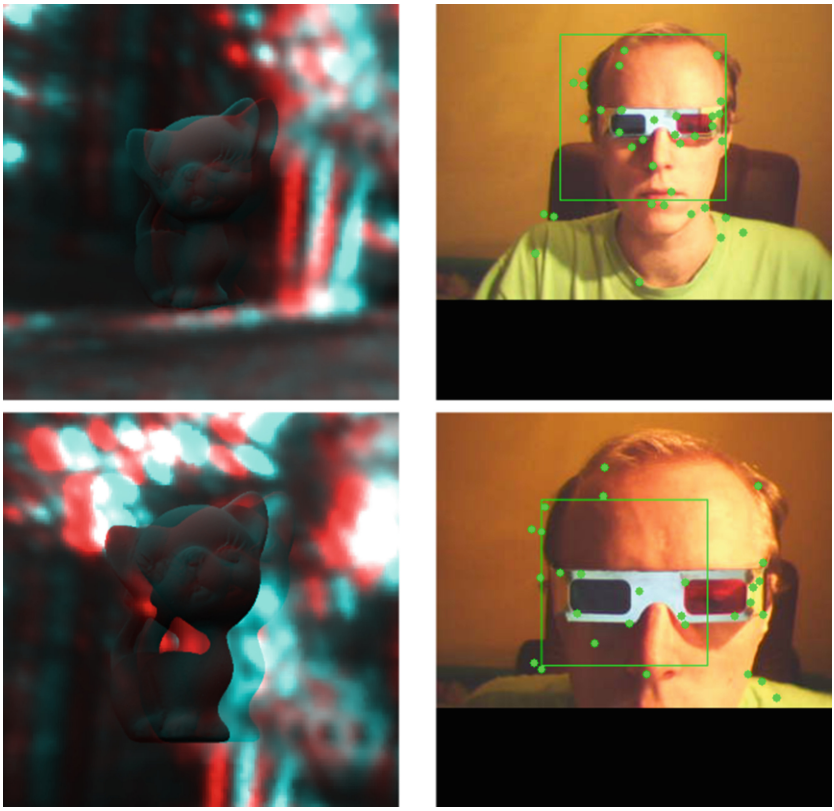

Fig. 4. Z movement - in the first row normal position, in the second row has changed.

We have not limited movement in any direction, so too dynamic movements can cause flipping the sphere, so the object will be seen from the other side. What is more, moving too close or too far away will change disparity too much, so it will be very difficult to verge on the object. This can be misleading for the user. The simplest solution is limiting the movement or correcting the disparity, but it also can cause some difficulties in understanding how user can influence on the rendered scene.

Even with that much processing our system, is quite responsive. We have not tried raytracing as a method of rendering, because it is very time consuming method and it will add more calculations between consecutive frames.

Face tracking technique is now a problem, because it seems that this is the part which is really time consuming. This is the part, that is done on the CPU, so probably moving it to the GPU will make system more responsive. Another problem is dependancy between the detection, tracking and working of whole system. If one of the parts goes wrong, the whole system will not work correctly, but of course, it is inevitable, because face tracking is the heart of the system.

Fig. 5. Head rotations - in the first row normal position, in the second row has changed.

In the future we plan to move face detection and tracking to the GPU to improve speed of the algorithm, because now the delay is visible. We would like to add raytracing to the current pipeline to handle reflections and refractions. Such addition can cause more problems that are connected to the visual discomfort in stereo 3D (for example rivalry). There are now some algorithms [1], that can cope up with this inconveniences, so we plan to add them to the pipeline. After adding raytracing we will be able to give user the chance to see really complicated scenes with many reflections (for example many mirrors), or refractions (materials like glass). There is a possibility to improve the current solution.

References

1. Dąbała, Ł., Kellnhofer, P., Ritschel, T., Didyk, P., Templin, K., Myszkowski, K., Rokita, P., Seidel, H.-P.: Manipulating refractive and reflective binocular disparity. Comput. Graph. Forum (Proc. Eurographics 2012) **33**(2) (2014)
2. Dąbała, Ł., Rokita, P.: Simulated holography based on stereoscopy and face tracking. In: Chmielewski, L.J., Kozera, R., Shin, B.-S., Wojciechowski, K. (eds.) ICCVG 2014. LNCS, vol. 8671, pp. 163–170. Springer, Heidelberg (2014)

3. Garstka, J., Peters, G.: View-dependent 3D projection using depth-image-based head tracking. In: 8th IEEE International Workshop on Projector Camera Systems PROCAMS, pp. 52–57 (2011)
4. Greene, N.: Environment mapping and other applications of world projections. IEEE Comput. Graph. Appl. **6**(11), 21–29 (1986)
5. Nguyen Hoang, A., Tran Hoang, V., Kim, D.: A real-time rendering technique for view-dependent stereoscopy based on face tracking. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013, Part I. LNCS, vol. 7971, pp. 697–707. Springer, Heidelberg (2013)
6. Howard, I.P., Rogers, B.J.: Perceiving in Depth, Volume 2: Stereoscopic Vision. OUP, USA (2012)
7. Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., Shapira, L.: Roomalive: magical experiences enabled by scalable, adaptive projector-camera units. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST 2014, pp. 637–644, ACM, New York (2014)
8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI 1981, pp. 674–679. Morgan Kaufmann Publishers Inc., San Francisco (1981)
9. Palmer, S.E.: Vision Science: Photons to Phenomenology. The MIT Press, Cambridge (1999)
10. Slotsbo, P.: 3D interactive and view dependent stereo rendering (2004)
11. Tyler, C.W., Likova, L.T., Atanassov, K., Ramachandra, V., Goma, S.: 3D discomfort from vertical and torsional disparities in natural images (2012)
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. pp. 511–518 (2001)
13. Zhang, C., Yin, Z., FlorÃłncio, D.A.F.: Improving depth perception with motion parallax and its application in teleconferencing. In: MMSP, pp. 1–6. IEEE (2009)
14. Zhukov, S., Lones, A., Kronin, G.: An ambient light illumination model. In: Proceedings of EGSR, pp. 45–55 (1998)