FROM SMARTPHONE TO VIRTUAL WINDOW

Emile Zhang, Hideo Saito, François de Sorbier

Keio Graduate School of Science and Technology
Hyper Vision Research Laboratory
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, JAPAN
{emile, saito, fdesorbi}@hvrl.ics.keio.ac.jp

ABSTRACT

This paper presents a prototype of virtual transparency on a hand-held device. The prototype is restricted to such device with no additional material to show that a semblance of virtual transparency can be achieved with today's smartphones and has been designed be as simple as possible to use. The user's head is tracked using the front camera and the eyes coordinates are estimated from the head position. The area to be displayed is then computed from the these coordinates, an estimation of the distance head-phone and phone-scene, as well as the phone and lens specifications. It provides a realistic illusion of virtual transparency, not an geometrically accurate render of the scene, and still works in non-optimal situations (non-flat scene).

Index Terms— User-perspective Rendering, Virtual Transparency, Augmented Reality, User Interfaces

1. INTRODUCTION

Virtual transparency is an increasingly popular concept. Transparent screens appear in any futuristic film, rumours of the next iPad screen being transparent go around, talks of the possibility of transparent screen phones hitting the market echoes on Internet. With the rise of portable displays such as smartphones and tablets, the option of a simple and accessible user-perspective rendering has to be explored. Currently, the perspective of the camera differs from the perspective of the user, what he sees does not align with the real world (Fig. 1). We have come to learn and expect that, as it has always been the case, but by no mean it is the most intuitive view. Having a camera working as a window to the world should be a possibility in our day and age.

The idea of virtual transparency is not novel, a number of user-perspective rendering projects have already been attempted. However, they necessitate heavy or cumbersome equipment such as Kinects, Wiimotes, additional cameras and are linked to a computer[2, 3]. HMDs can provide a perfect user-perspective view but are not suited for everyday life, although light and non-intrusive HMDs such as Google Glasses

(a) User perspective rendering

(b) Device perspective rendering

Fig. 1. Side-by-side comparison of user-perspective rendering (left) processed by our Virtual Window prototype and device-perspective rendering (right).

are being developed. Although currently available technology may not be sufficient for a perfectly immersive illusion, this goal may be reached in the near future as new and better hand-held devices appear in growing numbers on the market. Developers as well swell in numbers as new and improved applications appear on stores, providing new ideas and creating new projects. Mobile devices grant the user freedom

of movement, the ability to use their applications where they want when they want. The Virtual Window project attempts to bring a prototype of an easy and user-friendly application enabling user-perspective rendering using Augmented Reality (AR) to the everyday portable device.

Mobile AR does not lack challenge: Tracking the user's head position accurately, gathering information from the device and rendering an accurate model of the scene. If the first task has been made possible with the addition of front cameras to most hand-held devices [4], the second and third tasks remain at hand. However, the Virtual Window project does not aim to reconstruct a geometrically accurate 3D model of the scene. Its goal is to provide a realistic illusion of transparency without it being too demanding so other applications may be added on top of it.

The rest of the paper is structured as follows. Section 2 will cover related work. In Section 3 we will describe our current prototype, its limitations and its working process. Then, in Section 4 we will explain the experiments performed to test the prototype's accuracy and performance. Finally, Section 5 will conclude this paper with a summary.

2. RELATED WORKS

The Virtual Window project is inspired from optical see-through HMDs used in AR, which present a perfect user-perspective view of the real world [1]. Using HMDs however bring some serious issues to the table: People are not used to handle them, and they are cumbersome. User-perspective rendering on hand-held AR devices have been attempted [2, 3], but overlooked one of the main utility of such device: mobility. Both systems being stranded and linked to a computer, the mobility of their prototype is limited and thus work against the very idea of using a hand-held device. However, it allows them to have access to more options, namely IR tracking (by using a Wiimote) or depth estimation (by using a Kinect), something which most hand-held devices are not able to provide yet.

Mobile AR is still relatively new, and working on hand-held devices bring more problems to light, some of which are already well known in the AR community [5, 6]. When trying to provide virtual transparency on a mobile device, one quickly realizes that what the user is supposed to see may not be included in what the device is recording. A proposed solution to this problem was using Wide FOV cameras [7], solution we could not apply since our goal was to keep the user-perspective rendering accurate while keeping the prototype simple both in use and in code.

3. A VIRTUAL WINDOW PROTOTYPE

Our prototype was developed with several key points in mind: it had to be accessible, user-friendly and functional. The system being limited to a phone with no additional material, we had to consider several deviations from other window transparency systems [2, 7].

(a) $d_0 = 20cm, d = 40cm$

(b) $d_0 = 20cm, d = 3m$

(c) $d_0 = 1m, d = 40cm$

Fig. 2. Different (d_0/d) settings and Virtual Window output.

The prototype currently still requires human input for head-phone and phone-background values as we lack depth detection, however our final aim would be for it to fully operate on its own. Note that we are not angling for a perfectly accurate virtual transparency result, which would be impossible to achieve with what we are providing. The goal is simply to have the best, realistic illusion we can create for the user. The present limitations will be discussed below (Section 3.1) before covering the detailed process of the Virtual Window prototype (Section 3.2).

3.1. Apparatus and limitations

The project was conducted with a Samsung I9100 Galaxy S2. It offers both frontal and back lenses, which is essential for the project, with a 8-megapixel main camera. The higher the image quality is the better the immersion will be, as the software will essentially be stretching and zooming a portion of the camera output. The software was developed using Eclipse and OpenCV on Windows 7, 64-bit OS.

Relying entirely on the smartphone introduces several new problems. Our main issue comes from hardware limitation as it is impossible to activate both frontal and back cameras at the same time on the Galaxy S2, making the task of up-

dating the eyes coordinates in real-time impossible. The range of movements is thus limited to rotating the phone around the eyes' position to keep the relative head-to-phone coordinates accurate. The cameras are also unable to estimate depth, so the distance must currently be input manually for close-range situations. If the targeted scene is beyond 3m, the system will switch to long-range estimation and the value of d is not necessary any more. We believe these two limitations will be solved with time as newer and more powerful smartphones will hit the market.

Another problem stems from the main camera's field of view. Depending on the eyes' coordinates, the area rendered may be out of bound and the software will be unable to proceed. A proposed solution would be to use Wide FOV cameras [7], however this issue will not be solved with newer generations of smartphones as it will be very unlikely that their cameras' field of view will be increased. Until now we haven't yet figured out a simple and practical way to bypass that limitation. Adding additional external cameras to increase coverage would defeat the purpose of being able to experience Virtual Window with nothing more than a smartphone as well as severely reducing mobility.

It is relevant to note that although the prototype has been designed with the Galaxy S2 support in mind, it can still easily be adapted to other hand-held devices as long as the mechanical specifications are accessible. The size of the device, the size of the screen and the camera lens attributes are required in order for the project to work accurately. We currently use the Galaxy S2 native head-tracking functions, but we are looking to implement an eye-tracking program which would give us more control over the overall accuracy, as well as allowing the prototype to work on hand-held devices without native tracking functions. As of now, the cameras do not require any calibration process.

3.2. Short-range situation process

As explained earlier adjustments were made necessary due to hardware limitations. The user must first input manually the estimated distance phone-head as well as phone-background. If the distance phone-background is high enough, the process will switch to long-range estimation (Section 3.3). Fig. 2 shows the results using different settings with d_0 (distance head-phone) and d (distance phone-background) in short-range situations. These outputs will always be slightly inferior to long-range ones due to the amount of objects in the scene, as it is to be expected indoor, however Fig. 2 shows that the system works decently even when items clutter the scene.

Once the distance values are stored the front camera and head tracker are activated, eyes coordinates are then estimated from the head position. It is important to note that the head tracker used is already implemented in the Galaxy S2, so the accuracy will depend on the device used. If available, eye tracking will very likely return better results as there will be no need to compute estimations. An *out of bound* case (Fig. 3) will occur under the following situation, forcing the user to move and change eye coordinates:

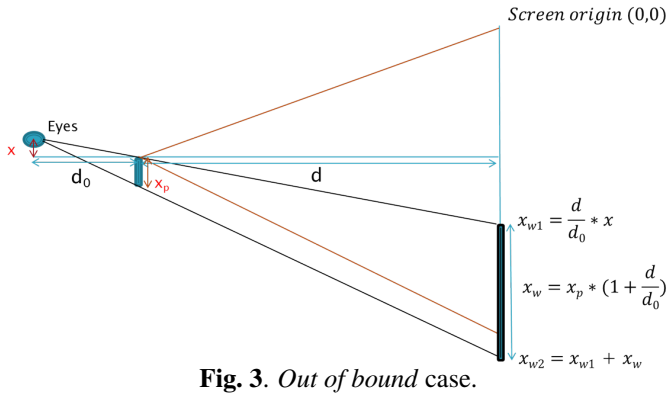


Fig. 3. *Out of bound* case.

Fig. 4. Upsampling Virtual Window area.

$$x_{w2}/x_v > 0.5 \tag{1}$$

where x_{w2} represents one of the edge of the Virtual Window area and x_v is either the height or length of the camera output image, estimated by using the lens size, focal length and d. Using the Galaxy S2, most *out of bound* situations occurred when the user's head was too much on the left as the main camera lens is located on the left edge of the phone.

When receiving correct eye coordinates, the front camera will turn off allowing the main camera to activate. Ideally the front camera would stay active to update eye coordinates and allow for a wider range of movements, but with the device used this has proven to be impossible. The Virtual Window size (x_w, y_w) and area edges $((x_{w1}, y_{w1}), (x_{w2}, y_{w2}))$ are computed:

$$x_w = x_p * (1 + d/d_0) \tag{2}$$
$$y_w = y_p * (1 + d/d_0) \tag{3}$$
$$x_{w1} = d/d_0 * x - 10 \tag{4}$$
$$y_{w1} = d/d_0 * y + 30 \tag{5}$$

where (x, y) is eyes coordinates and (x_p, y_p) is the phone's size. Note that the values $(+10/-30)$ will depend

on the device used as they represent the difference in position between the front camera lens and the main camera lens. The Galaxy S2 has the front camera lens 10mm to the left and 30mm below the main camera lens. The size of the scene captured by the camera (x_c, y_c) is then estimated from the camera lens specifications:

$$x_c = f/x_l * d \tag{6}$$
$$y_c = f/y_l * d \tag{7}$$

where (x_l, y_l) is the lens size and f its focal length. The Virtual Window area is then cropped from the main camera output and upsampled to the screen size (Fig. 4).

3.3. Long-range estimation process

The method presented above does not work properly when the value of d is too high. While it is fairly easy to estimate distance indoor in close-range situations, it becomes a lot less so outdoor. Depending on the level of accuracy required, our long range estimation method can be used for $d > 3m$ with an error margin of less than 3% when compared with the regular method used above (further analysis of the numbers will be treated in Section 4.1). Long-range estimation does not require the user to input the value of d as it will make approximations with $d >> d_0$. The Virtual Window area to camera output ratio is then calculated as followed:

$$x_w/x_c = (x_l * x_p)/(f * d_0) \tag{8}$$
$$y_w/y_c = (y_l * y_p)/(f * d_0) \tag{9}$$
$$x_{w1}/x_c = (x_l * x)/(f * d_0) \tag{10}$$
$$y_{w1}/y_c = (y_l * y)/(f * d_0) \tag{11}$$

The two methods applied to a long-range situation can be seen in Fig. 5. As expected, the close-range method with $d = 3m$ returns a much worse result (the scene is estimated to be around 6m).

(a) Long-range estimation

(b) Short-range method

Fig. 5. Side-by-side comparison of long-range estimation (left) and short-range method with $d = 3m$ (right).

4. EXPERIMENTAL RESULTS

Upon closer inspection, Fig. 2 reveals several small incoherences. Although the illusion seems acceptable at first glance, the final output is far from being perfect. As explained in Section 3.2, the prototype works by cropping and upsampling part of the camera output, which means that the only transformation used is stretching an image, ignoring any 3D related questions. The best results are achieved by viewing a flat background such as a wall or a screen.

Further testing show that the Virtual Window prototype still works decently in non-ideal conditions (Fig. 7 (b)), where the settings were meant for the door behind (Fig. 2 (b)). Slight errors when inputting depth values therefore do not damage the output too much, an important fact as the distance must be evaluated by the user and will not always be exactly on point. Extremely maladapted values (Fig. 7 (a)) will still return a bad output.

The Virtual Window prototype is also unable to work if the value of d_0 or d is too small, as both situations will result in an *out of bound* case: the camera output will just cover less area than what the user expects to see.

4.1. Long-range estimation accuracy

Long-range estimation allows the user to use the prototype without having to estimate d. As our final goal is to have an automatic system that does not require any manual input, long-range situation bypasses the current problem of smartphones cameras unable to estimate depth. Furthermore depth cameras are usually more accurate at close range, which means long-range estimation will still be relevant once depth cameras will be introduced to the system. Fig. 6 shows the

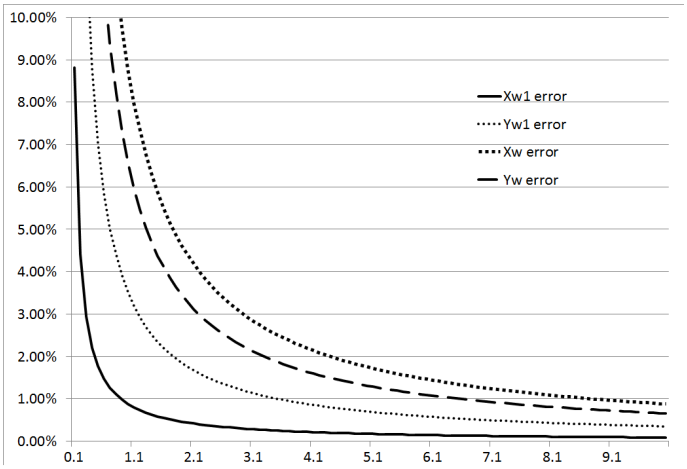
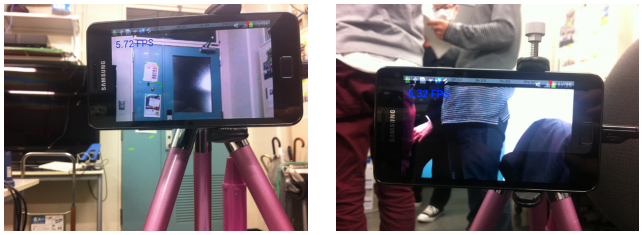


Fig. 6. Error percentage between $x_w/x_c, y_w/y_c, x_{w1}/x_c$, y_{w1}/y_c close-range and long-range values, depending on the distance d (in meters)



(a) Too far: $d = 40cm$ (b) With obstacles: $d = 3m$

Fig. 7. Virtual Window output in non-ideal situations.

error percentage when comparing long-range estimation to close-range ratios values.

As stated earlier, the switch between close-range method and long-range estimation depends on the precision required. As the error percentage is additive, $d = 3m$ carries an error margin of around 3%, drops under 2% at $d = 5m$ and can be consider to be inferior to 1% when $d > 10m$.

4.2. Discussion

One essential problem has yet to be tackled as pictures cannot show it well. When using the prototype, in most situations the user's eyes can't focus both on the phone's screen and the background at the same time, making either blurry. The only case where this problem doesn't appear is if d_0 has a large value while d has a small one. This is counter-intuitive as d_0 is usually small since the user holds the phone in his hand, and we just explained that a small value of d favours *out of bound* situations. Virtual transparency will very likely not be able to emulate real transparency as long as this problem exists, it is thus necessary to keep the user's focus on the screen.

Although this proves to be an issue here, our original plan was to develop applications with the Virtual Window prototype once it proved to be functional, which appears to be the case. Keeping the user's attention on the screen will result from these applications, such as adding markers (Wikitude, Layar, etc) or adding image recognition.

5. CONCLUSION AND FUTURE WORK

In this paper we have presented a functional prototype of virtual transparency on a hand-held device using no additional material than said device. It does not return a geometrically correct scene, however the illusion of transparency is present and fairly robust to interference (unwanted items in the scene, slightly wrong distance values) and the system can also bypass the depth estimation problem for long distance objects. We have noted the problem introduced by human focus, and will direct our future research on virtual transparency as a mean rather than an end to keep the user's focus on the device. An eye-tracking program independent to the actual device will also be worked on.

6. ACKNOWLEDGEMENT

This work was partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004.

7. REFERENCES

[1] Ozan Cakmakci and Jannick Rolland, "Head-worn displays: A review," *Journal of Display Technology*, vol. 2, no. 3, pp. 199–216, 2006.

[2] Domagoj Baricevic, Cha Lee, Matthew Turk, Tobias Hollerer, and Doug A. Bowman, "A hand-held ar magic lens with user-perspective rendering," *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 197–206, 2012.

[3] Makoto Tomioka, Sei Ikeda, and Kosuke Sato, "Rectification of real images for on-boad camera tablet-based augmented reality," *IEICE Technical Report*, 2013.

[4] Stylianos Asteriadis, Kostas Karpouzis, and Stefanos Kollias, "Head pose estimation with one camera, in uncalibrated environments," *Proceedings of the 2010 workshop on eye gaze in intelligent human machine interaction*, pp. 55–62, 2010.

[5] Ernst Kruijff, J. Edward Swan II, and Steven Feiner, "Perceptual issues in augmented reality revisited," pp. 3–12, 2010.

[6] Ronald T. Azuma, "The challenge of making augmented reality work outdoors," pp. 379–390, 1999.

[7] Hill Alex, Schiefer Jacob, Wilson Jeff, Davidson Brian, Gandy Maribeth, and MacIntyre Blair, "Virtual transparency: Introducing parallax view into video see-through ar," *2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, vol. 12, pp. 239–240, 2011.