User Perspective Rendering for Hand-held Augmented Reality using Deep Neural Networks

Ashwin Pathak
International Institute
of Information Technology,
Hyderabad, India 500032

Parishit Sakurikar
International Institute
of Information Technology,
Hyderabad, India 500032

P.J. Naryanan
International Institute
of Information Technology,
Hyderabad, India 500032

***Abstract*—The Handheld Augmented Reality magic-lens paradigm has been typically implemented by rendering the video stream captured by the back-facing camera onto the device's screen. However, this rendering displays the real world from the device's perspective which causes misalignment and inconsistencies due to spatial distortions presented from the device's perspective. This paper provides a user perspective rendering approach based on projective transformations.**

The implementations concerned with the user perspective rendering have the Augmented object statically displayed which causes an interrupted real time experience to the user looking for a seem-less consistency between the magic-lens and real world. This paper provides the movement of the augmented object along with the user-perspective rendering, thus, increasing the interactivity and enriching the seem-less experience for the user. The earlier implementations of the user perspective rendering use primitive approaches to detect the face, however, with the advancement of the deep neural networks and the increased capabilities of hand held devices, this paper has attempted to implement the problem of user perspective rendering using optimized versions of posenet.

I. Introduction

The applications of the Handheld Augmented Reality(AR) finds its use in many examples like : to provide visual support in maintenance [1] and construction site monitoring [2], for AR games [3] or maps [4], and in many more situations [5].

Handheld devices generally implies smart-phones or tablets. The magic-lens paradigm that is normally considered as a presentation of the Handheld Augmented Reality involves displaying the real world scene by partly converting into AR while keeping the rest of the scene unaffected. This type of rendering, known as device perspective rendering (DPR) causes a misalignment due to spatial distortions between the scene displayed through the magic-lens and the real world. This misalignment has been shown to be confusing in selection tasks where the part of the object lies with the magic-lens and also in the real-world [6].

User Perspective Rendering (UPR) has been proposed to overcome this problem [7]. The rendering is defined as the geometrically correct view of a scene from the point-of-view of the user, in the direction of the user's view, and the exact view frustum the user should have in that direction. Figure 1 and Figure 2 demonstrates the difference between DPR and UPR. The misalignment between the real-scene and the displayed image is evident in DPR while in UPR that inconsistency is absent. UPR poses three challenges (1) track the user's head positions accurately, (2) track the pose of the display with respect to the world accurately, and (3) obtain/ create an accurate model of the scene. Implementations of UPR have often been explored in various ways by using depth sensors or external camera tracking system, however, such kind of settings are not appropriate for hand-held devices. UPR implementations on hand-held devices are mostly implemented using 3D face tracking systems. Then using the head pose, the device perspective is transformed to the user perspective either using homography or perspective geometry based approach. However, most of the existing applications are based on the assumption that the user is stationary and only the hand-held device is moving. We discard this assumption and introduce

Fig. 1. Device Perspective Rendering

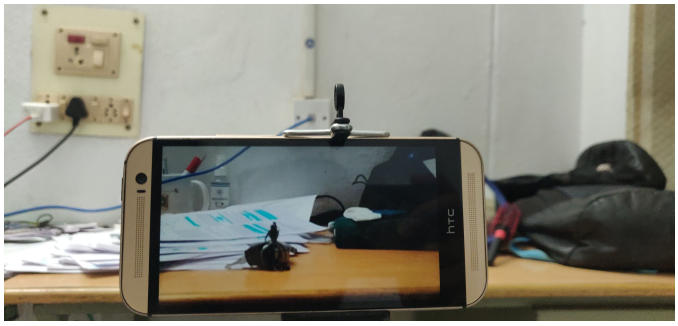

Fig. 2. User Perspective Rendering

freedom of movement of the hand-held device as well as the user.

This paper extends the concept of homography based UPR approach to a projective geometry approach for attaining UPR. The implementations of UPR generally do not consider augmented object also to be moving. The augmented object and the world scene are considered to be coupled together. This paper aims at using deep neural networks to track the position of the eyes in real time along with decoupling the augmented object and the real scene making the interactions more flexible and removing the need to ensure the consistency between the augmented object and the real scene. The decoupling of the object and the scene provides a lot of freedom to the users in real time.

Thus the major contributions of the paper can be summarized as follows :

(1) The paper allows user-perspective rendering by allowing the user as well as the hand-held device to not be restrained.

(2) The paper proposes a deep learning approach to retrieve the head pose and use the obtained pose to estimate the user perspective rendering.

(3) The paper proposes an approach for projective geometry while also providing a relationship between DPR and UPR in a planar environment.

(4) The paper decouples the augmented object and the real scene thus allowing more flexibility of movement. This also removes the need for computing homography.

II. Related Work

Effects of display size on UPR was evaluated by Baricevric et al. using a simulation and found that, a tablet-size display allows for significantly faster performance of a selection task compared to DPR [7]. They also proposed kinect based reconstruction and Wiimote for head tracking to achieve UPR. This concept was further extended to use the gradient domain image-based rendering method with stereo matching and semi-dense stereo [8,9].

Tomioka et al. and Hill et al. proposed a homography based approach for achieving transformations from the back-facing camera to the user's head pose [10]. They also outline various augmentation methods and analyze the comparisons between UPR and UPR + homography. There approach however deals with a stationary mannequin for face-tracking, thus restraining the flexibility of the system. Our work is highly related to the approach present in this paper. Our work decouples the augmented object with the scene, thus, providing more flexibility and removes the need to apply homography on top of UPR.

Puchihar et al. proposed a fixed point of view UPR (FUPR) [11]. They assume that the user's face is in a fixed and predetermined position while interacting with system. Instead of tracking the users head pose the author manually measure the distance of the head to the device once at the beginning of the application, and they assume the user looking perpendicular through the center of device over the entirety of the application. However, FUPR fails to generate UPR for large interaction spaces.

Grubert et al. employed UPR on mobile devices by combining head tracking using the built-in front-facing RGB camera for head tracking and natural feature-based tracking of the AR device using back-facing RGB camera [12]. Samini et al. proposed a perspective geometry approach to UPR. They proposed a dynamic view frustum based approach to address the registration inaccuracies that are caused by the detection and pose estimation errors [13]. However, due to absence of implementation details and results, the approach cannot be evaluated adequately.

Mohr et al. extends the concept of FUPR with thresholding and optical flow to reduce the computation costs for UPR [6]. They perform motion estimation using KLT tracking and spatial and temporal thresholding to decide if an update of the user's 3D head pose in necessary for FUPR. Thus, their implementation is strongly based on the assumption that the user is looking perpendicular through the center of the device.

III. User-Perspective Rendering by Projection

The method proposed in this paper generates the user-perspective rendering by using user-perspective projection.

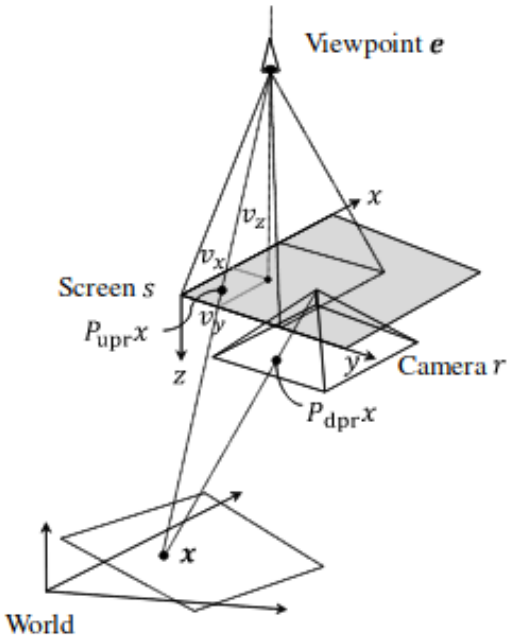

Fig. 3. User-perspective projection

We consider a hand-held device with two cameras front-facing camera (f) and back-facing camera (r). The intrinsic parameters K_c and extrinsic parameters M_c of each camera $c \in \{r, f\}$ are computed before-hand. The relative pose between the screen and each camera $M_{c \to s}$ is also computed. Thus, the

projections can be represented as :

$$P_{\text{dpr}} = K_{\text{r}} M_{\text{r}} \tag{1}$$

$$P_{\text{upr}} = P_{\text{e}} M_{\text{r}\to\text{s}} M_{\text{r}} \tag{2}$$

where P_e is the perspective projection matrix of a view frustum formed by the viewpoint and the screen surface as shown in Figure 3.

$$e = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}^{\intercal} \tag{3}$$

The matrix for P_e can be represented as follows :

$$P_{\text{e}} = \begin{bmatrix} -e_z & 0 & e_x & 0 \\ 0 & -e_z & e_y & 0 \\ 0 & 0 & 1 & -e_z \end{bmatrix} \tag{4}$$

The back-facing camera will be capturing the scene using P_{dpr}, to transform the scene from P_{dpr} to P_{upr}.

We denote the world coordinates of the real scene as X. x denotes the scene X captured by P_{dpr} and x' denotes the scene X captured by P_{upr}. Then we have the relation between x, x' and X as :

$$x = P_{\text{dpr}} X = K_r M_r X \tag{5}$$

$$x' = P_{\text{upr}} X = P_e M_{\text{r}\to\text{s}} M_r \tag{6}$$

$$M_r^{-1} K_r^{-1} x = X \tag{7}$$

$$x' = P_e M_{\text{r}\to\text{s}} M_r M_r^{-1} K_r^{-1} x \tag{8}$$

$$x' = P_e M_{\text{r}\to\text{s}} K_r^{-1} x \tag{9}$$

The rendered scene from user-perspective however, will remain inconsistent and will be misaligned due to distortions with the augmented object if they are transformed together, hence, we propose to decouple the augmented object with the rendered user-perspective scene. To obtain the position and scale of the augmented object, the face-position can again be used in a similar way as used for UPR to transform the augmented object to the desired position. This way, we propose that the augmented object will be free from distortions because the geometric transformations provided in terms of translation and scaling is independent of the scene translation, as a result the inherent rigid structure of the augmented object will remain preserved. This also removes the need to compute homography for the augmented objects as a result avoiding the issues of inconsistency all-together.

For obtaining the position of the eye, we use posenet architecture to obtain the position of eyes and nose (to verify the localization). The deep neural network takes the head pose from the front camera in real time and returns the position of the eyes and the nose which is used for obtaining threshold and to obtain the user perspective rendering.

Prototype Systems

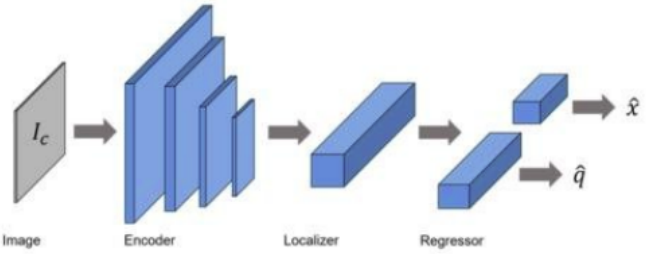

Fig. 4. Pose-Net architecture

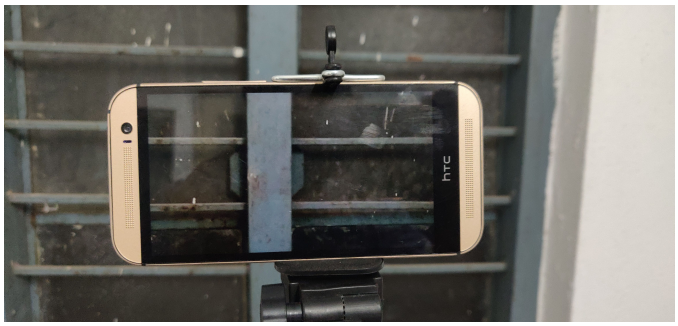

Fig. 5. Samples from UPR

A. Hardware and Software Configuration

1) Hardware configuration: : We implemented our prototype using HTC One M8 (CPU: Quad-core 2.3 GHz Krait 400, GPU: Adreno 330, RAM: 2GB, size: 146.4 x 70.6 x 9.4 mm, weight: 160g). The smart-phone has two cameras that works simultaneously(front: 5MP, 1080 X 720 pixels, 30 fps, rear: 4MP, 1080 X 720 pixels, 60fps).

2) Software configuration: : For marker based tracking and detection, we use Vuforia SDK with Unity 3D game engine overlay the augmented object on the marker in the magic lens. For head pose estimation we use the Camera.Face API for 2D face tracking, we extend the tracker to 3D by using the area of the face as an estimation to the depth. Spatial thresholding also is applied along-with on the augmented object and the scene transformations to ensure the consistency between the augmented object and the real scene. The relative pose between the rear-facing camera and the screen is computed using image of planar mirror reflections. We followed the implementation proposed in [13,14]. We use OpenSFM to solve the structure from motion and bundle adjustment for the feature points between the virtual front-facing camera captured by reflection and the back-facing camera.

We use pre-trained model for posenet from tflite library but fine-tune it to only give position of the eyes and nose as an output.

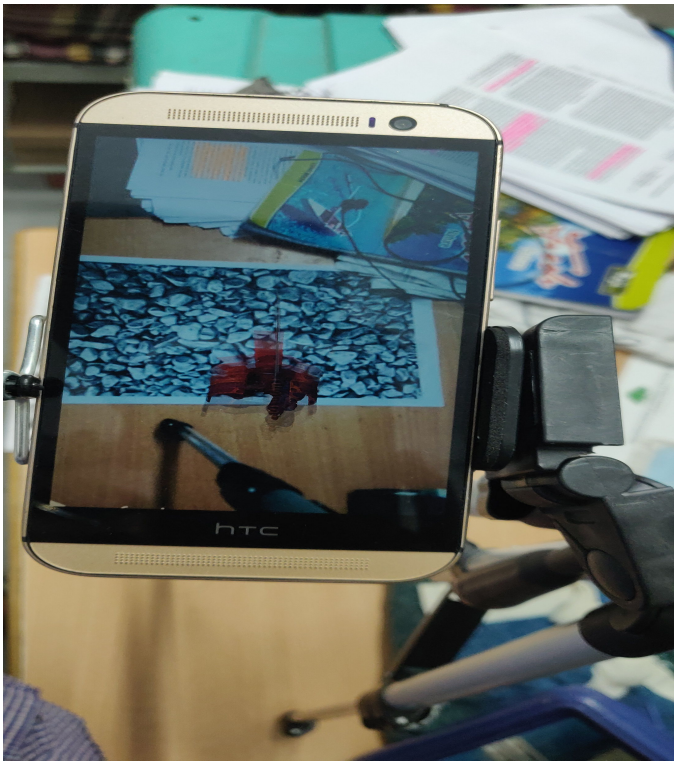

Fig. 6. UPR with Augmented Object

B. Operation test

As shown in figure 4 and figure 5, we analyzed the UPR in a cluttered environment. The depth of the scene was in the range of 0.5 - 3 m. The images in figure 4 and figure 5 are captured from One Plus 6. We can clearly observe that the displayed image and the real scene appear geometrically continuous with slight marginal discontinuities.

IV. Performance Analysis

The performance analysis compared to other implementations will go here.

V. Experiments

A. UPR and Augmented Reality test

As show in figure 5, the augmented object is rendered on the marker along with its responsiveness to the user's head pose. The user-perspective rendering observed in figure 5 shows that continuity of the geometry and objects kept in the scene along with augmented object.

B. Other Experiments

The experiments and survey based on the selection task will go here along with the discussion on that.

VI. Conclusion

This paper proposes a method for presenting real images on magic-lens for AR. The real scene is rendered by projective transformation. The implementation decouples the augmented object with the user-perspective thus eliminating the need to compute homography. Our future work is to improve the head-tracking part using deep neural networks.

Acknowledgment

The authors would like to thank...

References

[1] P. Mohr, B. Kerbl, M. Donoser, D. Schmalsteig, and D. Kalkofen, Retargeting technical documentation to augmented reality. In *Proceedings of CHI'15,* pages 3337-3346, 2015.

[2] S. Zollmann, D. Kalkofen, C. Hoppe, S. Kluckner, H. Bischof, and G. Reitmayr. Interactive 4d overview and detail visualization in augmented reality. In *Proc. of IEEE ISMAR,* pages 167-176,2012.

[3] E. Andrukaniec, C. Franken, D. Kirchoff, T. Kraus, F. Schondorff, and C. Geiger. Outlive an augmented reality multi-user board game played with a mobile device. In *Proc. of ACE,* pages 501-504, 2013.

[4] J. Grubert, M. Pahud, R. Garsset, D. Schmalstieg, and H. Seichter. The utility of magic lens interfaces on handheld devices for touristic map navigation. *Pervasive and Mobile Computing,* 18:88-103, 2015.

[5] D. Schmalstieg and T. Hollerer. *Augmented Reality: Principles and Practice.* Addison Wesley Professional, 2015.

[6] P. Mohr, B. Kerbl, M. Donoser, D. Schmalsteig, and D. Kalkofen. Adaptive User Perspective Rendering for Handheld Augmented Reality. *Proc. of IEEE ISMAR*, 2017.

[7] D. Baricevic, C. Lee, M. Turk, T. Hollerer, D. Bowman. A hand-held AR magic lens with user-perspective rendering. In *Proc. of IEEE ISMAR*, pages 197-206, 2012.

[8] D. Baricevic, T. Hollerer, P. Sen and M. Turk. User-perspective augmented reality magic lens from gradients. In *Proc. of VRST*, pages 87-96, 2014.

[9] D. Baricevic, T. Hollerer, P. Sen and M. Turk. User-perspective ar magic lens from gradient-based ibr and semi-dense stereo. *IEEE TVCG*, PP(99):1-1.2016.

[10] M. Tomioka, S. Ikeda and K. Sato. Approximated user-perspective rendering in tablet-based augmented reality. In *Proc of IEEE ISMAR*, pages 21-28, 2013.

[11] K. Copic Pucihar, P. Coulton, and J. Alexander. Evaluating dual-view perceptual issues in handheld augmented reality: device vs. user perspective rendering. In *Proc. of ACM International conference on Multimodal Interaction*, pages 381-388, 2013.

[12] J. Grubert, H. Seichter, and D. Schmalstieg. Towards user perspective augmented reality for public displays. In *Proc of. IEEE ISMAR*, pages 339-340, 2014.

[13] R. Rodrigues, J. a. P. Barreto, and U. Nunes. Camera pose estimation using images of planar mirror reflections. In *ECCV, ser. ECCV'10*, 2010.

[14] A. Delaunoy, J. Li, B. Jacquet and M. Pollefeys. Two cameras and a screen: how to calibrate mobile devices? In *3DV*, 2014.