Perspective Correction of Distorted Projectors with an Uncalibrated Camera

Qian Lin linqian@stanford.edu
Department of Applied Physics

Le Wang lewang2@stanford.edu
Department of Electrical Engineering

Shengtong Chen schen62@stanford.edu
Department of Material Science, Stanford University

Abstract

We demonstrate image processing algorithms that achieve the distortion-corrected projection on surfaces with horizontal discontinuity, with casually placed camera and projector. We focus on three cases of perspective-corrected projection: single projection on flat surface, single projection on wall corner, multiple projections on flat surface. Image pre-warping is used to achieve screen geometry correction. With such algorithm to achieve geometry registration and photometric alignment between projectors, users can project extremely wide field preprocessed images with multiple projectors. It has application in art visualization, head-up displays, and virtual/augmented reality.

1. Introduction

Large-scale displays on unlevel surface are interested in fields such as visualization and virtual reality. Camera-based image processing and computer vision techniques are involved in such applications. This project demonstrate image processing algorithms that allows the users to project extremely wide field distortion-corrected images on screens with horizontal discontinuity. The general work flow of our system consists of camera calibration, distortion estimation, image warping and intensity correction if necessary.

The system is hardware-software integrated. On the hardware side, it involves two casually placed projectors (Epson EX3240), a camera capturing the full projection field, two laptops and one flat or right-angled projection surface. On the software side, all algorithms are implemented on MATLAB, as well as the interface control with the hardware. In addition, A MATLAB graphical user interface (GUI) is created for user-friendly application. Fig. 1 are pictures of our hardware setup.

Such algorithm can be further developed for applications with projectors located at the back of the projection screen, multiple projectors with horizontal discontinuous surface, and simultaneous wide field video projection. With the development of curved displays, algorithm developed in this project can be used for large tiled high-resolution display. Further work need to be conducted for color correction, pincushion and barrel distortion and other optical aberrations introduced by the hardware imperfection. This project has wide application in art visualization, face registration for make-up testing, virtual/augmented reality and head-up displays.

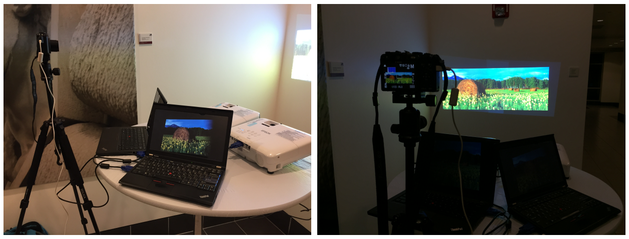


Figure 1. Setup includes a camera used for initial calibration, one or two projectors individually controlled by computer, and a screen.

2. Related Work

There are two major approaches to achieve the geometric alignment of the projectors, parametric approach and nonparametric approach. Variations and details of them are discussed in [7, 2, 3, 4, 1, 6, 5], which are summarized below.

In parametric approach, displayed image is mapped to display surface in a wallpaper fashion. Although the image doesn't look perfectly distortion-corrected from any individual viewing location, as long as the image appears correctly displayed as a wallpaper, the viewers generally have no problem accepting it. This method is used for make-up testing on models and purposes where image distortio, sajadi2010scalablen is corrected for multiple viewing locations.

In nonparametric approach, displayed image is based on the viewer's point of view. The limitation of this approach is that the correction is tied to the viewing location or the camera's point of view. As the viewer walk away from this

”sweet spot”, image may become distorted.

This project is based on the nonparametric approach. Both approaches are widely used for different projection applications. In the corner-projection session of our project, we achieve nonparametric display correction, generating prospectively-correct view at the calibration camera.

For displays with head tracking or eye tracking subdevices, the parametric and nonparametric approaches both have potential applications. In the first case, the rendering subsystem can generate images that appear correct for the viewer's eye location in a wall paper fashion. To generate an immersive 3D wide-field display experience, curved displays are desired. In the second case, updating images based on the viewer head location doesn't require curved displays, but it could introduce visible inconsistencies and distortions if the image updating is not fast enough.

3. Methodology

Details of our implementation are discussed in this section. Three cases of perspective-corrected projection are covered: single projection on flat surface, single projection on wall corner, multiple projections on flat surface.

3.1. Calibration and Feature detection

Point lattice are used to estimate the distortion caused by tilted placement of the projector. A 10x10 white points are spread evenly on an black image in the size of screen resolution. Their coordinates in the image are recorded when they are generated. The perspective transformation caused by distortion could be estimated by the points' original coordinates in the pre-distorted projector screen and their display coordinates in picture taken by camera. We consider each point as a feature, and try to detect and identify them in projector and camera coordinate system.

To detect the coordinates of points, preprocessing are performed on the camera captured lattice image. First, the image is binarized with a threshold that depends on the brightness of the captured image. Then, an erosion and dilation are applied to reduce possible noise. Finally, the center of each white plate are recorded for each captured image.

To identify a point, we use binary encoding. Any N distinguishable points could be uniquely identified using a binary code with $\log_2(N+1)$ number of bits. A dot array is created for each bit, but only points with 1 on that bit is shown in that dot array. A hundred points could be uniquely encoded using a seven-bit binary number. As shown in Fig. 2. This way, each point can be identified based on its appearance in the set of dot arrays.

3.2. Single Projector on Flat Surface: Keystone Correction

Using the coordinate of the projected dots and their detected camera coordinate (Fig. 3), a homography H_{pc} from projection to camera coordinate can be calculated.

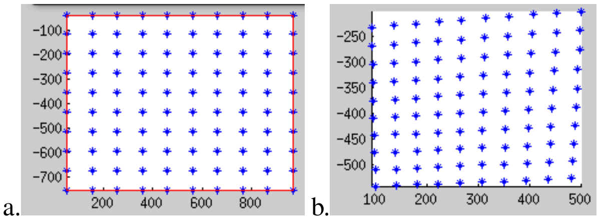


Figure 3. Calibration grid in (a) projector coordinate and (b) measured camera coordinate.

Using H_{pc} and the projector bounding box $B_p = [1 : dx, 1 : dy]$ where $dx = 1024$ and $dy = 768$ is the default resolution of the projector, the bounding box camera coordinate can be calculated using $B_c = H_{pc} * B_p$. Given B_c, and maximal inscription rectangle of aspect ratio $4 : 3$ can be calculated, whose camera coordinate is R_c. The projector coordinate of this inscription rectangle is $R_p = H_{pc}^{-1} * R_c$. The bounding box and image box is shown in Fig. 4.

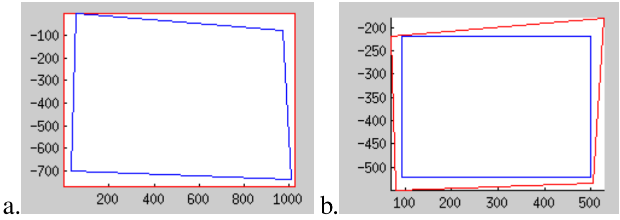


Figure 4. Bounding box (red) and image box (blue) in (a) projector and (b) camera coordinate. The inscription rectangle in camera coordinate [blue box in (b)] is horizontally aligned and has aspect ratio $4 : 3$.

A homography H from the projector bounding box B_p to the projector image box R_p can be calculated. Apply image warping with H produce the output image. Projecting the output image gives an aligned image viewed from the camera, shown in Fig. 5.

Matlab code relevant to this part:

```
[orig_coor, cap_coor]= grid_read('../Data/Grid1/',...
  1:7, '.JPG', threshold);
Hpc=calibrate(orig_coor,cap_coor);
H=getWarp(Hpc,dx,dy);
Iin=imread(fullfile(pathname, filename));
Iout=warpImage(Iin,H,dx,dy);
imwrite(Iout,'flat_output.jpg');
```

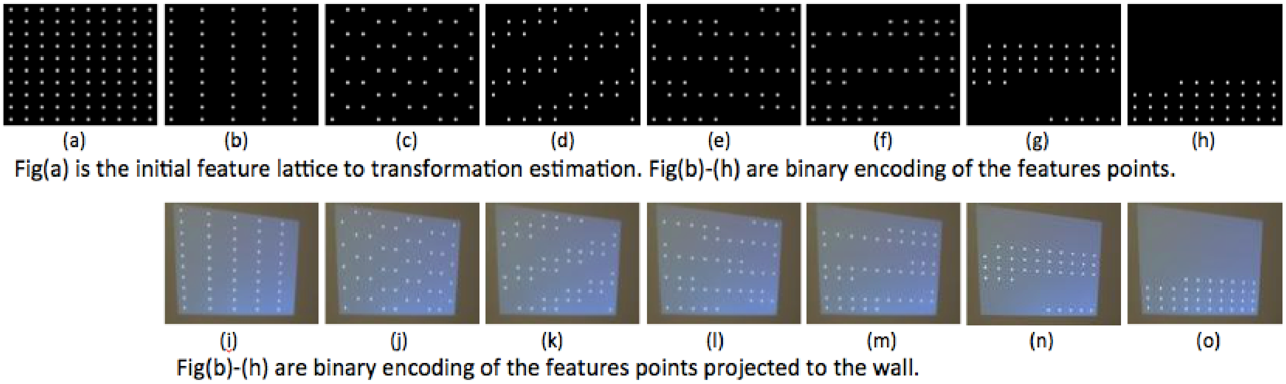

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
Fig(a) is the initial feature lattice to transformation estimation. Fig(b)-(h) are binary encoding of the features points.
(i)
(j)
(k)
(l)
(m)
(n)
(o)
Fig(b)-(h) are binary encoding of the features points projected to the wall.

Figure 2. Original and Camera Captured Point Lattice

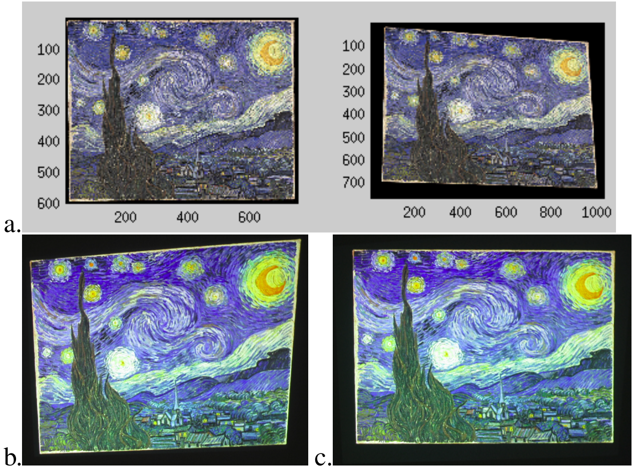


Figure 5. (a) original and output image. Bottom: image viewed from the camera, (b) origin and (c) corrected.

3.3. Single Projector on Vertical Corner

The same calibration process is used to obtain the grid coordinates, as shown in Fig. 6. When projecting onto a corner, the left and right part of the image has different perspective transformation. We use a RANSAC-like method to estimate the homography $H_{pc1(2)}$ from projector coordinate to camera coordinate for the left and right side, label the points as left or right, and then calculate the average homography using all the points classified as left/right.

Using $H_{pc1(2)}$ one can obtained the bounding box of projected area (red and blue box in Fig. 7b), and the largest inscription rectangle of aspect ratio $4 : 3$ (black box in Fig. 7b). Using inverse $H_{pc1(2)}$ to transform the image box back to projector coordinates (red and blue box in Fig. 7a), we obtain the corrected image contour for left and right side.

The homography $H_{1(2)}$ from left and right side image boxes to bounding box in projector coordinate can be calculated. Apply $H_{1(2)}$ for image warping produces the output image to project on the left and right side. Combining them using binary mask at the corner (intersection line of the two image boxes) produces the final output. Projecting the output image leads to a perspectively-corrected projection viewed from the camera, shown in Fig. 8.

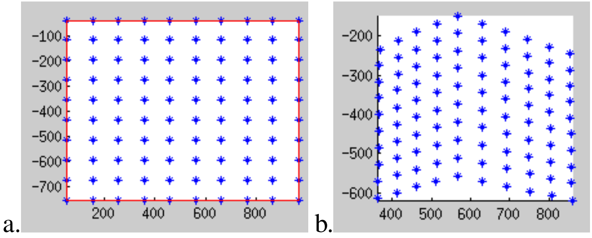


Figure 6. Calibration grid in (a) projector coordinate and (b) measured camera coordinate.

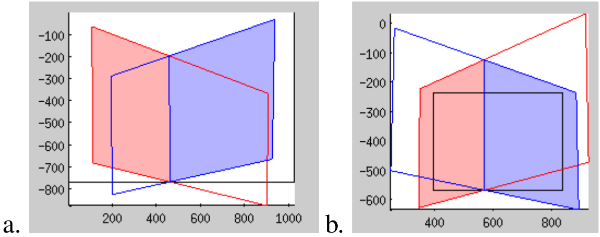


Figure 7. (a) Bounding box (black) and image box (red and blue) in projector coordinate. (b) Bounding box (red and blue) and image box (black) in camera coordinate. The image box in camera coordinate is horizontally aligned and has aspect ratio $4 : 3$.

Matlab code relevant to this part:

```
[orig_coor, cap_coor]= grid_read('../Data/Grid2/',...
  1:7, '.JPG', threshold);
[H1,H2]=corner_calibrate(orig_coor,cap_coor);
[Hw1,Hw2,mask1,mask2]=corner_getWarp(H1,H2,dx,dy);
Iin=imread(fullfile(pathname, filename));
```

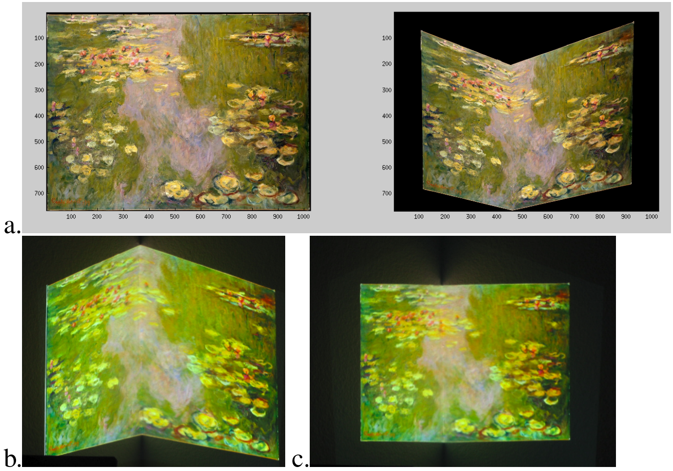


Figure 8. Top: original image and output image (to be projected). Bottom: image viewed from the camera, origin (left) and output (right).

```
Iout=corner_warpImage(Iin,Hw1,Hw2,mask1,mask2);
imshow(Iout);
imwrite(Iout, 'output_corner.jpg');
```

3.4. Multiprojectors Panorama on Flat Surface

To display panorama on flat surface using two projectors, the algorithm takes matched points in projector coordinates and camera coordinates for two projectors as inputs, and generates a perspectively corrected image for each projector such that the two projectors could together display an undistorted panorama image. It requires some overlapping between the two projector displays, and we only support horizontal panorama yet.

Using the same method described in keystone correction, a bounding box for projector display is computed in the camera coordinates (Fig.9). Based on the combined projection region in the camera coordinates, the maximal rectangle region for panorama display could be defined (Fig.10). With the image for display fills in the maximal region (Fig.11), the portion for each projector could be cropped out. Since the overlapped region of the projectors will have twice of lightning, a linear fading is implemented to compute a γ-corrected intensity weight for each projector. Finally, each projector has its portion of image warped to its coordinate system, using the method described in keystone correction (Fig.12 Fig.13).

4. Experiment and Result

A major part of this project is building a real-time demo for our implementations. In this section, details of the system setup and operation are discussed.

One Canon G9 camera with a standard tripod is used. In

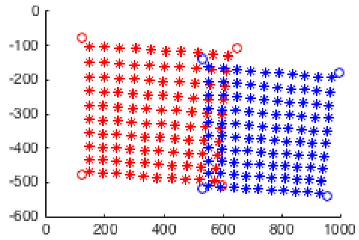


Figure 9. Point lattice and individual boundary in joint projection. * for lattice points, and o for boundary corner points. Red for projector 1, and blue for projector 2.

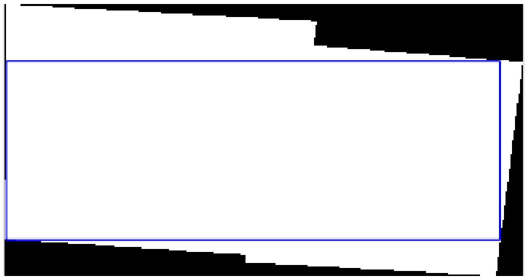


Figure 10. Detected maximal rectangle region in the joint projection. The projectable region is in white. The blue bounding box defines the maximal region

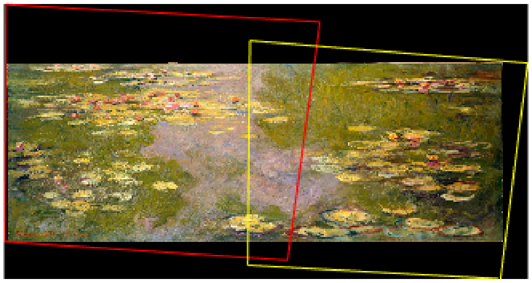


Figure 11. Projected image in the join display. Red box for the boundary of projector 1, while yellow box for the boundary of projector 2

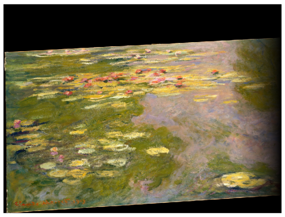


Figure 12. Image on Projector 1

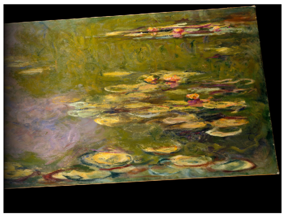


Figure 13. Image on Projector 2

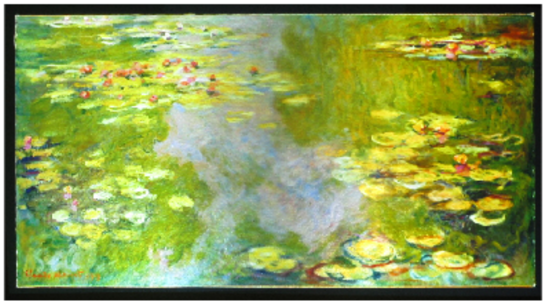


Figure 14. Camera picture of the joint projection result

the experiment, Canon G9 camera ISO is set to 80. Such setting is tested for proper dot array detection under the standard room lighting. Depending on illumination condition during projector registration, especially for projection room with strong lighting, users should change the camera ISO value for proper dot recognition or the threshold value for dot array detection in calibration step.

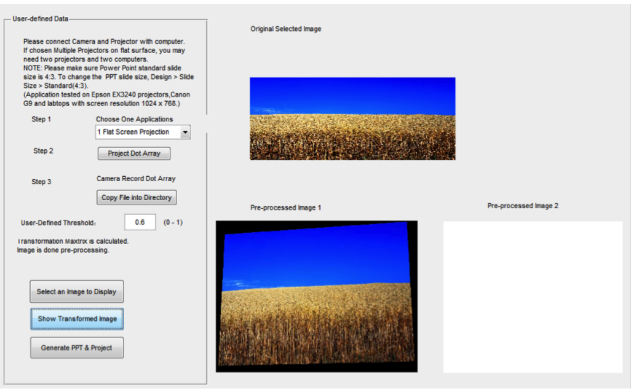
Two Epson EX3240 projectors are used. Their built-in automatic distortion correction are disabled. Two laptops are connected to control them. The screen resolution of both computers is set to 1024:768, to match the projector display region. The resolution can be setup to match other projectors.

The algorithm implementation, a user interface, and hardware controller are built in MATLAB. The implementation also utilizes functions from vl_feat [9]. MATLAB GUI is constructed to guide users through the process of projection application selection, dot array projection and capture, user-defined threshold, image transform and Power Point generation (Fig.15,16,17). In this GUI, automation is achieved by matching the time elapses of the Power Point image automation time with that of the camera. This way once the shutter is pressed, camera will automatically capture the set of 7 dot arrays.

For the single projector and flat surface projection application, the algorithms corrects for perspective distortion, commonly known as keystone correction. Fisheye and other non-linear lens effect is not taken into account.

For the single projector and corner projection application, the left and right side area needs to be at relatively evenly distributed (at least 25% of the image to be on either side of the corner).

For the multiple projector and flat surface application, two projectors are individually registered. It is recommended to have them positioned roughly at the same distance from the projection screen, and with at least 5% overlapping area in multiprojector case. When users register one projector, the shutter of the other projector needs to be closed. As long as the projectors are set to display the same screen resolution areas, the multi-projection code can be used to display horizontally wide field of view panorama images.



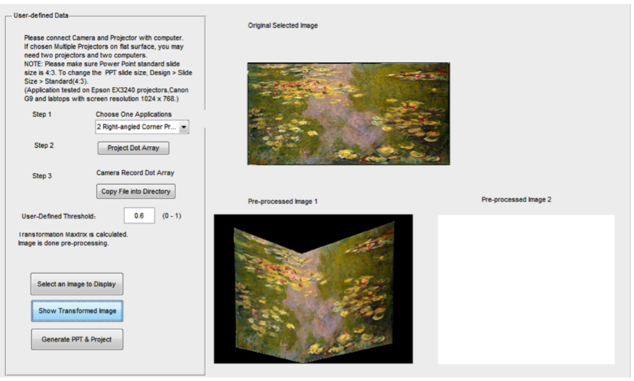
Please connect Camera and Projector with computer.
If chosen Multiple Projectors on flat surface, you may need two projectors and two computers.
NOTE: Please make sure Power Point standard slide size is 4:3. To change the PPT slide size, Design > Slide Size > Standard(4:3).
(Application tested on Epson EX3240 projectors,Canon G9 and labtops with screen resolution 1024 x 768.)

Step 1 Choose One Applications: 1 Flat Screen Projection

User-Defined Threshold: 0.6 (0 - 1)

Transformation Maxtrix is calculated.
Image is done pre-processing.

Figure 15. GUI for flat screen projection application.



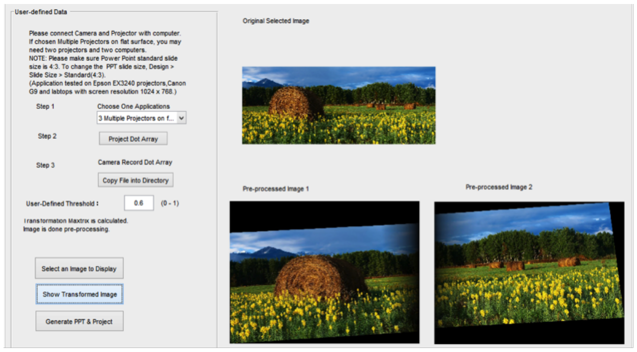
Please connect Camera and Projector with computer.
If chosen Multiple Projectors on flat surface, you may need two projectors and two computers.
NOTE: Please make sure Power Point standard slide size is 4:3. To change the PPT slide size, Design > Slide Size > Standard(4:3).
(Application tested on Epson EX3240 projectors,Canon G9 and labtops with screen resolution 1024 x 768.)

Step 1 Choose One Applications: 2 Right-angled Corner Pr...

User-Defined Threshold: 0.6 (0 - 1)

Transformation Maxtrix is calculated.
Image is done pre-processing.

Figure 16. GUI for right-angled corner projection application.



Please connect Camera and Projector with computer.
If chosen Multiple Projectors on flat surface, you may need two projectors and two computers.
NOTE: Please make sure Power Point standard slide size is 4:3. To change the PPT slide size, Design > Slide Size > Standard(4:3).
(Application tested on Epson EX3240 projectors,Canon G9 and labtops with screen resolution 1024 x 768.)

Step 1 Choose One Applications: 3 Multiple Projectors on f...

User-Defined Threshold : 0.6 (0 - 1)

Transformation Maxtrix is calculated.
Image is done pre-processing.

Figure 17. GUI for multiple projectors on flat surface application.

Fig. 18. 19, 20, and 21 are camera captured results of projection adjusted by our system. As seen from those images, our algorithms achieve perspective correction for projectors at the camera's view.

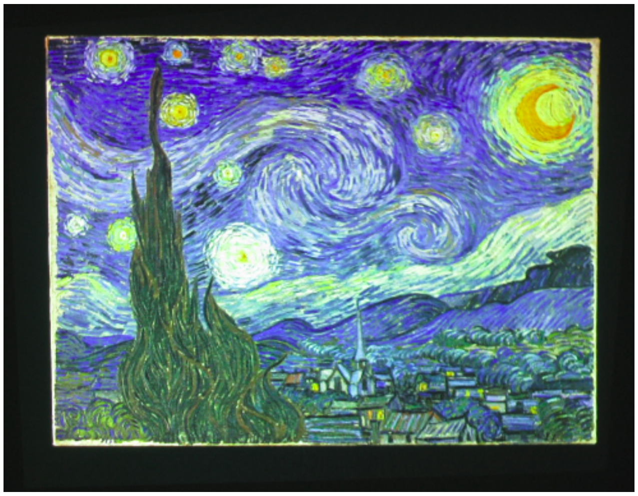


Figure 18. Corrected Single Projection on a Flat Surface

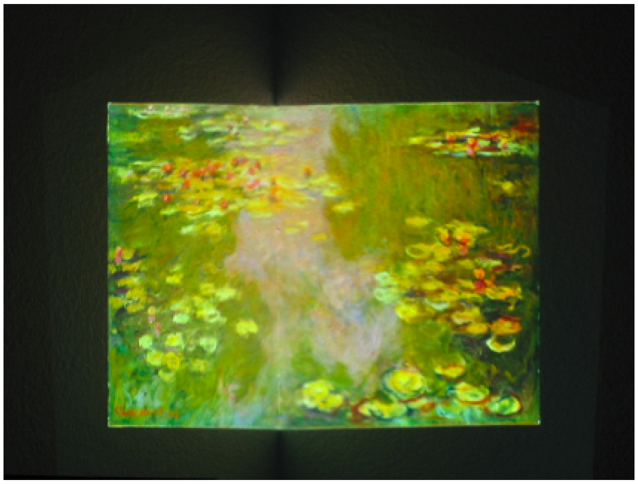


Figure 19. Corrected Single Projection at a Right-Angled Corner

5. Discussion and Future Work

Although the software and hardware automation is achieved in GUI, the project have some limitations. One major issue is that the processed image is based on the non-parametric geometric distortion correction. In other words, the processed image is only corrected for the camera sensor location. For practical applications, this location should be the viewer eye position. If the viewer walks away from this ?sweet spot?, a headeye tracking system is needed to detect the new camera location. Based on the simultaneous detection, the distortion-corrected images need to be updated. Many promising follow-up work should be done for more mature automation and for wider projection applications:

- The automation of hardware and software is semi-manual due to the lack of resources. With a scientific camera controlled by the MATLAB Image Acquisition toolbox, better automatic projector registration, including automatic camera capture with automatic dot array projection, can be achieved.

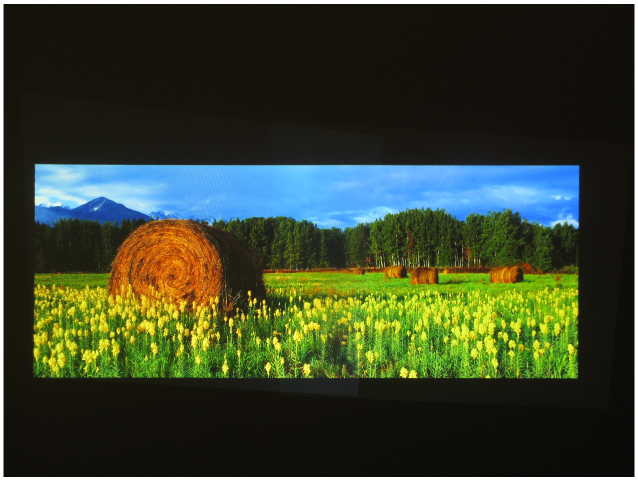


Figure 20. Corrected Multi Projections on a Flat Surface

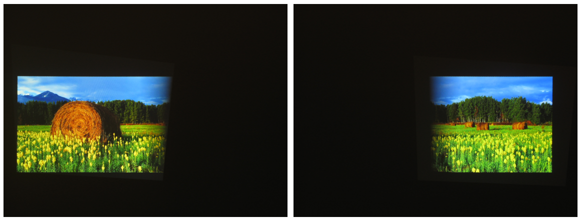


Figure 21. Corrected Multi Projections on a Flat Surface With Only Left Projector Display or Right Projector Display

- Assuming the projector lens has no aberration, we can directly use the image instead of the dot arrays for the calibration for the geometric registration purpose. But the benefit of using dot arrays is that we can further correct optical radial distortion (pincushion/barrel) with the detected dot arrays.
- This project is designed and tested for the set-up where projectors are located in the front of the projection screen. With simple modification of the code, users may extend the applications to cases with projectors located at the back of the projection screen.
- This project uses two identical projectors so there is no color difference between the projectors. However we do notice the difference between image color appearing in the monitor and the projected image color. With a color meter, users may further implement color correction.
- This project only demonstrates the multi-projection

with two projectors due to the limitation of resources. Based on the same method, users can implement the code to many projectors. In that case users may need one or multiple wide field of view cameras to capture the dot arrays.

- This project only demonstrates the image preprocess for corner projection, due to the limitation of resources. With algorithm optimization and development, users can apply part of the code to the horizontally-curved projection surfaces.

- This project demonstrates the projector registration method for surfaces only with horizontal discontinuity. Further research and testing needs to be done for projection surfaces with both vertical and horizontal discontinuities.

References

[1] A. Agarwal, C. Jawahar, and P. Narayanan. A survey of planar homography estimation techniques. *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.

[2] M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multiprojector displays. *Visualization and Computer Graphics, IEEE Transactions on*, 11(2):193–206, 2005.

[3] A. Majumder and M. S. Brown. *Practical multi-projector display design*. AK Peters, Ltd., 2007.

[4] R. Raskar and P. Beardsley. A self-correcting projector. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–504. IEEE, 2001.

[5] P. Roman, M. Lazarov, and A. Majumder. A scalable distributed paradigm for multi-user interaction with tiled rear projection display walls. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1623–1632, 2010.

[6] B. Sajadi, M. Lazarov, M. Gopi, and A. Majumder. Color seamlessness in multi-projector displays using constrained gamut morphing. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1317–1326, 2009.

[7] B. Sajadi and A. Majumder. Markerless view-independent registration of multiple distorted projectors on extruded surfaces using an uncalibrated camera. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1307–1316, 2009.

[8] B. Sajadi and A. Majumder. Scalable multi-view registration for multi-projector displays on vertically extruded surfaces. In *Computer Graphics Forum*, volume 29, pages 1063–1072. Wiley Online Library, 2010.

[9] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.