ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse163942020.pdf
https://doi.org/10.30534/ijatcse/2020/163942020

Auto-fit two movable projections into a single 3D display screen for virtual and augmented reality scenes

Zineb Rebbani [1], Driss Azougagh [2], Ahmed Rebbani [3], Omar Bouattane [4]
[1] SSDIA Laboratory, ENSET, Hassan II University of Casablanca, Morocco, z.rebbani@gmail.com
[2] SSDIA Laboratory, ENSET, Hassan II University of Casablanca, Morocco, azougagh@gmail.com,
[3] SSDIA Laboratory, ENSET, Hassan II University of Casablanca, Morocco, a.rebbani@gmail.com,
[4] SSDIA Laboratory, ENSET, Hassan II University of Casablanca, Morocco, o.bouattane@gmail.com

ABSTRACT

High quality virtual and/or augmented scenes using (multiple) mobile/robot projectors requires precise auto-positioning and auto-correcting of projectors and their corresponding frames' buffers, respectively, in order to auto-fit projections into a single planar 3D display screen. The ideal projector camera system, using a stationary calibrated camera as a feedback, needs to automatically real-time calculate correction from previously captured camera frames and estimate all transformations to recommend for each projector and/or to apply on each projector's 2D frame. The pixels of projections need to geometrically coincide so that the displayed 3D images will always be aligned and centered on the screen without experiencing any kind of distortion or degradation.

In our previous works, auto-fitting movable projections into one display screen and auto-guiding a movable projector were studied and investigated. The auto-correction is obtained first by measuring the actual coordinates of the projectors and finding the maximum intersection rectangle of the participating mobile/robot projectors, and second either by calculating the transformation matrix to mechanically correct positions of projectors toward the right location and orientation or basically auto-adjusting each 2D video frame in graphic card buffers to match the maximum intersection rectangle. However, noticeable shape distortion (quadrilateral representation) and the color (and/or light intensity) degradation of projected rectangular pixels in the projection extremity were experienced.

This work extends and complements our previous work to intelligently correct the projector location and it corresponding projection center and auto-adjust frames in graphic card buffers in order to eliminate, hide or reduce the degree of shape distortion and the level of color degradation of projected pixels. Promising results and observation were disclosed and exploited in the simulated testing system that mostly adopts only linear methods while keeping the overall execution time of auto-correcting and displaying a frame less than the time between displaying two consecutive 2D frames.

Key words : Auto-position, auto-correct, auto-fit, projectors.

1. INTRODUCTION

In future, with the hi-tech era, mobile projectors for 2D or 3D scenes will take place in many applications [1]–[3]. Nowadays, the electric display play an import role in society [4]–[6] and a lot of 3D display techniques have emerged in recent years [7]. The projector-camera systems use feedback cameras to capture projected images and extract information in order to real-time auto-calibrate projectors for any required correction as in [8]–[10]. To ensure a high-quality performance for projector-type 3D display techniques, the calibration and correction are necessary to make the output images of all the projectors coincide well on the display screen [8]–[13]. Other techniques, such as [14]–[15], include geometric as well as photometric frames correction of two (or multiple) projectors for high-resolution projections. Until this date, many solutions have been introduced for systems with stationary projectors and only fewer works studied and analyzed systems with mobile projectors. Creating 3D scenes using two (or multiple) mobile projectors suffer from synchronization in timing as well as calibration in space (screen surface) more than using stationary projectors.

In our work [1], we provided a mathematical expression that calculate the robot projector's position using only the four corners of the captured projection frame by the camera from a planar screen. Having the projector's position (and the four corners), we calculate transformation matrix used by the robot projector to guide itself. In another previous work [2] of ours, we extended the mentioned approach to auto-guide not only one but two (or multiple) mobile projectors to project into one target display screen. This is achieved by auto-fitting movable projections to at least cover the critical zone for tolerable video watching. We also mainly adopt only linear deterministic solutions instead of iterative converging ones to reduce the time of processing. However, noticeable shape distortion (quadrilateral representation) and the color (and/or light intensity) degradation of projected rectangular pixels in the projection extremity were clearly experienced.

In this work, we extend and complement our previous works to intelligently correct the projector location and it corresponding projection center and auto-adjust frames in graphic card buffers in order to eliminate, hide or reduce the degree of shape distortion and the level of color degradation of projected pixels. Moving robots to the suggested location and adjusting and modifying shape and color of images in memory buffer will give better improvement in the presence of projectors with high luminosity and resolution that play scenes with high resolution images. Our main target is to correct the following factors while minimizing the real time calculation; the displacement of the components of the projection system, the alignment of the 3D virtual/augmented reality scenes based on the captured shape distortion (quadrilateral representation) and the sensed color (and/or light intensity) degradation of every particular projected rectangle pixel.

In the next section we start with an introduction of the components involved. In the third section, we introduce the projector positioning approach followed by a section dealing with the projection modification and adjustment. The fifth section shows results and analysis. And finally, we conclude with perspective remarks.

2. COMPONENTS

2.1 Hardware

Our 3D virtual/Augmented reality set is composed of a pair of raspberry pi 2, a pair of movable projectors, a camera used for feedback purpose, a pair of polarizing filters each is placed, orthogonally to each other, up against the lens of each projector, a linearly polarized eye glass containing a pair of orthogonally polarized filters in a parallel orientation to that of the pair placed up against the lenses and a 3D display screen.

The 3D display screen is placed on the middle of the upper part of the wall where the *xy*-plane lies for coordinates as shown in Figure 1 Therefore, the camera must capture not only the display planar screen but also an extra margin of the wall to cover for the miss projected frames when projectors are moving.

In the setup, each projector is mounted into its corresponding arm bracket lift or robot with six degree of freedom of movement. Both arms/robots need to be guarded near to the camera in a closed space such as class rooms, theaters and vehicles. This solution can be applicable in environments suffering from frequent vibrations like military/commercial transportation such as cars, touristic buses, train planes, ferries and car/flight simulators.

2.2 Software

The software is based on the Linux operating system and most of the tests and simulations were conducted using C language for programming and modifying the open source media player VLC in raspberry. The two raspberry pi devices communicate through GPIO pins by direct manipulation of GPIO registers.
Simulation was developed to off-line emulate the detection of corners of projected images and calculate the corresponding corrections in a raspberry pi. A simulation software is introduced to study and analyze the behavior of the projection from different location in the space.

3. PROJECTOR POSITIONING

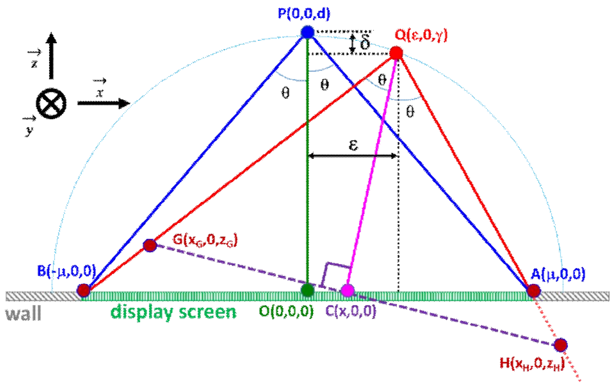

Figure 1: Geometry of a projection covering the screen display

In order to keep movable projectors' beams well exploited to cover the target display screen, Figure 1 shows the *xz*-relation between the displacement coordinates (ε,δ) from the ideal position P and the movable projector Q projecting to the center C.

In the figure, it can be noticed that angle of projection 2θ is the same for P and Q. The center of projection keeps shifting with the direction to where the projection is headed to in a non-linear manner. A function relationship between ε and δ is shown in (1).

$$(d-\delta)^2+\varepsilon^2+\frac{\eta^2\beta}{\alpha}\pm 2\sqrt{\lambda\eta^2(\alpha\varepsilon^2+\eta^2)}=0 \tag{1}$$
$$\lambda=(\frac{1-f^2}{1+f^2})^2, \alpha=\lambda-1, \beta=\lambda+1, and \quad \eta = fd$$

Without loose of generality, we choose to set focal lenses factor to 1. Therefore, we can notice that λ = 0, α = -1 and β = 1 and can be substituted in equation 1 to obtain a simplified expression in (2) of a circle with a center P and a radius $\eta = fd$.

$$(d-\delta)^2 + \varepsilon^2 = \eta^2 \qquad (2)$$

The projector orientation need to be adjusted to point to the projection center C(x,0,0) in order to cover the screen along the x-axes where the value of x depends on the displacement coordinates center ε and δ as in (3).

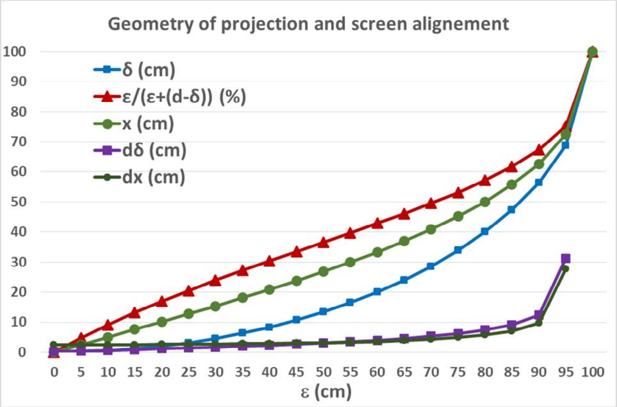

Figure 2: Parameters correlation of projection/screen alignment

$$x = \frac{k-j}{k+j}\eta \quad where: \begin{cases} k = \sqrt{(\eta+\varepsilon)^2 + (d-\delta)^2} \\ j = \sqrt{(\eta-\varepsilon)^2 + (d-\delta)^2} \end{cases} \qquad (3)$$

In order to see the relation that links the parameters x and δ with respect to ε, Figure 2 depicts their corresponding functions for f=1 and d=100. The difference dx increases as the value of ε increases. The circular shape can be observed when checking the draw of δ in function of ε.

This finding can be extended to the two axes x and y leading to a projector Q moving on the surface of a sphere centered at the origin O and have a radius d, where, without loose of generality, both focal lenses' factors (f_x and f_y) are set to 1. When these factors are different the projector Q need to be on the surface of a shape similar to an ellipsoid to covers the inner ellipse of the display screen.

The display screen can be covered from which ever position the projector Q moves to as long as it stays on the surface of the candidate surface and projecting toward the center C that can be calculated from the coordinates of Q. Each projector far from the safe surface might be mechanically guided to move toward the closest point in the surface while heading projecting the corresponding center C.

4. PROJECTION MODIFICATION AND ADJUSTING

The only remaining issue to study is the texture of the projected frames on the display screen regardless of the location of the projectors. We introduce software solutions

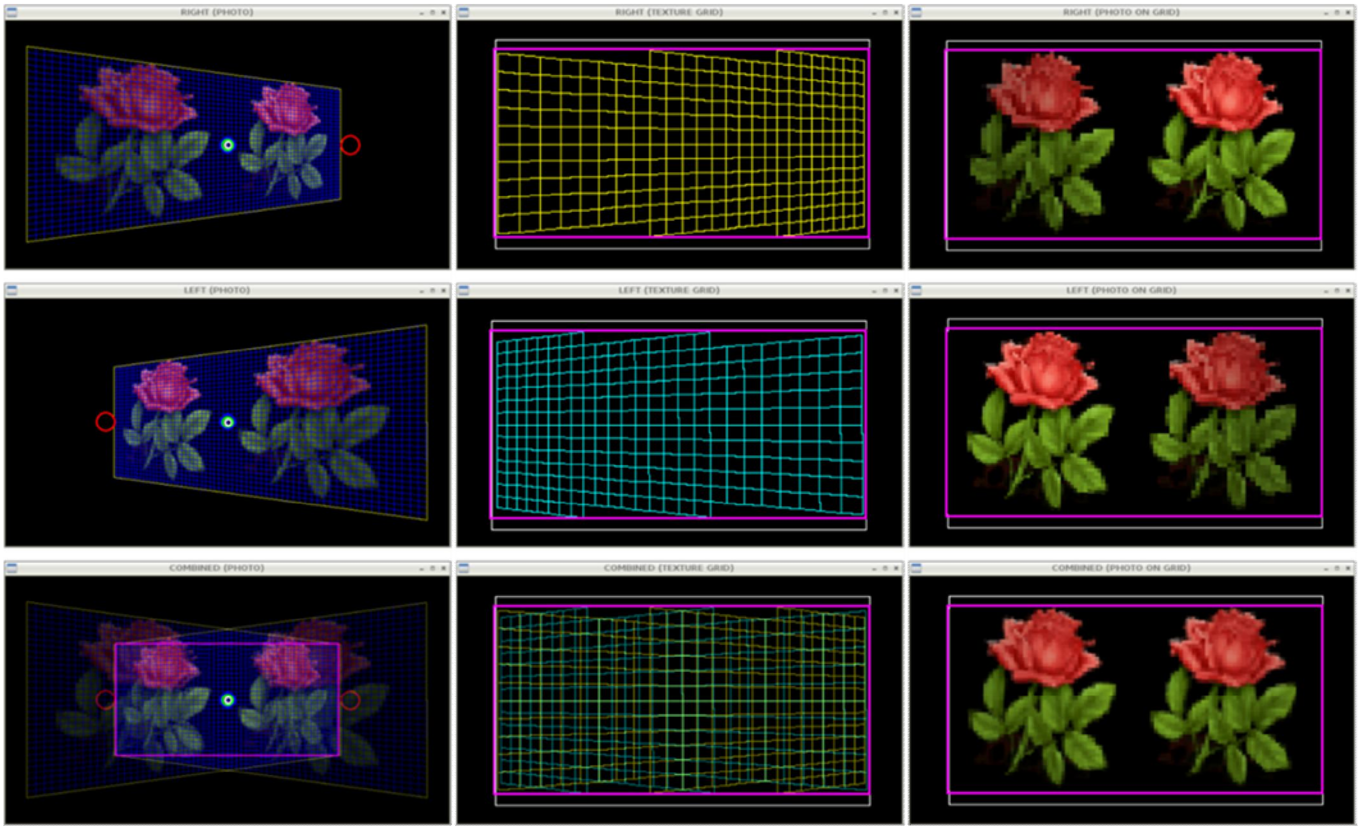

Figure 3: Apply only geometry modification on images in memory

instead of mechanical ones. In order to tackle such an issue, we implemented a simulation tool that illustrate the details of the projection texture of pixels all along the screen display depending on where the projector Q is standing in the space.

Figure 3 shows nine windows where rows are designated for LEFT, RIGHT and COMBINED projections, and the columns for direct wall, grid texture and adjusted/corrected projections. The combinations in the third row is witnessed when both first and second rows of the corresponding projectors are aiming to the same display screen. It can be noticed that the combined projection in the third row in the first column cannot be tolerated to be considered as a normal/regular scene that audiences can bear.

The small red circles in the windows of the first column represent the locations of the projectors in the space proportional to the window. A zoomed grid textures of pixels inside the pink rectangle, the intersection zone with width 2×w and height 2×h, of both RIGHT and LEFT projectors and their combinations are as shown in the second column. There is nine indices (u_i, v_j) in the memory buffer can be calculated using the inverse matrix projection of the nine points (i×w,j×h) of the intersection zone for i and j in {-1,0,1}. Using all these indices are sufficient to adjustably map a real image to the memory encircled by the four corners (u_{-1}, v_{-1}), (u_{-1}, v_1), (u_1, v_1) and (u_1, v_{-1}) so that the outcome projection will be evenly printed in the intersection zone.

Each quadrilateral in the grid textures corresponds to the projection of n×m pixels of a rectangle zone in an image. For the RIGHT projection, the quadrilaterals on the right side of the texture are smaller in size and have higher resolution (concentration of pixels) than the quadrilateral on left side. Due to symmetry, the LEFT projector also suffers from lower resolution quality on it right side instead. Combining both texture gives a mismatch to some degree of their quadrilaterals and as the distance between both projectors get further far from each other the miss match increases. The mismatch also might become more sever when the n×m is smaller than a level where the severity leads to the image recovery failure.

When a part of the projection is located far from the projector position in comparison to the other parts, the corresponding quadrilateral has its color intensity, shape and surface become degraded, distorted and bigger, respectively. In order to resolve this issue, zones with higher resolution need to be reduced in their light intensity and the parts with lower resolution need to be elevated to keep a homogeneous distribution of colors proportional to the original image according to some color factor γ as in (4). The factor γ is a function proportional to the distance (|QM|) between a projection point M and the projector Q. In this simulation, we choose to use 20% of luminosity reduction and 70% of elevation for images with n×m=2×2 pixels per quadrilateral.

$$(r, g, b) = (\gamma \times r_{image} + \gamma \times g_{image} + \gamma \times b_{image}) \tag{4}$$

Using the software approach, we first find the frame geometry of any given projection and estimate the projector location in the space based on the work in [1]. Also, we calculate the rectangle intersection that has the center at the origin O as explained in [2]. Then, we can get the geometry of both grid textures and map them with their corresponding RAM memory buffer. And finally, we apply modifications to the images in the memory so that each quadrilateral in the grid texture is corrected. The third column shows the resulting images and their combinations when applying the modifications without taking into consideration the color degradation for n×m=2×2 pixels per quadrilateral. For RIGHT projector and due to an uneven grid texture, the left side of the projected image has some zigzagging pattern unlike the right side. The combination of both projections seams to recover from the distortion experienced due to the locations of both projection. However, this recovery happens only due to location symmetry around the origin of both projectors.

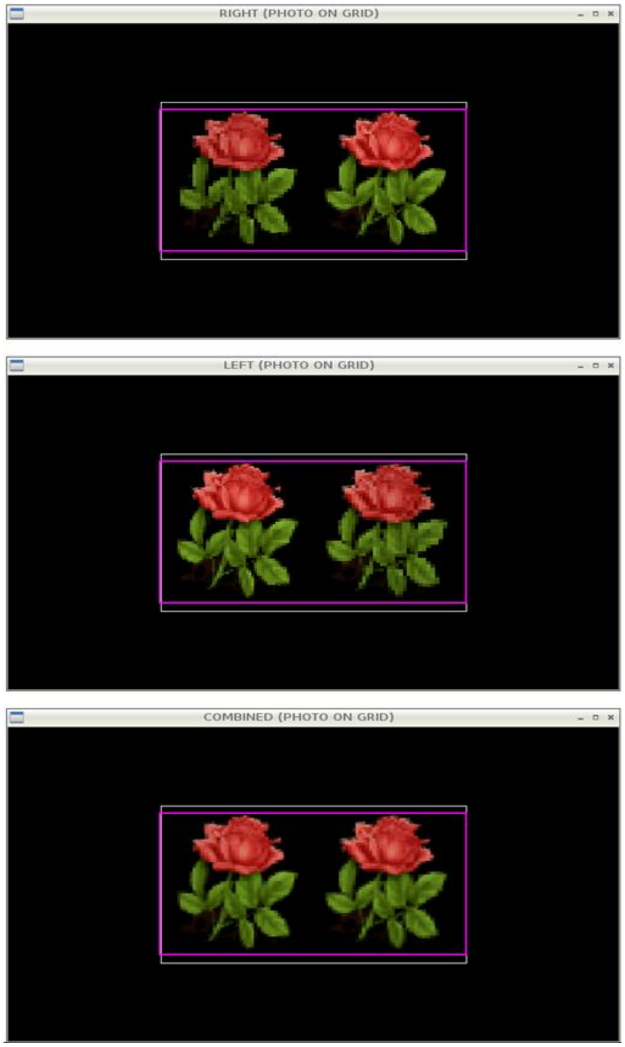

Figure 4: Apply geometry and color modification on images in memory

Figure 4 shows the application of color correction in order to achieve a homogenous distribution of colors for the LEFT and RIGHT projections. In 3D scenes images with even color distribution, the audiences experience less stress on their eyes. This application is important for high quality 3D scenes as well as for multiple dispersed projectors in a large theater. With the applied correction, the left/right side of the RIGHT/LEFT projection is improved regarding their color homogeneity but still share the same shape deformation degree as in the previous figure. The combined projection is improved slightly. Since this paper aims in improving the 3D virtual and augmented reality, the interest is focused mainly on improving the RIGHT/LEFT projection and COMBINATION is optionally used to guarantee the alignment of both projection.

The number $n \times m$ of pixels in a quadrilateral and the function factor γ can be either hardware and/or software dependent. For instance, the projector resolution might limit the best value of $n \times m$ to use. And, the quality of the projector lenses might affect the function factor γ to apply.

5. DISCUSSION AND ANALYSIS

For realistic simulation, as a robot can freely move, we experimented several case where the position of the projectors are selected without having symmetry of their location. Figure 5 shows an example of a complicated projection. The RIGHT and LEFT projections are not symmetric to each other.

In the figure, references are introduced to show the coordinates of $P(x_p,y_p,z_p)$ and $C(x_c,y_c,0)$ corresponding to red and green indicator circles. The blue indicator circle gives the proportion of the z parameters of P starting from the left side of the window. Therefore P represents the coordinates of the projector in the space. C represents the coordinates of the spot in the display screen where the projector P is aiming to.

The LEFT and RIGHT projections' frames are not aligned with the display screen as well as with each other. The experienced combined projection is more distorted than the one shown above. The quadrilaterals in both grid textures have a higher mismatch and require more care when adjusting and correcting the images in the memory buffer. The larger the quadrilateral we have the worst resolution experienced in the corrected images in the third column. The coloring adjustment was applied as described above. Overall, the combination image shows a tolerable alignment and color homogeneity between the two projections. The total execution time of geometry computing and image adjusting/modifying in the memory buffer varies depending on the position and orientation of both projectors. The time ranges between 5ms and 15ms with an image size of 640x480 without using HDMI acceleration.

The software takes many factors into consideration when auto-synchronizing and mutualizing both projectors to match the same projection spot. Given the center of the 3D display

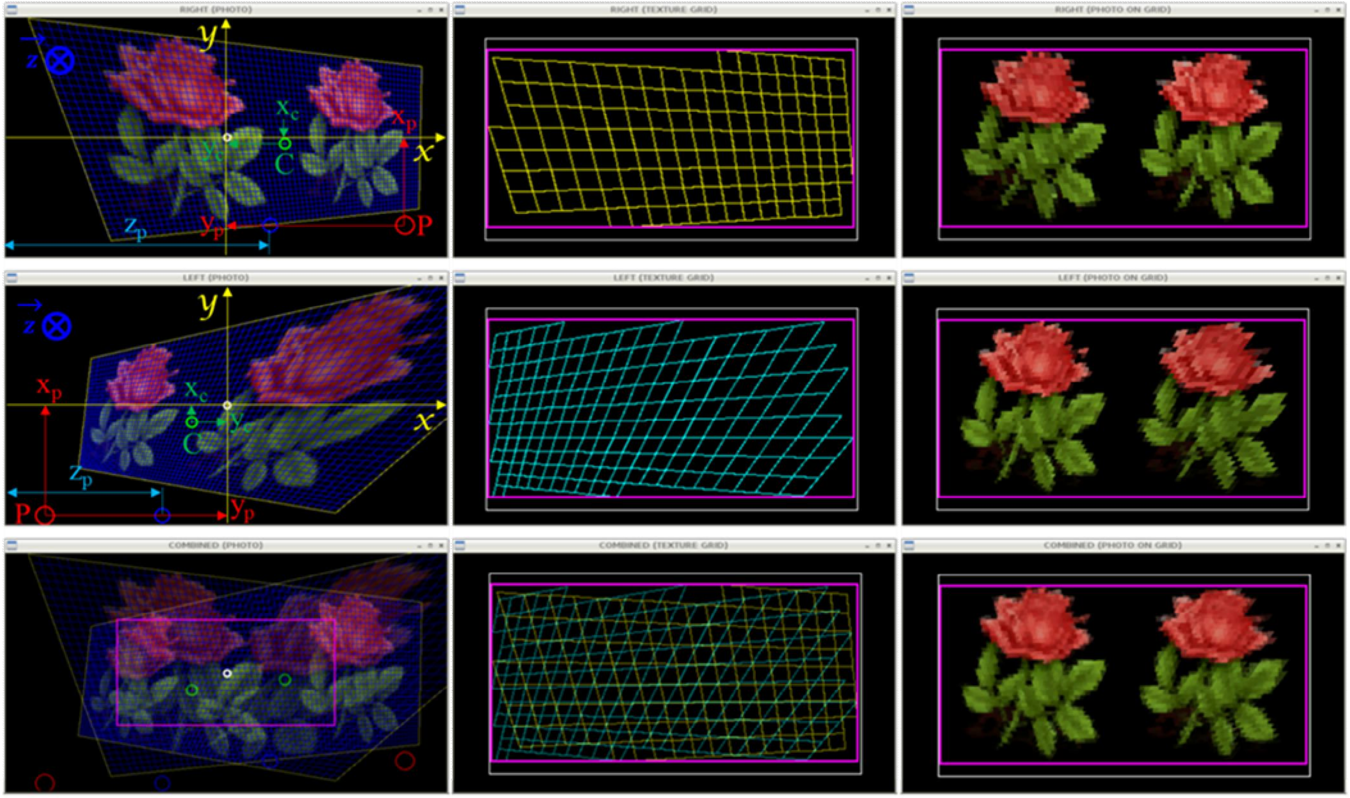

Figure 5: Apply geometry and color modification on complex projections

screen and the resolution rapport (a,b) extracted from the played video (such as 4:3 or 16:9), the common zone shared by both misplaced projectors can be defined as the maximum rectangle inside both projections, centered at the center of the 3D display screen and respecting the extracted resolution rapport (a,b). Even when applying the above proposed adjustment and modifications, the projection of the video needs to be paused or stopped when either the intersection rectangle is small, the distorted parts of the image is large and/or the color distortion is high to some level where watching the video is not tolerable.

Moving robots to the suggested location and adjusting and modifying shape and color of images in memory buffer will give better improvements in the presence of projectors with high luminosity. These limits need to be investigated, configured and fixed to some ratio τ proportional to the size of the 3D display screen where the common zone is required to occupy at least τ proportion of the total display screen.

6. CONCLUSION

This work demonstrates major factors needed to be taken by the mobile projectors for 3D virtual and augmented reality. These factors are summarized into three major ones; the correct projector location and it corresponding projection center, the quadrilateral representation of a distorted rectangle of pixels, and the color (and or luminosity) degradation of the image.

For the projection location we introduced the mathematical formula to find the closest point where the robot needs to head to and the center of projection where it should aim to. Unlike the first factors, the other two are persistent and can only be changed if the robot moves toward a different location. The ideal location is when facing the center of the display screen.

Facing the center of the display screen might not be practical due to the other roles the robot might be engaged to fulfill. Depending on the number of pixels used for each quadrilateral we reshaped the pixels in the memory buffer to adjust the projected image. Then, we redistributed the colors of the image so that the distorted and the intense areas get evened in their luminosity distribution. The overall results was promising and the overall execution time (5~15ms) are mostly consumed by the raw writing of a frame into a memory buffer.

REFERENCES

1. Z. Rebbani, D. Azougagh, A. Rebbani, H. Bahatti, O.Bouattane. **Auto guiding a mobile projector**, 12th International Conference Intelligent Systems: Theories and Applications (SITA'18), October 2018, doi: 10.1145/3289402.3289503.
2. D. Azougagh, Z.Rebbani, H. Bahatti, A. Rebbani, O. Bouattane. **Auto-fit multiple movable projections into one display screen**, International Conference on Wireless Technologies, Embedded and Intelligent Systems, April 2019, doi: 10.1109/WITS.2019.8723667.
3. H. Cabibil, V. Lopez, T. Obero, P. Delima. **Museodabawenyo: an interactive virtual reality museum application using unity**, International Journal of Advanced Trends in Computer Science and Engineering, May 2020, Vol.9, No.3, doi: org/10.30534/ijatcse/2020/161932020
4. C. Zoido, J. Maroto, G. Romero, J. Felez. **Tiled projector displays correction for dark scenes in railway simulators**, Fifth European Symposium on Computer Modelling and Simulation, November 2011, pp. 317-322, doi: 10.1109/EMS.2011.69.
5. SC. Mallam, S. Nazir, SK. Renganayagalu. **Rethinking maritime education, training, and operations in the digital era: applications for emerging immersive technologies**, Journal of Marine Science and Engineering vol. 7, no. 428, December 2019, doi: 10.3390/jmse7120428.
6. B. Sajadi, M. Lazarov, A. Majumder, M. Gopi. **Color seamlessness in multi-projector displays using constrained gamut morphing**, IEEE Transactions on Visualization and Computer graphics, Vol. 15, No. 6, pp. 1317-1326, December 2009, doi: 10.1109/TVCG.2009.124.
7. R. Desierto, A. Reciña,T. Arroyo,P. Delima. **GoonAR: a bilingual children storybook through augmented reality technology using unity with vuforia framework**, International Journal of Advanced Trends in Computer Science and Engineering, Vol.9, No.3, May 2020, doi: org/10.30534/ijatcse/2020/180932020
8. J. Deglint, A. Cameron, C. Scharfenberger, H. Sekkati, M. Lamm, et al. **Auto-calibration of a projector-camera stereo system for projection mapping**, Journal of the Society for Information Display, August 2016, doi: 10.1002/jsid.464.
9. CQ. Song, J. Cheng. **A robust projector-camera interactive display system based on finger touch control by fusing finger and its shadow**, Journal of The Society for Information Display, Vol. 25, October 2017, doi: 10.1002/jsid.608.
10. M. Brown, A. Majumder and R. Yang, "**Camera-based calibration techniques for seamless multiprojector displays**," in IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 2, pp. 193-206, March-April 2005, doi: 10.1109/TVCG.2005.27.
11. J. Zhou, L. Wang, A. Akbarzadeh, and R. Yang. **Multi-projector display with continuous self-calibration**, in Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems, 2008, doi: acm.org/10.1145/1394622.1394626.

12. O. Bimber, A. Emmerling. **Multifocal projection: A multiprojector technique for increasing focal depth**, IEEE Transactions on Visulization and Computer Graphics, Vol. 12, No. 4, August 2006, doi: 10.1109/TVCG.2006.75.
13. T. Okatani and K. Deguchi. **Easy calibration of a multi-projector display system**, Vol. 85, No. 1, October 2009, pp. 1–18, doi: 10.1007/s11263-009-0242-0.
14. C. Zoido, J. Maroto, G. Romero, and J. Felez. **Optimized methods for multi-projector display correction**, International Journal on Interactive Design and Manufacturing (IJIDeM), vol. 7, no. 1, pp. 13–25, February 2013, doi: 10.1007/s12008-012-0161-0.
15. M. Harville, B. Culbertson, I. Sobel, D. Gelb, A. Fitzhugh, et al. **Practical methods for geometric and photometric correction of tiled projector**, in 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), June 2006, pp. 5–5