A Personalised Stereoscopic 3D Gallery with Virtual Reality Technology on Smartphone

Abstract—**Virtual reality (VR) is becoming more and more popular thanks to the recent advance of smartphones and low-cost VR headsets. In this article; we propose an online system that creates a stereoscopic 3D gallery for individual use. The system allows the users to upload two photos of an interested object (e.g. a personal collectable artefact) acquired by their smartphone. The photos are needed to be close to a stereo pair: i.e. the camera is slightly moved/translated to the right before acquiring the second shot. The uploaded images are automatically rectified into an epipolar stereo pair (photos will be horizontally aligned) after several corresponding points were detected. A simple stereo matching algorithm is applied to identify the average disparity range between the two photos, to generate the most comfortable VR viewable stereoscopic pictures. The system then builds a gallery with the collected photos to display 3D visualisation on VR devices such as Google Cardboard, the Samsung Gear VR or Google Daydream. This system is low-cost, portable, simple to set up and operate. With such system, Internet users all over the World could easily visualise and share their collectable items in 3D; which are believed to be useful for VR and social media community.**

Index Terms—**Computer Vision, Stereo Vision, Virtual Reality, Online, Internet.**

I. Introduction

Colour and depth signals deliver visual information of the World to us. Our brain continuously receives visual cues (from the two eyes) to rebuild a spatial 3D structure of the surrounding effortlessly. In other words, it is natural for people to see things in 3D. This enables humans to discover the appearance, shape, and distance of distinct objects on their surroundings. The stereo vision system of animals including humans has been evolving for millions of years and is evidenced to be an essential factor for survival. There are many advantages to the human vision; the ability to perceive distance is considered as an important factor. Even the human depth estimation can be made from analysing the perspectives of objects and their shadows. From our experience, the best and fastest way is to solve the correspondence problem of stereo vision. It determines the patterns viewed from the left eye corresponding to which viewed from the right eye, to allocate the same points or regions [1]. For instance, the separation between two correspondences of the same point from the two views defines how close or how far from us that point is in space.

A. 3D Visualisation with VR Devices

In recent years, the increasing popularity of 3D cinemas and 3D TVs have attracted attention from the public to the science of this two-eyed depth perception. There has been a dramatic expansion in the number of 3D display devices, movies, and games with 3D content in the consumer market. In principle, they attempt to deliver two different views, one to each eye; and force the brain to recover the desired 3D scene. For instance, 3D movies are simply made by two side-by-side video cameras. They are placed mimicking the arrangement of human eyes when observing scenes through two perspectives, which are horizontally separated by a small distance. Similarly, this principle also applies to the LCD screens of Virtual Reality (VR) systems. When people view these with special eye-wears, the 3D illusion of a stereoscopic scene still appears. The human brain attempts to determine the same scene points in 3D based on the similarities between left and right viewed points [1]. It is scientifically known as the binocular vision or stereo vision system. VR employs this knowledge to bring 3D perception to individuals home and office. Today, there are many commodity VR systems such as Google Cardboard, Samsung Gear VR. Underneath such systems are one flat LCD screen that project two stereo images to the human left and right eyes synchronously and simultaneously.

B. The Power of todays Smart-phones

Together with the World Wide Web, mobile devices and their applications have also grown exponentially. In the last ten years, there has been an unprecedented evolution of cell-phones. The devices have been dramatically improved, from the outside to the inside, from pricing to usability and connectivity. Also, most newly-released smart-phones have built-in Wi-Fi, 3G, GPS, high-resolution screens, and multiple cameras, which enables efficient localisation data acquisition, data transfer, and visualisation. They have become “could not live without” devices for many people. Moreover, many cameras and cell-phones today are manufactured with more than one camera sensor: one at the front and one at the back. The current mainstream VR consists of two types of VR headsets: dedicated-hardware type, and smart-phone-based type. While the former offers superior experience thanks to a custom head-mounted display and a custom input device, the premium price-tag has prevented them from becoming ubiquitous. Smart-phone-based VR is designed to make use of the smartphone display, processor, and sensor, thus reducing the cost of the VR headset to as low as a couple of dollars when excluding the smartphone cost. One popular example is Google Cardboard, which only requires a cardboard-made VR headset to experience VR through the smartphone. The platform has been shipped over 10 million units in March 2017 [2]. With the advantages of low-cost and wide availability, Smart-phone-based VR has a vital role of helping people around the world create and enjoy their own 3D contents. However, to acquire 3D photos, it is necessary to have a system of two side-by-side cameras, which is still relatively rare on the market.

C. Research Goal

Our ultimate goal is to fill this gap by bringing the latest results of VR technology and Computational Stereo Vision techniques to *the public domain* (general users) via one of the most flexible platforms – the Internet. We build an interactive public system and its network architecture that allow users to build a personal online stereoscopic 3D gallery using only their smart-phone. The users can navigate and visualise the gallery using one of the low-cost VR devices such as the Google Cardboard. The users only need to acquire two consecutive shots of the personal artefact (the cameras position for the second snapshot is slightly translated horizontally to the right of the first one). The result is equivalent to stereoscopic images taken by conventional 3D cameras; because we provide automatic alignment of the image pair. From a practical aspect, this system presents some benefits over other comparable products:

- It works online, and pictures from phones can be uploaded directly.
- It provides stereo image rectification (horizontal alignment) using uncalibrated methods.
- It runs a simple stereo matching algorithms to estimate the disparity range of two images, to reconstruct the most comfortable visualised VR image pairs.
- It graphically renders VR scene on smartphones to allow users to view and navigate data sets.
- Its VR applications for Apple iPhones and Android phones are to be released.

II. System Design and Implementation

We propose a system that allows the users to capture 3D photos, and share their collection to other users through the internet as well. The system combines human-computer interaction with local, and global systems collaboration. There are two main modules of the system that can be seen in Figure 1: Client Side, and Online Server Side.

A. Client-Server Architecture of the Interaction

The entire interaction and communication of the system were built upon the foundation of a Client-Server architecture [3]. This structure model defines two roles for computers within a network: Client and Server. The Client is the user's smart-phone while the Server is a powerful machine which solves problems requested by Clients. The overall interactions between Servers and Clients are demonstrated in Figure 1, they are separated and can only communicate with each other via the Internet. The most complex and cumbersome works are located on the powerful server with a fast connection. The system is therefore capable of handling more flexible data and from a wider range of inputs. Furthermore, a distinct advantage of this architecture is its excellent portability, which can be performed safely on any web browsing clients without the installation and execution of third party software. The proposed system is fast, lightweight, and feasible to work on a broad range of mobiles devices including smartphones and tablets over a 3G, 4G or Wifi networks.

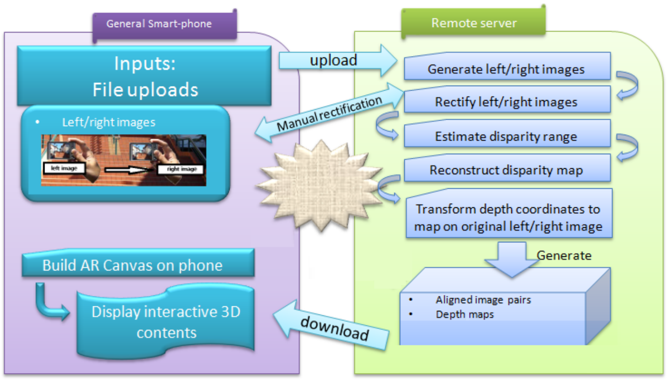


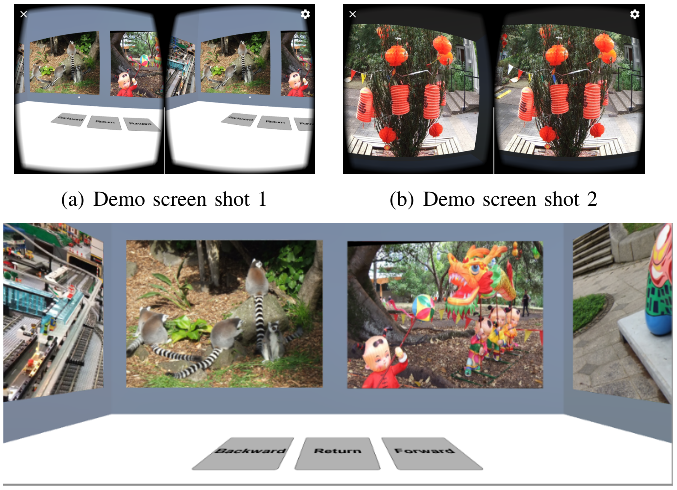
Fig. 1. Basic processing steps: user uploads smart-phone images to a remote server and receives a 3D description of the depicted scene.

B. Client Side - User Interface

The system client provides a user-interface as well as the system presentation for the users. The private, and portability nature of this project encourages us to focus on developing a mobile app as the system client. The app consists of two main features: 3D photos capturing, and 3D photos gallery. Each feature is separated from another. They have different user-interface and can be chosen when the system client starts.

In 3D photo capturing feature, there are two options for the users to take a stereoscopic photo. The first option is using a stereoscopic lens attached to their phone camera. With this approach, it is only necessary for the users to take a single photo to produce a stereoscopic photo. This is because the stereoscopic conversion has been done through the lens. The second option is using their phone camera without additional hardware. This option requires the users to take two different photos of the same object. Each photo represents a different angle of the object. In 3D photo gallery feature, the users can choose to either view their photo collection or input a private key to see the gallery from other users. The user-interface of this feature has a design based on a virtual museum. With the help of Unity 3D engine and VR technology; the users can explore a 3D area where all of their or other users 3D photos are placed around them.

Some screenshots of the client-side can be seen in Figure 2. The first two are side-by-side views of the phone screen to be observed by a standard VR headset. These scenes are in 3D photo gallery feature. The last one is the image of the demo in Unity 3D developer's mode. Initially, there will be a prompt that asks the users to either view their gallery or input a private key to see a gallery from the other users. Many pictures are hanging on the wall surrounding the users in such a way that they are in a gallery in real life. On the floor, there are three buttons: "Backward", "Forward" are used to navigate next and



(a) Demo screen shot 1

(b) Demo screen shot 2

(c) Demo screen shot 3

Fig. 2. Screen-shots of our current Android application

last collection, and “Return” is used to go back to the feature selection scene. (Figure 2(c)).

C. Data Acquisition and Processing

When the system client is installed, it will automatically create a unique private key. This private key is used to help the online server identify the ownership of each photo it receives or in its online database. This key is also used as a way for the users to share their photo gallery to the other. The system client can send any private key that the users input to the online server. Based on this key, the online server will push all of the detailed stereoscopic photos in its database to the network client that has just sent the key. When the users produce any photo in the system client, depend on the picture capturing option, the online server will perform a suitable task. If the picture is already in stereoscopic format, it will be uploaded straight to the online database. If there are two separated pictures of the same object, the online system will perform image calculation to convert them into a single stereoscopic photo of the object, then upload it to the online database.

D. Server-Side Components

The server-side is the location where specific client requests are processed. Except for the client component, the other three server components are all on this side. They are the web server, the database server, and the processing servers. The three components work together to return results to the appropriate users as quickly and securely as possible. The Apache web server manages the web location www.ivs.auckland.ac.nz and its sub-domains. It directly receives requests and delivers content to all Internet users via a user agent tool such as a mobile-phone application. In our system, the functions of this server are to receive image contents and commands from users, then send information to the MySQL server and deliver tasks to processing servers. It also collects results and returns them to users. To achieve this, a sequence of tasks is executed on the server machine, called server-side actions. They are implemented with a specific server-side scripting language, in our case, Hypertext Preprocessor scripting language (PHP). To initiate a process, a minimum of two images are required to be posted to the server. The server script contains all the necessary commands used to launch the entire server-side process of the 3D content extraction. Processing servers are computers which handle the most complex tasks. They are computers connected to the network but have special permission to access the web-server’s resources (hard-drives).

III. Image Processing on User Inputs

The server-side system automatically extracts or reconstructs left/right stereo images and disparity range from user uploaded data. Stereo reconstruction is used to estimate a dense disparity map from a stereo pair within a chosen disparity range. The user’s inputs are two consecutive photos from single conventional cameras.d In particular, both the left and right images of a stereo pair can be obtained either directly or indirectly and the depth information can be extracted. The obtained left and right images are converted into an epipolar horizontally aligned stereo pair, and the anticipated disparity range for stereo matching is estimated.

Here, a stereo-like pair is made by two consecutive captures of a static scene as shown in Figure 3 (top-left image) by any conventional camera from two different positions. The apparent benefit of this stereo acquisition is cost efficiency as no specific stereo camera is required. Another advantage is the ability to adjust the stereo baseline length between two views. However, such sequential acquisition has some drawbacks. The inconsistent sharpness, quality, and intensity between the two captures can be named in particular. Achieving an entirely static scene is a challenging task. Also, no matter how carefully the images have been acquired, the acquired left and right images are still misaligned.

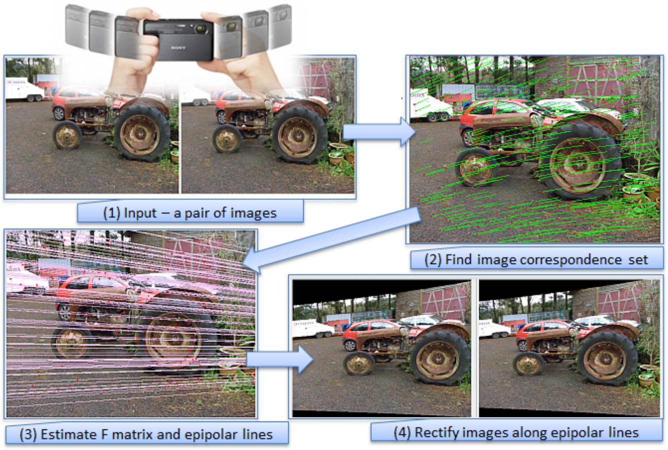

(1) Input – a pair of images
(2) Find image correspondence set
(3) Estimate F matrix and epipolar lines
(4) Rectify images along epipolar lines

Fig. 3. The process of stereo image alignment.

Assuming that both photos are of equivalent quality, let us concentrate only on horizontal image re-alignment by an uncalibrated stereo image rectification. Stereo image rectification projects two stereo images onto a common image plane and does not need the camera calibration process. It is a well-studied technique to make a left-right stereo pair with canonical epipolar stereo geometry [4] (the epipolar geometry). In the epipolar stereo images, every pair of corresponding image points is located on the conjugate horizontal scan lines. In other words, the stereo image rectification turns the unaligned images (Figure 3, top-left) into a well-aligned stereo image pair (Figure 3, bottom-right) as if a stereo camera produced it.

In this application, the camera's focal lengths, internal settings, and image quality are assumed to vary slightly between the two snapshots. The rectification must be carried out without the prior knowledge of the camera parameters. Currently, uncalibrated rectification is often accomplished by estimating a fundamental matrix introduced by Luong and Faugeras [5]. The estimation is based on a set of known correspondences between the left and right images of a near-stereo pair. This 3×3 matrix of rank 2 combines the column 3-vectors $\mathbf{p}_l$ and $\mathbf{p}_r$ of homogeneous coordinates of the corresponding 2D points in the left and right images respectively as follows:

$$\mathbf{p}_r^\top F \mathbf{p}_l = 0 \tag{1}$$

If the matrix F is determined, an image pair can be re-sampled to conform to the standard stereo geometry. The rectification process moves the actual epipoles to the horizontal position at infinity. The rectification process implemented in our system is discussed next.

A. Image Rectification from the Fundamental Matrix

The image rectification pipeline in our system and as accepted in various robust methods for estimating the fundamental matrix [6], consists of four procedural steps:

1) **Feature point detection** is used to select informative pixels (such as corners) from both images.
2) **Correspondence matching** is the process which decides corresponding pairs between the two sets of feature points.
3) **Robust estimation of the fundamental matrix** uses robust regression techniques to achieve a 3×3 matrix that best fits all the points in the epipolar constraint function.
4) **Image rectification** generates horizontally aligned epipolar stereo pair.

Different approaches can be applied at each step, to obtain the desired outputs. For instance, there are at least three comparable algorithms that can be deployed in the correspondence matching process as seen in Figure 4. Our task is to figure out a suitable approach at each step to achieve the positive outcomes under the short waiting time, i.e. to determine the best path in Figure 4. In this section, we illustrate some experimental results which help us choose specific methods to be used. In Section III-B, we evaluate three tracking algorithms: (i) the Lucas-Kanade optical flow [7] in a pyramid, which matches good features to track (GFT) points [8], namely the KLT method; (ii) the Speeded Up Robust Features tracker (SURF) [9]; and (iii) the Scale-Invariant Feature transform tracker (SIFT) [10]. Followed by Section III-C, where we demonstrate other experiments to determine the appropriateness of the two robust regression algorithms – RANSAC [11] and LMedS [12] which obtain the best-fitting fundamental matrix.

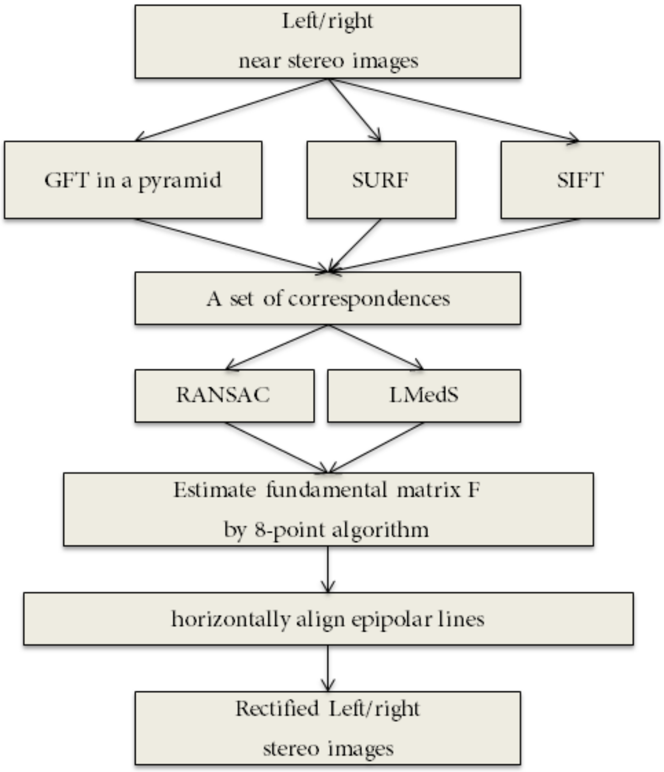


Fig. 4. Possible processes of the uncalibrated stereo image rectification.

B. Evaluation of KLT, SIFT, and SURF

In stereo image rectification, the first crucial task is to obtain a relatively large set of correspondences between left and right images. These correspondences can be used in a robust regression algorithm for the fundamental matrix estimation. This image processing technique is often called feature or salient point detection. It is widely used for tracking [13], stereo matching [14], object recognition [15], and so on. In general, the system initially finds a set of points on a reference image, then it matches them to another set of correspondences in another picture.

One of the most simple methods is Kanade-Lucas-Tomasi (KLT) feature-tracker. KLT is based on Shi and Tomasi's good feature to track (GFT) [8] and Lucas-Kanade optical flow in a pyramid [7] of Lucas et al. KLT is fast and relatively reliable. However, it may return many outliers in some cases. To enhance this correspondence matching process, we tried two of the more advanced ones: the Scale-Invariant Feature transform (SIFT) by Lowe et al. [10] or the Speeded Up Robust Features (SURF) by Bay et al. [9]. Both of them are among the most favourable and popularly used feature trackers [16] nowadays. Both are relatively robust in various situations where there are large scale changes and affine transformation of features between the two images. Consequently, it was difficult to determine whether they are more suitable than the current KLT tracker based on the constraints of our system.

To find out, all KLT, SIFT and SURF feature trackers are implemented and run. The three methods detect correspondences in a large number of image pairs; the total processing time and correspondence matching accuracy are collected to determine their overall performances. Two sets of images were tested:

1) Laboratory-produced stereo images with the known correspondences (Section III-B1).
2) Real-life near stereo images with ground truth correspondences are not known (Section III-B2).

1) Evaluation of KLT, SIFT, and SURF on 2005 and 2006 Middlebury Stereo Datasets: There are 30 pairs of indoor images given in the reduced-sized 2005 and 2006 Middlebury datasets; they are at a resolution of 430×370 pixels. The images are carefully acquired in the Middlebury's Vision lab and ground truth is obtained using Structured Light techniques [17]. All the image pairs are horizontally aligned; therefore, the correspondence points are lying on the same horizontal scan lines. We run KLT, SIFT and SURF on these images to obtain 30 sets of correspondences for each. In more detail, each of the methods finds as many as possible correspondence points; then only the strongest pairs are selected to go further. On average, KLT obtains 811 correspondences in 1.15 seconds, SURF collects 810 correspondences in 1.72 seconds, and SIFT collects 714 correspondences in the longest time – 2.04 seconds. Overall, the measurements are summarised in Table I.

TABLE I
STATISTIC DETAILS OF MATCHES WITH KLT, SURF AND SIFT ON 2005 AND 2006 MIDDLEBURY STEREO DATASETS.

Method	AVG matches	STD matches	AVG time	STD time
KLT	811	60	**0.95s**	0.29s
SURF	810	167	1.72s	0.36s
SIFT	714	235	2.04s	0.67s

As the images are rectified, the matched points should be horizontally aligned. To determine the accuracy of a method, we find the misalignment in y-direction $\varepsilon = |y_L - y_R|$. The average and standard deviation of each set are collected. From these obtained data, if all ε are taken into account, the averages are all very large (21 to 31-pixel misalignment) which indicate all three methods contain a large number of outliers. When outliers are discarded by three thresholds $\varepsilon < 0.5$, $\varepsilon < 1.0$, and $\varepsilon < 2.0$, the average misalignments are relatively close to zero. SIFT yields the best; its averages are the smallest. However, with large standard deviation values, there are many cases SURF and KLT are better than SIFT. Overall from the 2005 and 2006 Middlebury data sets, the result shows that KLT, SURF and SIFT are efficient tracking systems, the obtained correspondence sets do contain mismatches. Moreover, they are not significantly different in the correspondences' vertical misalignment. To conclude, SURF and SIFT, in general, take longer to process (see Table I); however, do not obtain better correspondence sets than KLT.

2) Evaluation of KLT, SIFT, and SURF on Real-life Images: In the second test case, the three trackers: KLT, SIFT and SURF are evaluated on 200 real-life images which are randomly selected from our shared image gallery[1]. These images are resized to resolution of 1024×768 pixels. Moreover, rather than only indoor, they are acquired under different conditions. Similarly, all image pairs are tracked with KLT, SURF, and SIFT for correspondences. After the experiment, on average, KLT runs in 2.1 ± 0.7 seconds, SURF runs in 5.7 ± 2.9 seconds, and SIFT takes 7.5 ± 2.0 seconds to obtain up to 500 strongest correspondences. With the standard deviation of 0.70 seconds, KLT is found significantly faster than both SURF (~ 2 times) and SIFT (~ 3 times). The ground-truth correspondences are not known in these 200 pairs of images, thus, their relative matching accuracy can only be evaluated after the images are rectified (to be concluded at the end of Section III-C). In conclusion, KLT is more suitable for our project.

C. Evaluation of RANSAC vs. LMedS

Assuming that we have a good set of correspondences found by KLT. Robust regression algorithms can be used to estimate the corresponding fundamental matrix. The five most popular robust regression techniques are: *Maximum-likelihood estimators* (M-Estimators), *Least-Median-Squares* (LMedS), *Random Sampling* (RANSAC), *Maximum Likelihood Estimation SAmple Consensus* (MLESAC) and *Maximum A Posteriori SAmple Consensus* (MAPSAC). Their details and performances were thoroughly investigated by Armangu et al. [11]. They concluded that the geometries related to the positions of the camera produced by RANSAC were the closest to reality, but RANSAC was sensitive to outliers. Moreover, the performance of RANSAC depends on the value of the chosen threshold for outliers. On the other hand, LMedS could obtain better results where many outliers are presented. Which method: RANSAC or LMedS should we use?

In this experiment, the correspondences obtained from Section III-B2 are supplied to both RANSAC and LMedS regression processes. We assessed six different method combinations: KLT+RANSAC, KLT+LMedS. Each combination returns 200 stereo pairs of rectified images, thus, totally $200 \times 6 = 1200$ rectified pairs of images are obtained. To get the quality evaluation, we first assume that the best method combination makes the best horizontally aligned image pairs. The misalignments between these correspondences are measured. Approximately 270,000 correspondences are examined. We consider that correspondences with misalignments: $|y_L - y_R| > 4$ pixels are outliers and ignored, we have constructed a statistic table comparing the combinations in Table II.

[1] http://www.ivs.auckland.ac.nz/web/scene_gallery.php

TABLE II

Statistic details of misalignment of images after running KLT + RANSAC, KLT + LMedS on 200 real-life near stereo pairs.

Method	Matches	$\|y_L - y_R\|$	[MIN, MAX]	STD
KLT+RANSAC	48,948	**0.76**	[0.0, 4.0]	**0.72**
KLT+LMedS	46,657	**0.77**	[0.0, 4.0]	0.74

From these details, with smallest means and standard deviations, the combinations KLT + RANSAC, KLT + LMedS are relatively equivalent. They all have the statical means of between 0.76 and 0.77 pixels and standard deviation between 0.72 and 0.74 pixels. KLT + RANSAC is slightly better; thus, it is chosen for the rectification of our system. Overall, we can conclude that KLT + RANSAC combination is the best choice for uncalibrated image rectification within our system.

D. Stereo Matching to Vertically Align Images

Disparity range between stereo image significantly affects the comfortability of viewing in VR devices. The best disparity for viewing is zero, in this case, human eyes are paralleled, and the focus point is on the horizon (very far point). For the best viewing, the stereo pair should have a disparity range $[min, mid, max]$ where $min < 0$, $mid = 0$, and $max > 0$.

Stereo Matching extracts disparity information from a stereo pair of images. Given a stereo pair, the matching process outputs a disparity map, which characterises the observed 3D surface. We apply a simple stereo matching on obtained rectified stereo images, then from the disparity map, we calculate the average disparity value a pixels. After that, we can only need to shift one of the images by a pixels. This action makes the average disparity between the two image becomes zero. Now, the two images are horizontally aligned and have zero average disparity, they ready to be sent back to the client to display on VR devices.

IV. Limitation & Future work

Currently, we have developed a system demo only in the Android environment. Therefore, the current test result is only valid in this smart-phone operating system. In our previous work, The stereoscopic image reconstruction feature in our server was tested; which has a result of being fast, and accurate. When we set up a connection between the client and the server; the feature was tested again through the use of client interface. The current test result shows that the feature works as intended, and our system is stable, accurate, and easy to implement on our tested smart-phone model. However, it is still not clear if the system is compatible with all types of smart-phones available on the market, especially with the ones having limited hardware.

In the future, we will port these frameworks to other popular VR environments, and smartphone operating system such as iOS, WebVR. More tests will be performed to ensure that the system can achieve equally good performance in all popular environments.

V. Conclusion

This paper describes an online Virtual Reality (VR) frameworks that allow users to capture and views their photos in stereoscopic 3D, as well as sharing them with the others. With the use of our specialised client-server architecture and the mobile VR platform from Google, our frameworks are designed to achieve good performance in all of the smartphones that are compatible with Google Cardboard and Google Daydream. While our evaluation is limited due to time and budget. The current results have shown a big potential for the system to be implemented in community, education, and entertainment.

References

[1] P. S. Churchland and T. J. Sejnowski, *The Computational Brain*, 1st ed. Cambridge, MA, USA: MIT Press, 1994.

[2] H. Jonnalagadda, "Google has shipped 10 million cardboard vr headsets since 2014," Mar. 2017. [Online]. Available: http://www.androidcentral.com/google-has-shipped-10-million-cardboard-vr-headsets-2014

[3] A. Berson, *Client/server architecture*. McGraw-Hill, Inc., 1996.

[4] H. Richard and Z. Andrew, *Multiple view geometry in computer vision*. Cambridge University Press, 2000, vol. 2.

[5] Q. T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International Journal of Computer Vision*, vol. 17, pp. 43–75, 1996.

[6] P. H. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International journal of computer vision*, vol. 24, no. 3, pp. 271–300, 1997.

[7] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International joint conference on Artificial intelligence*, Vancouver, Canada, Aug 1981, pp. 674–679.

[8] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, New York, USA, Jun 1993, pp. 593–600.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, Kerkyra, Greece, Sep 1999, pp. 1150–1157.

[11] X. Armangué and J. Salvi, "Overall view regarding fundamental matrix estimation," *Image and Vision Computing*, vol. 21, no. 2, pp. 205–220, 2003.

[12] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.

[13] C. Tomasi and T. Kanade, *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ., 1991.

[14] W. Hoff and N. Ahuja, "Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 121–136, 1989.

[15] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of sift features for face authentication," in *Proceedings of IEEE Computer Vision and Pattern Recognition Workshop*, New York, USA, Jun 2006, pp. 35–35.

[16] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[17] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Wisconsin, USA, Jun 2003, pp. 195–202.