2013 IEEE International Symposium on Multimedia

Undistorting Foreground Objects in Wide Angle Images

Mahdi Abbaspour Tehrani
Department of Computer Science
University of California Irvine
Email: mabbaspo@uci.edu

Aditi Majumder
Department of Computer Science
University of California Irvine
Email: majumder@ics.uci.edu

M. Gopi
Department of Computer Science
University of California Irvine
Email: gopi@ics.uci.edu

Figure 1: Examples of perspective distortions.

***Abstract*—The use of wide angle lens on commodity cameras are becoming increasingly popular. However, this leads to image distortions due to the large difference in relative orientation of the different foreground objects with respect to the camera's image plane. More importantly, these distortions are different in different parts of the image since it depends entirely on the position and orientation of the foreground object with respect to the image plane. Such distortions often manifest themselves as objects in the foreground appearing fatter than they are supposed to be or parts thereof (e.g. hands or head of people) looking disproportionately larger than the rest of them. Though there are earlier works addressing other common distortions in cameras like radial distortions, little attention has been given to this problem. In this paper, we present an effective method to remove such distortions in foreground objects minimally distorting the background. This is achieved efficiently using a mesh-based pixel displacement technique assisted by a simple and intuitive user interface design.**

***Keywords*-Image Editing; Image Distortion;**

I. Introduction

Wide angles lenses are becoming more and more popular in commodity cameras. Unlike a narrow angle lens camera, in this case the foreground object may be very different in orientation than the image plane. As a result, such objects show significant distortions (Figure 1) and are usually entirely different for different foreground objects. Such distortions are called *perspective distortions*. For foreground humans, for example, these distortions can manifest themselves as stretching (leading to the perception of a fatter individual), or parts of the body appearing disproportionately larger or stretched than other parts. However, the background usually is much farther away from the image plane and hence does not show such distortions. Also note that this distortion is very different from the standard non-linear lens distortion which usually changes straight lines to curves. The causes of this distortion are thus entirely different (Figure 2) and need to be handled differently.

In this paper, we present the first work that identifies and addresses the perspective distortion. The challenge lies in undistorting different foreground objects differently based on the relative difference of its orientation with respect to the image plane while minimizing the background distortions. First we segment the distorted foreground object and undistort it by projecting it to a different, more appropriate image plane. This creates holes in the background of the undistorted foreground object that need to be filled. We use a mesh-based pixel displacement technique to warp the background to fill the holes. The algorithm is described in details in Section II.

A. Related Work

Our work falls within the general category of image editing applications. There has been several works in this direction in the past dealing with both the geometric (location of pixels) and photometric (color of pixels) aspects of the image [1]–[3]. Since our work deals with movement of pixels in the image, it falls in the category of editing the image geometry and does not change its photometry. Similar operation can be achieved manually using photoshop or similar other software, but consumes considerable time. Our solution is an effort to achieve this interactively and automatically.

978-0-7695-5140-1 2013
U.S. Government Work Not Protected by U.S. Copyright
DOI 10.1109/ISM.2013.17

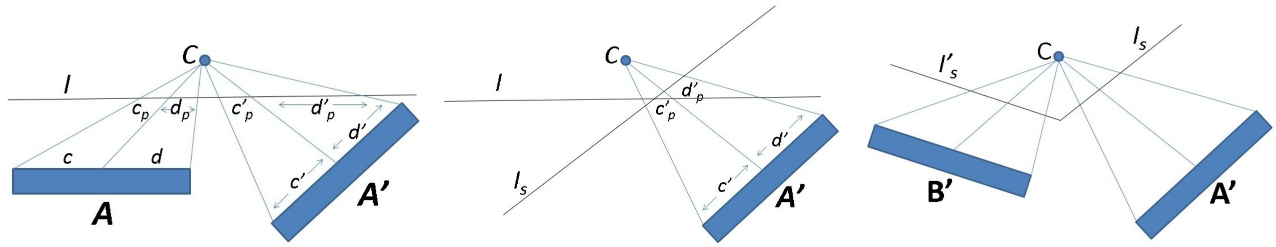

Figure 2: Left: This figure illustrates the cause of the perspective distortion. Foreground object A's orientation is parallel to the image plane providing an undistorted image ensured by similar triangles and hence the same ratio $c : d$ and $c_p : d_p$. However, note that things are different for object A' whose orientation and depth are different from the image plane. The ratio $c' : d'$ can be very different from $c'_p : d'_p$ resulting in image distortion. Middle: This figure shows the suitable plane I_s for the projection of A'. However, the background still needs to be projected in I to avoid distortion. Right: This figure shows that the appropriate projection planes for different foreground objects can be different. In this work, we assume the image plane for undistortion is a rotated original image plane either in the horizontal or vertical direction, and the rotation angle is computed from the 'unstretching factor' provided by the user.

Any image undistortion technique needs to move pixels or change pixel locations and has been termed in various manner like geometric image transformation, image warp, image distortion and so on. Several such techniques have been explored in the past. Depending on the problem at hand, the pixel movements can be global (where all pixels are moved using the same function) or spatially varying (where different pixels in different location of the image are moved differently); linear or non-linear depending on the kind of function used to define the movement; or content-aware (where the movement respects the content of the image in some manner) or content-agnostic (where the pixel movements do not depend on the content of the image). Also, note that any such image pixel movement technique usually involves mismatched resolution of the source and the target image where different kinds of interpolation and/or subsampling techniques are used to generate the target image.

The classic problem of changing the pixel locations in an image is the one that corrects for the non-linear lens distortion. Several models can be used to apply a global non-linear content-agnostic image warp to achieve this correction [4] [5] [6] [7] [8]. Global linear image warps have been used for keystone correction in projectors [9]. More complex spatially varying non-linear image warps have been used for stitching images on planar [10]–[12] and non-planar [13]–[15] surfaces in multi-projector displays. However, warps in the context of multi-projector displays has usually been content-agnostic. A large body of work exists on image retargeting [16]–[21] where pixels are moved differently for foreground and background to achieve different effects on the specific content of the image. Thus these are highly content-aware spatially varying movement of pixels. When viewed from this perspective of the different categories of changing pixel locations, our method is close to the image retargeting methods and achieve a spatially varying content-aware movement of pixels.

B. Main Contribution

The main contribution of this paper are as follows. First, we present the first work to address the local image distortions most common in wide angle images due to the relatively large difference in the orientation of foreground objects with respect to the image plane. Second, to achieve this with minimal distortion in the background, we use a mesh-based pixel displacement technique. Our method can be easily included in any image editing software to provide an easy and effective solution to this problem very commonly faced in today's commodity cameras.

II. Algorithm

Figure 2 illustrates the distortion we address in this paper. Consider the camera C with a wide field-of-view (FOV) as is common in cameras with wide angle lens. Let I be the image plane. Consider two objects A and A'. A is almost parallel to the image plane while A' has a large variation in orientation with respect to I. Let us consider the equal parts of A as c and d and that of A' as c' and d'. Let their projection on I be denoted by c_p, d_p, c'_p and d'_p respectively. Note that the projection of A maintains the proportion between the parts, i.e. $c : d$ is similar to $c_p : d_p$. However, the same is not true for A' where $c' : d'$ differs significantly from $c'_p : d'_p$. Thus, the part d' of A' would look stretched. This is the kind of distortion that we address in this paper. As shown in Figure 1, this manifests itself as stretching of the figure, or stretching of body parts or disproportionate sizes of body parts. This types of distortions are more common in wide angle cameras. Further, this distortion is perceived more in people since unlike buildings and other large inanimate objects, we are more used to seeing them in non-perspective views.

Figure 3: This figure shows our foreground correction operation. From left to right:(a) First, the user marks the picture using GrabCut interface to segment the foreground object. (b) The segmented foreground, denoted by F. (c) (c) The distorted foreground object is projected in a more appropriate plane to create an undistorted foreground object F'. This removal of foreground object stretching leaves holes in the background. The boundary of F and F', denoted by b (red) and b' (blue) respectively, is shown in this image. The new foreground projection plane is computed by rotating part of the original image plane. (d) Difference between b and b' can be reduced further by translating b' to a more appropriate position. This amounts to translating the projection plane.

Our algorithm has two main components. First, we achieve a foreground correction (Section II-A). This step is assisted by the user using a simple and intuitive user interface. However, correcting part of an image creates holes in the background which are then filled by displacing the pixels in the neighboring background smoothly from the boundary of the undistorted foreground object (Section II-B). This smooth pixel displacement is achieved by meshing the background part of the input image, and displacing the vertices of the textured mesh triangles to the new locations dictated by the correction process.

A. Foreground Correction

In this section, we describe our user assisted foreground correction method. The result of this step in shown in Figure 3.

Segmentation: The goal of this step is to extract the distorted foreground. For this we use the GrabCut [22]. We use the standard user interface used in GrabCut to mark the foreground and background appropriately to segment the distorted foreground. The user can improve this segmentation, as in GrabCut, using feedback lines on the background. Let $O(x,y)$ be the original image (where (x,y) denotes the image coordinates) and the segmented foreground object be $F(x,y)$.

Foreground Correction by Image Plane Rotation: The distortion in F is due to the large difference in the orientation of the foreground object with respect to the image plane. Note that since this kind of distortion is always a stretching, correction requires compressing the foreground image. In this step, we use user assistance to identify a suitable image plane, I_s, for the distorted foreground object (Figure 2). The user provides an estimate of the amount of correction that needs to be achieved using a (scale) parameter s, $s < 1$. Without loss of generality, let us assume that this scale factor is along the horizontal direction. We show in Section III, that an estimate of correction in the y direction can be handled similarly. Let the angle between I and I_s be θ. Note that $s = cos\theta$ (for y-direction $s = sin\theta$) and hence we can find θ from s. Let the corrected foreground be F'.

Since F and F' are both projections of the same 3D object from the same center of projection on two different planes rotated by an angle θ, F' can be found from F using a standard homography H given by

$$H = KR_\theta K^{-1} \tag{1}$$

where R_θ is the rotation of the I by angles θ and K is the intrinsic parameter matrix of the camera given by

$$K = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

where f is the focal length of the camera retrieved from the image meta-data.

Next, we find F' by applying H to F. Then, we remove F from O creating a black hole and paste F' back into this hole. This fills up only part of the hole created by the removal of the segmented foreground and thus leaves some black pixels around the boundary of the foreground object as shown in Figure 3. We denote the image thus created as $U = O - F + F'$.

B. Hole Filling By Mesh-Based Pixel Displacement

In this section, we present a mesh-based method to fill the holes left in U after we replace the distorted foreground with the corrected foreground image. For this we use a mesh-based pixel displacement technique. Our goal is to retain the boundary of the corrected foreground object and smoothly displace the pixels away from this boundary in the background, so that the holes are filled.

To achieve this, we first create a mesh in the original image O denoted by M. The set of vertices in M are given by samples (of the pixels) in the four external edges of

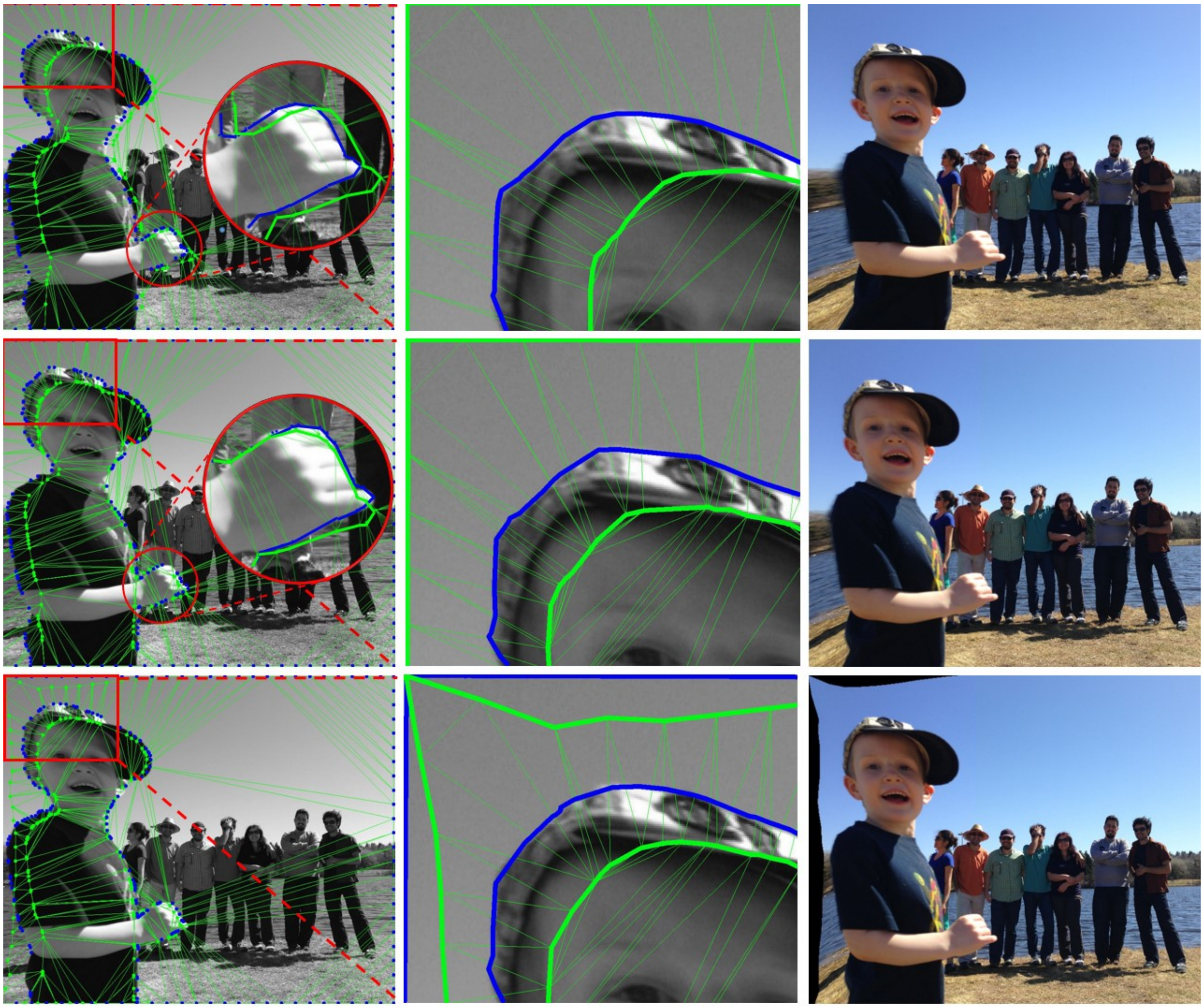

Figure 4: The top row shows the result of correcting the foreground using just a rotation of the image plane. The middle row shows the result of using rotation followed by a translation translation of the foreground. While in these two results, the vertices on the image boundary are not moved, the bottom row shows the result of also moving these vertices also to reduce distortion. In all these figures, the original vertices V are shown in blue, and the transformed vertices V' in green. The transformed mesh M' is also shown in all the figures. Notice that stretching of the background is reduced due to every additional operation.

the image O (image edges) and the boundary b of the foreground/background segmentation (segmentation edges). The vertices on the image edges are denoted by the set V_E and the vertices on b are denoted by V_B. Thus, the set of vertices in M is given by $V = V_E \cup V_B$. Our goal is to displace the vertices in V to create a new mesh such that the holes in U are filled. This can be achieved by just moving the vertices $v_b \in V_B$ lying on b to their corresponding locations on b' given by $H(v_b)$. The set of these new vertices on b' is given by $V_{B'}$. The vertices of new mesh M' thus created is given by $V' = V_E \cup V_{B'}$. We triangulate V using Delaunay triangulation [23] and then use the same connectivity for V'. The pixels within the triangles in M are moved using linear interpolation within the triangle in M'. This is illustrated in the images in the top row of Figure 4.

In the above approach, we notice that while vertices in V_B are all translated almost in the same direction, vertices in V_B that near the center of the image (right part of b' in Figure 3) move into the background region of O, and the vertices away from the center (left part of b' in Figure 3) move into the foreground region. This creates movement of pixels everywhere in the image and creates excessive distortion. To minimize this distortion, we translate b' away from the center of O along the horizontal direction (or the vertical direction, as the case may be), such that the distortion is confined to the far side of b' with minimal distortion on the other side. This translation make the foreground correction from a linear function (homography) to an affine function, and has the effect of changing the origin about which the image plane is rotated. This is shown in Figure 3, and the effect of this operation is reduced distortion in the background as shown in the middle row of Figure 4.

In the above result, since we do not move any vertex in V_E, we still see undesirable stretching of the background. To minimize this distortion, we displace the vertices in V_E in addition to the vertices in V_B. For each $v_e \in V_e$ we find all the vertices V_b adjacent to it, and we denote that set by $A(v_e)$. We move v_e by the average displacement of all the vertices in $A(v_e)$. This makes the size of each triangle in M' to be close to that of the corresponding triangle in M, thereby reducing the undesirable stretching near the foreground, as illustrated in Figure 4. However, this makes the external boundary of the image jagged which can be fixed by a simple cropping. Result of this change to the algorithm is shown in the bottom row of Figure 4.

In any of the above method, since the vertices are moved while the connectivity is maintained, the orientation of the triangle from M to M' can get flipped, as illustrated in Figure 5, resulting in incorrect pixel displacement and texturing. In such cases, the vertices that cross its star (far boundary of the triangles incident on that vertex) is moved back to be inside the star. Since this movement is usually very small - one or two pixels, the effects of this displacement is not noticeable.

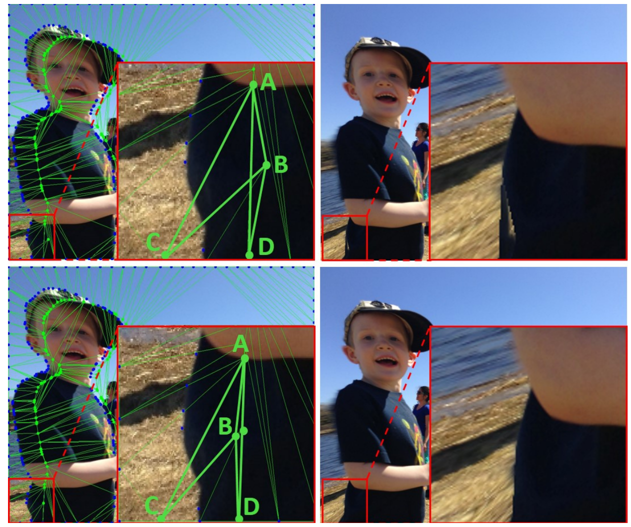

Figure 5: This figure illustrates the effects of the flipped triangle orientation (top) and its effect on the filling of holes. We reposition the vertices of the affected triangles (bottom) to correct this problem.

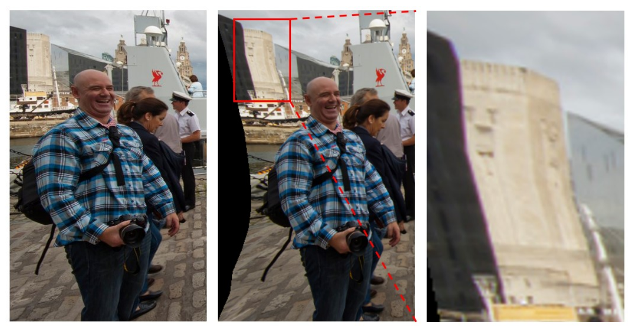
Figure 7: One of the limitation of our mesh warping technique is that the features in the background image such as straight lines may get affected due to foreground correction.

III. Implementation and Results

We implemented our method in Matlab on a machine with 4GB RAM and Core2 Quad CPU and Nvidia GeForce GPU with 896MB Memory. The average time taken to correct the image of size 1000×750 after processing the user input is 0.6 seconds.

Figure 6 shows the results of our method. The bottom row shows how the image plane of the wide angle camera is split into multiple differently oriented image planes to be conducive to different foreground objects. While most pictures with humans needs the image plane to be split in the horizontal direction, some images (e.g. airplane) needs the image plane to be split in the vertical direction.

Limitations and Future Work: This is our first effort to correct distortions common in wide angle lens camera images. In this work we correct the foreground distortion by moving the background pixels. But we do not constrain the background pixels in any way. Hence this can lead to non-uniform stretching of the background which might not preserve geometric features like straight line edges. These artifacts are illustrated in Figure 7. We plan to address this issue in our future work by constraining the movement on the background pixels in a feature-sensitive manner. Further, the issue of flip triangles is common in all mesh based image correction techniques, and its solution is ad-hoc. We plan to find a generic solution to this problem.

IV. Conclusion

In this paper we address the distortion in the images of wide angle lens cameras due to difference in orientation of objects away from the image plane orientation. We first segment the relevant foreground object using an intuitive user interface. Then we undistort the foreground object by rotating and translating the image plane. This creates holes in the background which are then filled up using mesh based image retargeting. Our method can cause some background distortion in the presence of strong features which will

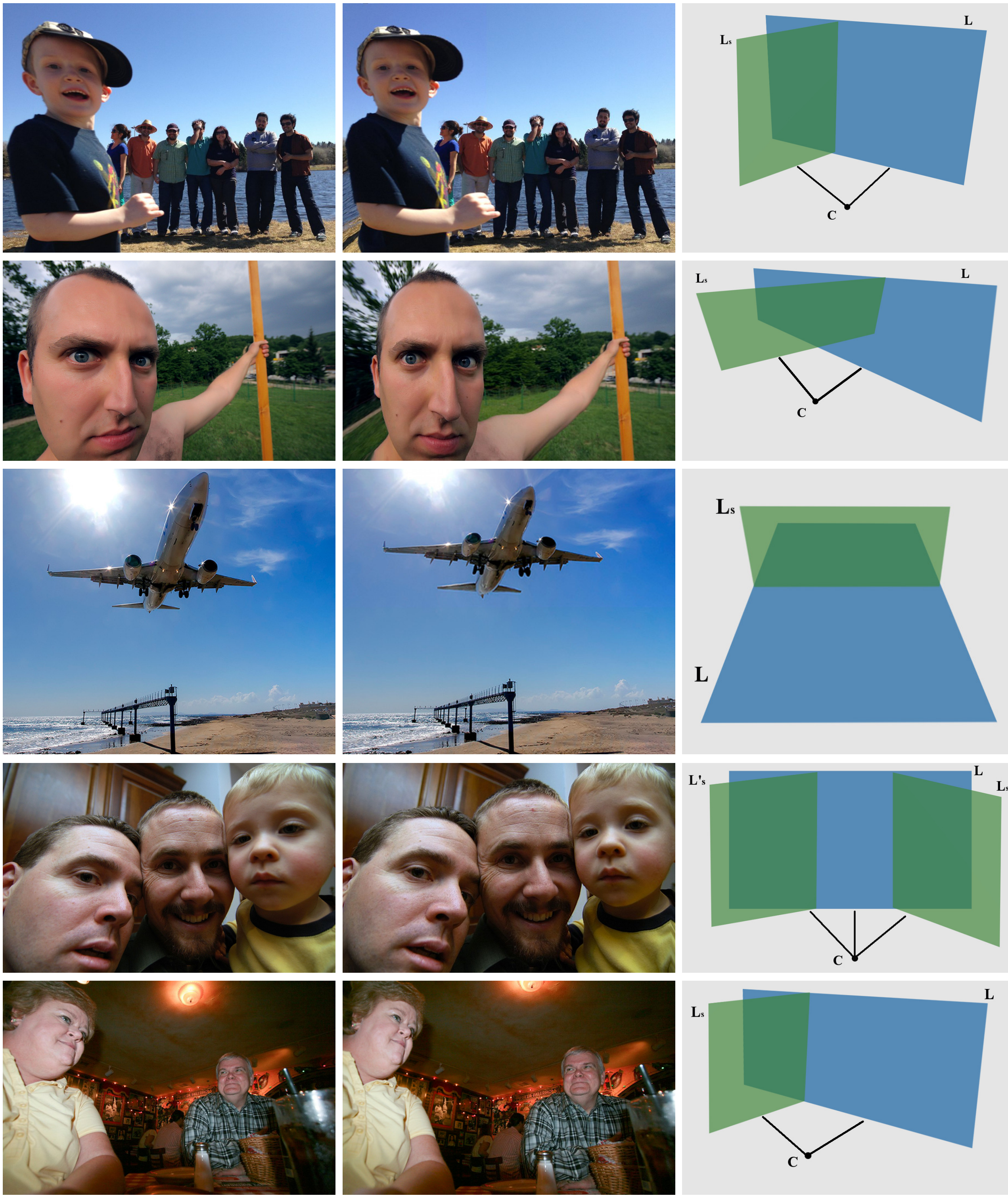

Figure 6: Results of our method. Left Column: Input Images. Middle Column: Corrected Images. Right Column: The changed image planes to achieve the corrections.

motivate our future work in this direction. Further, currently we detect foregrounds using interactive techniques. But for specific foregrounds like faces, we can probably use automated segmentation techniques to extract the face and distance between specific features like eyes to detect the rotation plane. This is something we would also like to explore in the future.

REFERENCES

[1] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, Jul. 2003.

[2] C. Qin, S. Wang, and X. Zhang, "Image editing without color inconsistency using modified poisson equation," in *Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008, pp. 397–401.

[3] A. Majumder and S. Irani, "Perception based contrast enhancement of images," *ACM Transactions on Applied Perception*, 2007.

[4] L. Lucchese, "Correction of geometric lens distortion through image warping," in *Proceedings, International Symposium on Image and Signal Processing and Analysis*, 2003.

[5] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "A hough transform-based method for radial lens distortion correction," in *In Proc. 12th International Conference on Image Analysis and Processing*, 2003.

[6] B. Prescott and G. F. McLean, "Line-based correction of radial lens distortion." *CVGIP: Graphical Model and Image Processing*, vol. 59, no. 1, pp. 39–47, 1997.

[7] L. Ma, Y. Chen, and K. L. Moore, "A new analytical radial distortion model for camera calibration," *CoRR*, vol. cs.CV/0307046, 2003.

[8] ——, "Rational radial distortion models of camera lenses with analytical solution for distortion correction," *I. J. Information Acquisition*, vol. 1, no. 2, pp. 135–147, 2004.

[9] R. Sukthankar, R. Stockton, and M. Mullin, "Smarter presentations: Exploiting homography in cameraprojector systems," *Proceedings of International Conference on Computer Vision (ICCV)*, 2001.

[10] H. Chen, R. Sukthankar, G. Wallace, and K. Li, "Scalable alignment of large-format multi-projector displays using camera homography trees," *Proc. of IEEE Vis*, 2002.

[11] E. Bhasker, P. Sinha, and A. Majumder, "Asynchronous distributed calibration for scalable reconfigurable multi-projector displays," *IEEE Transactions on Visualization and Computer Graphics*, 2006.

[12] P. Roman, M. Lazarov, and A. Majumder, "A scalable distributed paradigm for multi-user interaction with tiled rear projection display walls," *IEEE Transactions on Visualization and Computer Graphics*, 2010.

[13] B. Sajadi and A. Majumder, "Markerless view-independent registration of multiple distorted projectors on vertically extruded surface using a single uncalibrated camera," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[14] ——, "Automatic registration of multiple projectors on swept surfaces," *Proceedings of ACM Virtual Reality and Software Technology*, 2010.

[15] ——, "Automatic registration of multi-projector domes using a single uncalibrated camera," *Computer Graphics Forum*, 2011.

[16] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[17] Y. Guo, F. Liu, J. Shi, Z.-H. Zhou, and M. Gleicher, "Image retargeting using mesh parametrization," *Multimedia, IEEE Transactions on*, vol. 11, no. 5, pp. 856–867, 2009.

[18] D. Vaquero, M. Turk, K. Pulli, M. Tico, and N. Gelfand, "A survey of image retargeting techniques," in *SPIE Applications of Digital Image Processing*, 2010.

[19] S. Cho, H. Choi, Y. Matsushita, and S. Lee, "Image retargeting using importance diffusion," in *Proceedings of the 16th IEEE international conference on Image processing*, 2009, pp. 973–976.

[20] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch, "Automatic image retargeting," in *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, 2005, pp. 59–68.

[21] S. Wang and A. Abdel-Dayem, "Image retargeting using non-uniform scaling with adaptive local search window," in *SPIE International Conference on Graphic and Image Processing*, 2011.

[22] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, 2004.

[23] R. Seide, "Constrained delaunay triangulations and voronoi diagrams with obstacles," Inst. for Information Processing, Graz, Austria, Tech. Rep. 260, 1988.