

# 基於虛擬視角之立體影片合成

研究生：張鈞凱

指導教授：杭學鳴 博士

國立交通大學

電子工程學系 電子研究所碩士班

## 摘要

立體數位內容日益被重視，新型態的技術包含了自由視點視訊(FTV, Free Viewpoint Television)與擴增實境(AR, Augmented Reality)，這些應用以任意視點合成技術為最主要的關鍵議題。有許多的任意視點合成演算法被提出，通常都是利用多重影像以及其對應的深度資訊圖來產生虛擬試點的影像以達到任意視點的效果。我們利用這種基於影像與深度的影像合成渲染(DIBR, Depth Image-based Rendering)來產生背景置換後的立體視訊內容。

輸入為兩組多個攝影機分別拍攝的兩組視訊，我們希望結合這些輸入來產生新的立體場景。此立體場景由其中一組輸入的前景物體，與另外一組的背景場景共同組成。為了這個目的，我們將以多組視訊間場景間不匹配(mismatch)的角度來觀察，在此論文中主要將討論包括攝影機參數以及攝影機定位的不匹配。當使用者在背景影像中選取了定位點(landing point)，我們需要經由改變攝影機相關參數來合成出背景場景的對應虛擬視角(配合前景攝影機)，以達成背景置換。這樣的方式可以大幅增加創作的自由度。

相較於傳統的影像創作(Image Composition)，上述的過程需要利用到深度幾何的資訊。欲被合成的背景場景需要經由虛擬攝影機參數的計算。此外，為了保持場景物體間互相遮蔽的關係，在背景置換時深度競爭(Depth Competition)是另外一個被探討的議題。當我們將靜態影像延伸至視訊時，我們需要攝影機移動行為的資訊來補償不同場景間攝影機的移動不匹配問題。實驗結果顯示我們可以達成令人滿意的視覺觀感。



# Virtual-view-based Stereo Video Composition

Student: Chun-Kai Chang

Advisor: Dr. Hsueh-Ming Hang

Department of Electrical Engineering &

Institute of Electronics

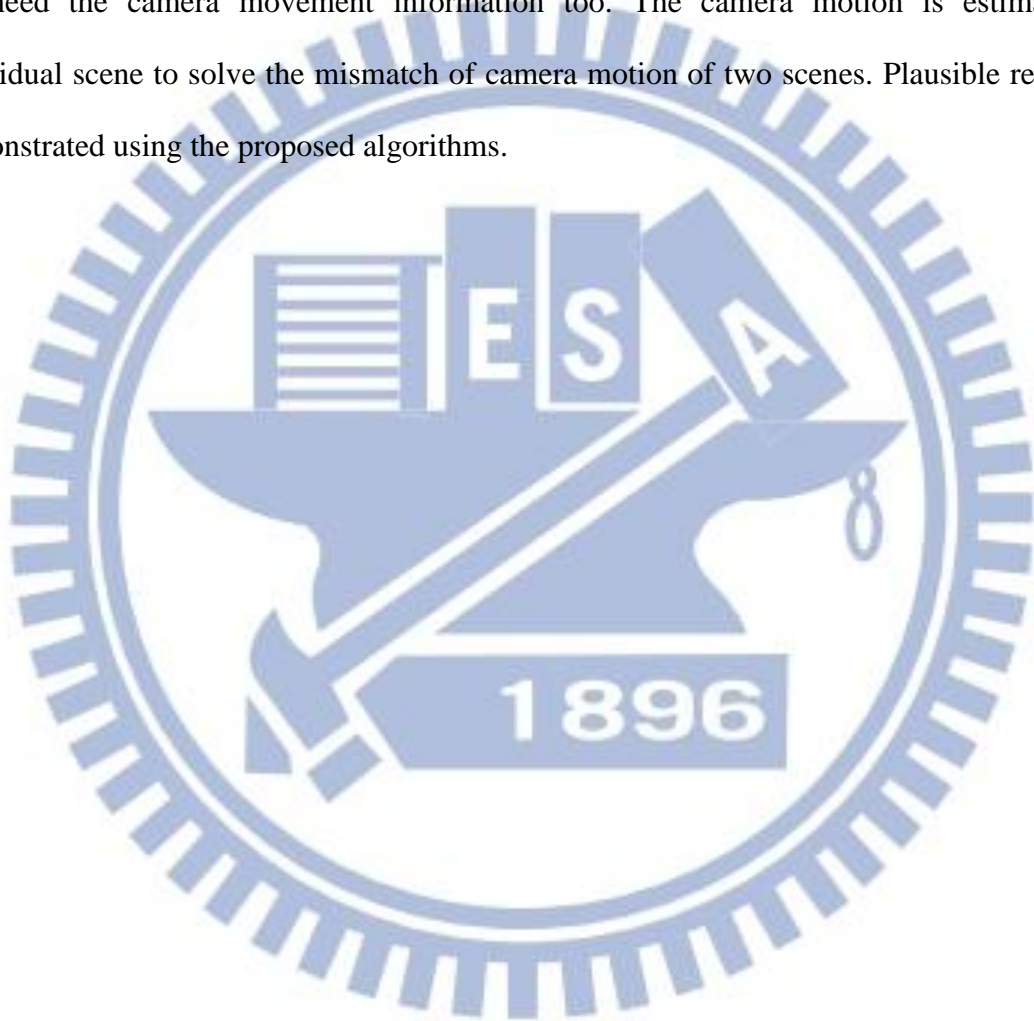
National Chiao Tung University

## Abstract

3D video is gaining its popularity recently. In addition to the conventional left-right view 3D pictures, new forms of 3D video such as free viewpoint TV (FTV) and augmented reality (AR) are introduced. The Depth Image-based Rendering (DIBR) technique is one enabling rendering technique behind these applications. Typically, it uses multiple views with depth information to generate the intermediate view at any arbitrary viewpoint. We can use the DIBR techniques to produce new stereo videos with background substitution.

Given two sets of videos captured by two sets of multiple cameras, we like to combine them to create a new stereo scene with the foreground objects from one set of video and the background from the other set. We will study a few mismatch issues between two scenes such as camera parameter mismatch and camera orientation mismatch problems in this thesis. We propose a floor model to adjust the camera orientation. Once we pick up the landing point (of foreground) in the background scene, we need to adjust the background camera parameters (position etc.) to match the foreground object, which enriches the freedom of composition.

In contrast to the conventional 2D composition methods, the depth information is used in the above calculation. Thus, the new background scenes may have to be synthesized based on the calculated virtual camera parameters and the given background pictures. The depth competition problem is another issue to maintain the inter-occlusion relationship in the composite new scene. If we extend this 3D composition form still pictures to motion pictures, we need the camera movement information too. The camera motion is estimated for individual scene to solve the mismatch of camera motion of two scenes. Plausible results are demonstrated using the proposed algorithms.



## 誌謝

首先要感謝杭學名老師，謝謝老師在這三年來的指導。除了在研究專業上，更感謝的是老師平常對學生的照顧跟關懷，從老師身上看到很多對人的態度和做事的方法，這些都是我人生未來努力的目標。特別感謝能和老師討論這樣子有趣的題目，讓研究不致於太枯燥而有熱忱。

感謝哲瑋，謝謝你和我分享生活的點滴；感謝長廷，謝謝投影片魔術師的教導(當然不只投影片)；感謝志堯，讓我由不一樣的角度看一些東西。

感謝葆崧、怪盜基德、夏銘、信宏、桑 group 和簡 group，實驗室人太多就不一一列舉了，謝謝你們讓我兩年的研究生活充滿樂趣。之後大家也要走往不同的方向了，但我想我會記得這些日子裡一起刻劃的碩士生活樣貌。也感謝育綸、建誠、欣哲、靖倫、敬坤、小哲瑋、治戎、司頻等杭 group 學弟妹這些日子的陪伴，你們看到這篇的時候應該也是明年你們要畢業的時候了。

感謝碩士班兩年來的室友明揚、奕晴、思齊、青維、振鴻，謝謝你們忍受我的洗衣籃。我會想念那些一起宅的日子，也祝你們畢業順利，一帆風順。感謝電資學士班大家庭，謝謝你們的陪伴。

最後要感謝我的家人，謝謝你們期間的關心和支持，當自己像橡皮筋彈性疲乏的時候，可以回家充電，讓我有更大的動力和熱忱完成研究。

寫「致謝」這件事情，起先只是抱著點禮貌性的成分，但後來卻陷入了一連串的回憶，看到一個名字就想起一些小故事，真的是滿懷感謝。要感謝的人真的太多了，沒辦法一一列舉，或有疏漏的，謝謝你們。

張鈞凱

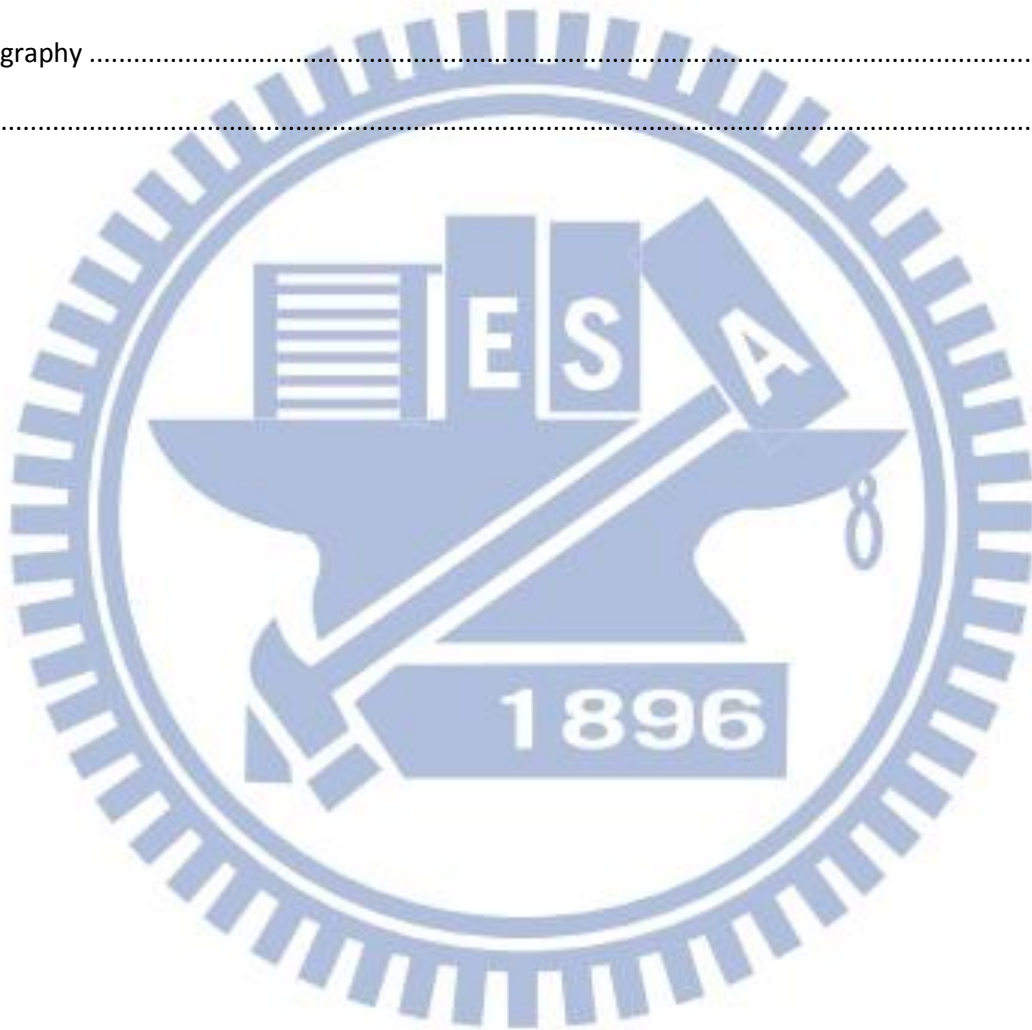
一〇二年七月於 風城 交大 謹致

# Contents

摘要.....	I
Abstract.....	III
誌謝.....	V
List of Figures.....	IX
List of Tables.....	XII
Chapter 1. Introduction.....	1
1.1 Background.....	1
1.2 Motivations and Contributions.....	1
1.3 Organization of Thesis.....	2
Chapter 2. Stereo Geometry Basics.....	3
2.1 Camera Model.....	3
2.1.1 The Basic Pinhole Model.....	3
2.2 Stereo Geometry.....	5
2.2.1 Relation Between Two Images.....	6
2.2.2 Fundamental Matrix.....	8
2.2.3 Projective Geometry and Transformations.....	9
Chapter 3. Virtual View Synthesis and Chroma Keying.....	12
3.1 View Rendering Basics.....	12
3.2 Depth Image-Based Rendering (DIBR) and View Synthesis Reference Software (VSRS).....	13
3.2.1 Texture Warping.....	14
3.2.2 Depth Image-based Rendering.....	15
3.2.3 VSRS.....	17
3.3 Introduction to Image Composition.....	17
3.3.1 Chroma key.....	17

3.3.2	Image composition .....	18
3.3.3	Other Issues .....	21
Chapter 4.	VSRS Framework based Stereo Video Composition System .....	23
4.1	Stereo Composition System Overview .....	23
4.1.1	Composition of Natural Scenes .....	23
4.1.2	Limitation of Conventional Composition.....	24
4.1.3	System Definition .....	26
4.1.4	Overall System Flowchart.....	27
4.2	Camera Orientation Adjustment by 1-D Floor-adjustment Model.....	28
4.2.1	Relative Works of Orientation Mismatch.....	28
4.2.2	Proposed 1-D Floor-adjustment Model.....	30
4.2.3	Depth Correction under Rotation.....	32
4.3	Camera Orientation Alignment with Assigned Landing Point .....	37
4.3.1	User-assistant Landing Point Selection .....	38
4.3.2	Depth Prediction of the Landing Point when Occluded .....	41
4.4	View Synthesis Stage .....	41
4.4.1	Virtual-view-based rendering by modified VSRS.....	41
4.4.2	Virtual Depth Generation .....	42
4.5	Inter-Occlusion of the Stereo Composition View .....	43
4.5.1	Depth Competition under User-assistance .....	43
4.5.2	Hole Filling .....	44
4.6	The Extension to Video .....	48
4.6.1	Camera Matchmoving .....	48
Chapter 5.	Conclusion and Future Work.....	55
5.1	Conclusion.....	55
5.2	Future Work.....	55
Appendix	.....	56

A. Segmentation and Matting .....	56
A.1 Matting .....	56
A.2 Local Thresholding.....	57
A.3 Automatic Tri-map Generation .....	58
B. Color Grading .....	61
B.1 Vector Space Color Balancing.....	62
Bibliography .....	66
自傳.....	68





# List of Figures

Figure 1 : Pinhole camera geometry [5].	3
Figure 2: Principal point offset.	4
Figure 3: Change of Euclidean coordinate	4
Figure 4: Two frames of a sequence shot by a still camera (Object motion)	6
Figure 5: Different views of the same scene (camera motion and scene structure)	6
Figure 6: Epipolar constraints [5]	7
Figure 7: Epipolar lines for the two corresponding points	7
Figure 8: Homography transform for the projective transform of the points on the same 3D plane.	8
Figure 9: Various transformations of 2D [5]	9
Figure 10: Projective transformation	10
Figure 11: Homography transform for the pure rotation of the camera around its centre	11
Figure 12: Rendering Methods Taxonomy	12
Figure 13: Proposed advanced system concept for 3DTV [1]	13
Figure 14: Occlusion problem: Each blank is a pixel filled with its disparity when warping from Cam_L to Cam_R. Note that the black holes are only emerged in the new view.	16
Figure 15: View Synthesis Reference Software (VSRS) [10].	16
Figure 16: Pixel-Intensity problem: certain pixels are filled with wrong intensity due to the non-ideal sampling.	17
Figure 17: Chroma keying [12]: An example for replacement of the background behind a foreground object by green-screen	18
Figure 18: An example of pure composite technique [13]: Content-based Image Synthesis	20
Figure 19: An example of pure composite technique [14]: Photo Clip Art	20
Figure 20: An example of mixed composite technique [15]: Left figure (A) shows input frame on the left and 3D geometry of the baby (foreground object) on the right (B).	21
Figure 21: An example of the implicit composition using color information[16].	21

Figure 22: Problems of composition without geometry	24
Figure 23: Overall system flowchart.	28
Figure 24: Problems of the Orientation Mismatch [14].	29
Figure 25: User marks lines on the floor (green line on the left and red line on the right) for calculating the 1-D floor model. The left and right images are Lovebird1 and Poznan_Street, respectively (MPEG test sequences).	30
Figure 26: 1D camera orientation vs. floor model.	31
Figure 27: Geometry relation of pure rotation	33
Figure 28: The top is the reference image, and the bottom is the target image. The lines indicated the referenced 1-D line, which is on the desktop (so-called the floor) the drink stands on.	34
Figure 29: Calibration for 1-D floor-adjustment model.	35
Figure 30: Before (Left) and after (Right) applying the depth map updated formula.	35
Figure 31: After warping to the identical relative position The left shows the synthesized view using the information of camera pose and height. The right shows the target view.	36
Figure 32: Upper images are the stereo composition results without floor-adjustment; Lower images are with floor-adjustment stage, which appears more natural.	37
Figure 33: Some examples of the picked landing points. The camera location needs to be adjusted to match the assigned ground point. The top figure is the background scene. The red dots in the background are the picked landing points. The result is shown in the order of red points (landing point) <b>from right to left</b> .	40
Figure 34: An example of final stereo composition result. The background is Pozna_CarPark [18] (MPEG test sequences).	42
Figure 35: Depth competition map under different racing threshold (top-left: no thresholding / top-right: $th=-2000$ / bottom-left: $th=-2500$ ). Bottom-right is the bottom-left 's depth competition result( $th=-2500$ ).	44
Figure 36: Large holes appear due to warping process	45
Figure 37: Top figure shows all the connected holes. The lower figure is the enlarged region marked by the red rectangle.	46

<i>Figure 38: Histogram of the neighboring pixels of a hole. The horizontal axis is depth value from 0~255. The vertical axis is the pixel number.</i>	47
<i>Figure 39: Result after applying our hole-filling techniques</i>	47
<i>Figure 40: Found putative matches from two images.</i>	50
<i>Figure 41: Reconstruction ambiguity [5].</i>	51
<i>Figure 42: Reconstruction ambiguity even in the calibrated case: the actual 3d structure may be under a similarity transform.</i>	52
<i>Figure 43: Motion compensation result of Poznan_Hall2 sequence</i>	54
<i>Figure 44: Composition result by combing static foreground scene and moving background scene. The top is the stereo pair in frame #1. The bottom is the stereo pair in frame #51</i>	54
<i>Figure 45: Supervised image matting by using trimap. The left figure is the reference image. The middle figure shows the trimap as input, which labels some region that is definitely foreground (white) or background (black).The right figure shows its output (alpha matte), which is usually followed by a compositing process to create a new image by linearly blending the extracted foreground object image and a new background image with the output alpha matte.</i>	56
<i>Figure 46: Flowchart of tri-map generation</i>	57
<i>Figure 47: Two examples using proposed row-by-row thresholding: The left image is the selected original depth; the middle is the result using Otsu's threshold; the right is the result using our method.</i>	58
<i>Figure 48: Result of edge detection by sobel flter and thrsholding.</i>	59
<i>Figure 49: Temporary result of automatic tri-map generation.</i>	59
<i>Figure 50: Result of ballet sequence(non-cropped mode)</i>	60
<i>Figure 51: Temporary result of row-by-row thresholding of cropped mode (Lovebird1): The top left is the depth image; the top right is the binarized depth after row-by-row thresholding; the bottom left is the tri-map and the bottom right is the output alpha matte.</i>	61
<i>Figure 52: Linear blending result with black background.</i>	61

Figure 53: Color balancing[16].	62
Figure 54: Color gamut: each color is viewed as a vector in 3-D space.	63
Figure 55: Marked color correspondence (marked as the circles with the same color by users).	64
Figure 56: The left shows the original image; the right shows the tuned result after the vector space color balancing.	64
Figure 57: Test with color calibration by using 7 sparse color correspondences. The left is the reference image; the middle is the target image; the right is the adjusted result after the vector space color balancing.	65

## List of Tables

Table 1: List of 2D transformations	9
Table 2: List of 3D transformations	10
Table 3: Mismatch type under consideration	26

# Chapter 1. Introduction

## 1.1 Background

The newly developed 3D visual effect offers a whole-new visual experience to human beings, such as 3DTV [1][2], free-viewpoint TV [3] (FTV), and 3D movies. Thanks to more and more matured advanced video compression coding technique, they are included in the new Multi-view Video plus Depth (MVD) format specified by the international ISO-MPEG/ITU-T standard committee in 3D video coding. Examples are 3DV-ATM (Advance Video Coding Test Model) and 3DV-HTM (High Efficiency Video Coding Test Model), which are AVC (Advance Video Coding) based and HEVC (High Efficiency Video Coding) based reference software of 3D video coding, making 3D home entertainment feasible. However, the production of 3D content is rather difficult compared to that of the conventional 2D multimedia.

Combining two sets of 3D scenes into a set of 3D new scene is a very challenging task. Thanks to the progress of computer computational ability and computer vision techniques, the production of 3D movie is feasible. Yet, the production still needs heavy manual effort. Take “Life of Pi” directed by Ang Lee for example, the special effect company R&H takes thousands of people working around the clock to make the post-production of the movies to achieve such an outstanding special effect.

## 1.2 Motivations and Contributions

This work is inspired by the conventional video composition video on Youtube [4], which demonstrates some post-production special effect in the movies and dramas. It shows several video composition techniques such as chroma key and scene completion. The video composition technology allows one to substitute whatever scene he/she wants into another

sequence as background as in the video clip [4]. We extend the similar idea to the stereoscopies by integrating two or more MVD (Multi-view Video plus Depth) sequences into one 3D video. We wish to develop techniques that can produce a new 3D scene in a semi-automatic way. The goal is to make the creation and manipulation of composition as simple and effortless as possible.

### **1.3 Organization of Thesis**

In this thesis, we first review the stereo geometry basics in chapter 2. We then introduce the image rendering techniques in chapter 3. We will also briefly introduce some existing image composition examples in chapter 3. The proposed virtual-view-based stereo video composition system except for color balancing and segmentation (which are discussed in Appendix) will be described in chapter 4. The intermediate and final results are also shown in chapter 4. Finally, conclusions and future work will be mentioned in chapter 5.

# Chapter 2. Stereo Geometry Basics

## 2.1 Camera Model

An image is formed by mapping the 3D world (object space) to a 2D image through camera devices. Due to the simplicity, here we only introduce the most specialized and simplest camera model, which is the basic pinhole model. More detail can be found in [5].

### 2.1.1 The Basic Pinhole Model

For simplicity, first we let the centre of projection be the origin of a Euclidean coordinate system. Consider the plane  $z = f$ , which is called the image plane or focal plane. Based on this model, a point in 3D world coordinate  $X=(X, Y, Z)^T$  is mapped to the point  $(fX/Z, fY/Z, f)^T$  on the image plane.

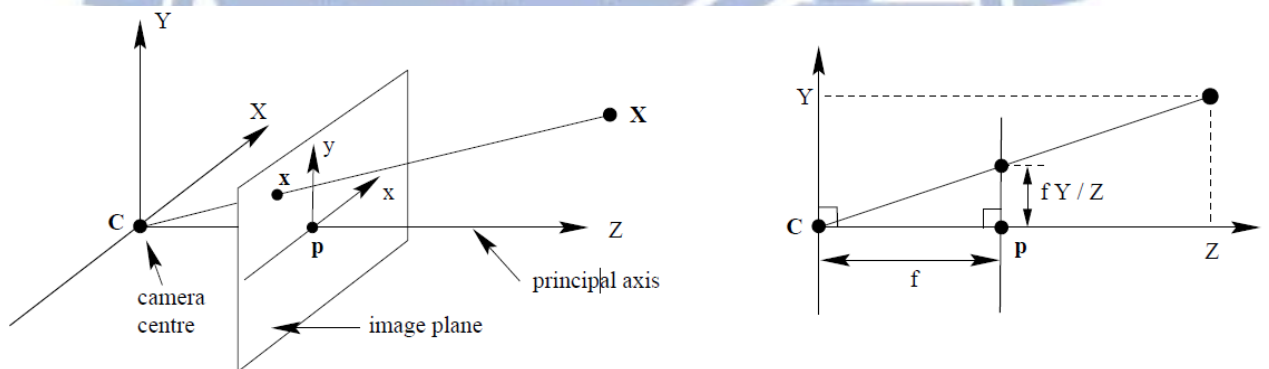


Figure 1 : Pinhole camera geometry [5].

In homogeneous coordinates, we have

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & & \\ & f & & \\ & & 1 & \\ & & & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

Thus, we define projection matrix  $P$  which shows how a 3D point projects to the image plane.

$$x = PX, \text{ where } P = \text{diag}(f, f, 1) [I | 0] \quad (2)$$

Here, we assume the origin of coordinates in the image plane is at the principal point. In general, there is a mapping

$$(X, Y, Z) \mapsto \begin{bmatrix} fX + zp_x \\ fY + zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & p_x & 0 \\ f & p_y & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K[I | \underline{0}]X_{cam}, \quad (3)$$

where  $K$  is called **camera calibration matrix** or **intrinsic matrix**.  $X_{cam}$  denotes the point located in this camera image coordinate system.

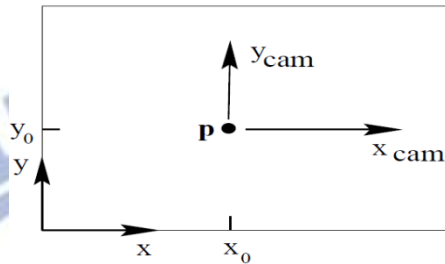


Figure 2: Principal point offset.

Practically, consider the skew parameter and the possibility of non-square pixels (in CCD camera), the most general expression of the intrinsic parameter is of the form

$$K = \begin{bmatrix} f_x & s & pp_x \\ & f_y & pp_y \\ & & 1 \end{bmatrix}, \quad (4)$$

where  $f_x$  and  $f_y$  refers to the focal length in the  $x$  and  $y$  direction, and  $pp_x$  and  $pp_y$  refers to the principal point offset. Both of them are defined in terms of pixel unit. In this form, the camera calibration matrix builds the model for the **finite projective camera**.

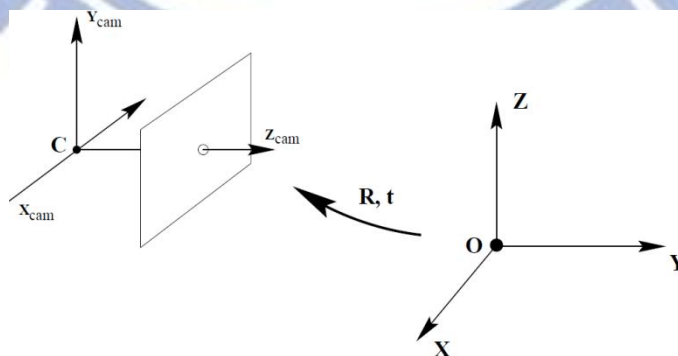


Figure 3: Change of Euclidean coordinate

In general, the camera coordinate may not be aligned with your assumed coordinate. In other words, points in space are expressed in terms of Euclidean coordinate frame or **world**



**coordinate frame.** Relative to the camera coordinate frame, we can relate the two with a rotation and a translation. We may write  $\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$ , where the “~” symbol refers to a point located at the world coordinate frame.  $\tilde{C}$  represents the coordinates of the camera centre, and R is a 3x3 rotation matrix. In the homogeneous coordinates, we have

$$\tilde{X}_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X \quad (5)$$

Putting them together and we obtain

$$x = KR[I | -\tilde{C}]X. \quad (6)$$

Here, the projection matrix P, which shows how a 3D point is mapped to the image plane has been derived in our familiar way,

$$x = PX, \text{ where } P = K[R | t], \quad (7)$$

and  $t = -R\tilde{C}$  stands for translation vector. We can see that the translation actually comes from the camera centre in the world coordinate system and goes through a rotation due to the change of Euclidean coordinate.

## 2.2 Stereo Geometry

Based on the finite projection camera model, we can further describe the relationship between two images. Typically we are interested in the two images at different views at the same time (Figure 4), or at an identical view at different time slots such as two consecutive frames in a video shot by a still camera (Figure 5).



Figure 4: Two frames of a sequence shot by a still camera (Object motion)

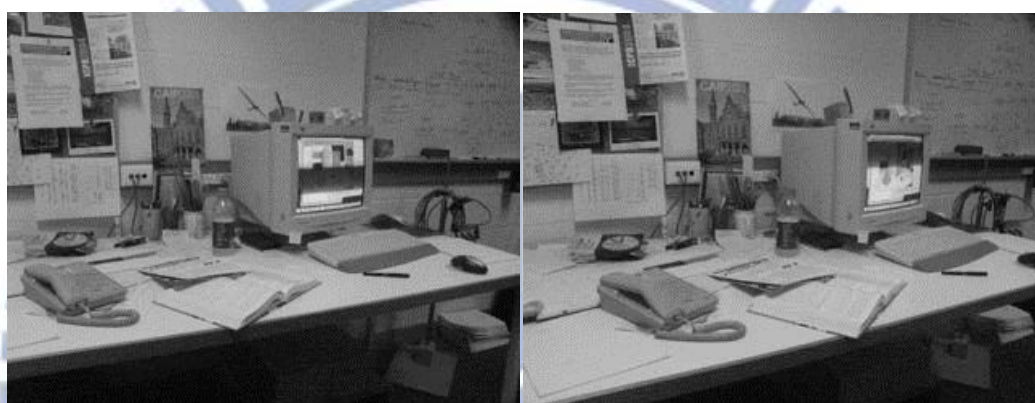


Figure 5: Different views of the same scene (camera motion and scene structure)

### 2.2.1 Relation Between Two Images

Consider the cases in Figure 6, if two cameras take pictures at different views, we can exploit the 3D geometry relationship to obtain the information of this space.

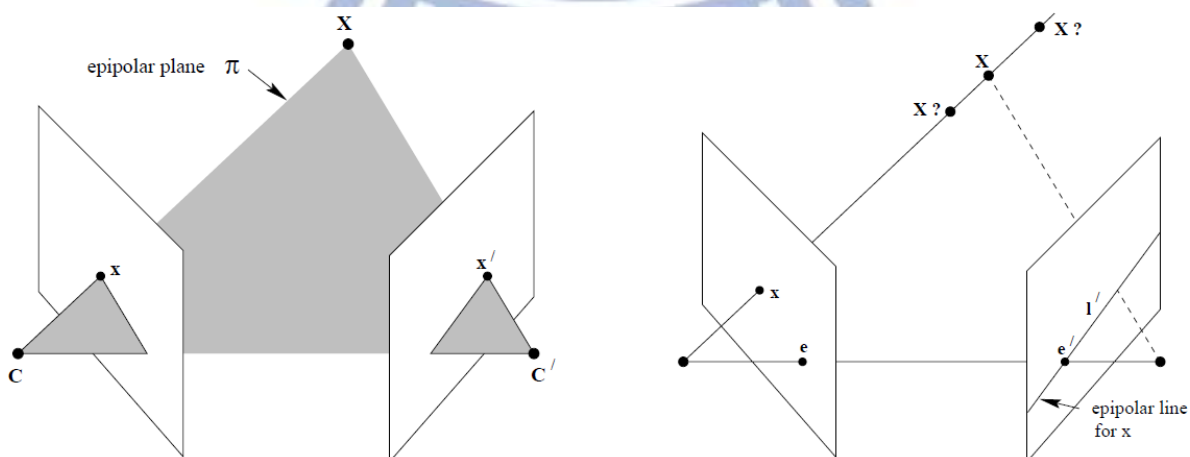


Figure 6: Epipolar constraints [5]

A point  $X$  shows up in two camera's view, and is projected to two image planes as  $x$  and  $x'$ . Typically, there are three questions that will be addressed:

- (i) Correspondence geometry. For a point shown up in the first view, can we find the point  $x'$  in the second view accurately?
- (ii) Camera geometry (motion). If we know a set of points in the two views corresponding with each other, can we retrieve the projection matrix for each of the two views?
- (iii) Scene Geometry (structure). In addition to the set of corresponding points and the two projection matrix, can we say something about the position of  $X$  in 3D-space?

In this thesis, we focus more on the first two issues but the last since we are not going to construct the 3D model. For the first two issues, we need to exploit the stereo geometry.

In Figure 6, the points  $X$ ,  $C$ , and  $C'$  form a common plane called epipolar plane. Supposedly, only projected points  $x$  and  $x'$  are known. We may determine the plane along with baseline. A certain point  $x$  in one image corresponds with an epipolar line in another image plane as  $l'$ .



Figure 7: Epipolar lines for the two corresponding points

This fact limits the range for searching the correspondence of a particular point in one of

the image planes in another view. And it thus tremendously reduces the complexity and calculation amount. We will explain its property in 2.2.2.

### 2.2.2 Fundamental Matrix

The fundamental matrix is an algebraic representation of the epipolar geometry. As mentioned above, any particular point projected on the image plane corresponds to an epipolar line in another image. Actually, the fundamental matrix satisfies the condition that for any pair of corresponding points  $x \leftrightarrow x'$  in a homogeneous system of the two images:  $x'Fx = 0$ .

We can verify by seeing that  $x'$  lies on the epipolar line

$$l' = e' \times x' = [e']_x x'.^1 \quad (8)$$

And we know that there exists a 2D homography  $H_\pi$  mapping each  $x'$  to  $x$  such that  $x' = H_\pi x$ , where  $H_\pi$  is the mapping function from one image to another via any virtual plane  $\pi$ .

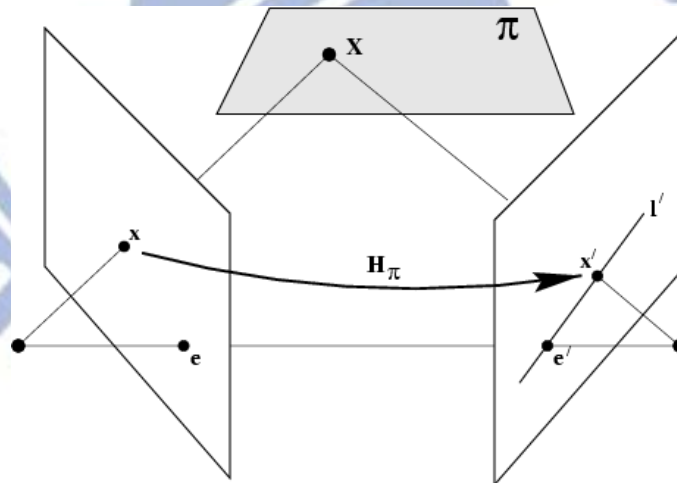


Figure 8: Homography transform for the projective transform of the points on the same 3D plane.

<sup>1</sup>  $[a]_x$  refers to the skew symmetric matrix for vector  $a$ . Cross product between  $a$  and any vector can be written in terms of inner product's form. i.e.  $a \times b = [a]_x b$ .

So  $l' = [e']_x H_\pi x = Fx$ , and the collinear relation of  $x'$  and  $l'$  made  $x'Fx = 0$ . Actually,  $F$  is a 3x3 rank2 matrix, with (degree of freedom) d.o.f. = 7.

### 2.2.3 Projective Geometry and Transformations

The following table is the summary of the 2D transformations:

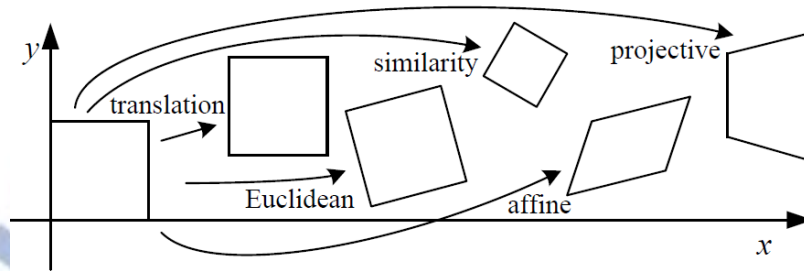


Figure 9: Various transformations of 2D [5]

Table 1: List of 2D transformations



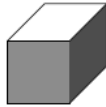
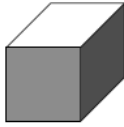
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix}_{2 \times 3}$	2	orientation + ...	
rigid (Euclidean)	$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}_{2 \times 3}$	3	lengths + ...	
similarity	$\begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}_{2 \times 3}$	4	angles + ...	
affine	$\begin{bmatrix} A \\ 0 & 1 \end{bmatrix}_{2 \times 3}$	6	parallelism + ...	
projective	$\begin{bmatrix} \tilde{H} \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3}$	8	straight lines	

Note that the table is listed in a hierarchy order (The lower classes contain those upper) with increasing d.o.f.. The most important one is the projective transform, which accounts for the change of view<sup>2</sup>.

<sup>2</sup> The symbol  $H$  represents the Homography transform. Usually it has 8 d.o.f..

For 3D transformations:

Table 2: List of 3D transformations

Group	Matrix	Distortion	Invariant properties
Projective 15 dof	$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}$		Intersection and tangency of surfaces in contact. Sign of Gaussian curvature.
Affine 12 dof	$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity, $\pi_\infty$
Similarity 7 dof	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$		The absolute conic, $\Omega_\infty$ .
Euclidean 6 dof	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$		Volume.

The table is also listed in a hierarchy order. The lower class of transformation is a particular case of the upper one.

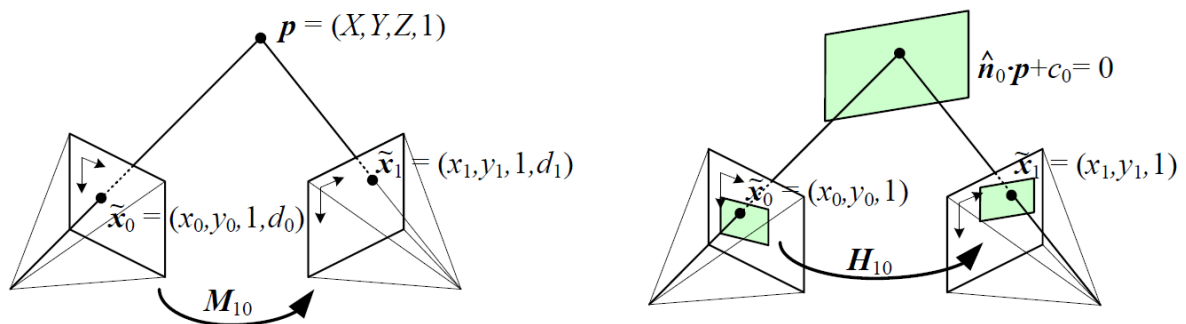


Figure 10: Projective transformation

Generally, for the change of view as in the left figure, the two images relates with each

other with a projective matrix (d.o.f. = 15). Yet, if the points  $p$  in 3D space lies on the same plane as the right figure, the projected points on the two image planes relate with each other by a homography transform (d.o.f. = 8). Similarly, if the two cameras have the same centre but different rotation, the two images are related by a homography transform. We can see that by assuming  $d=0$ .

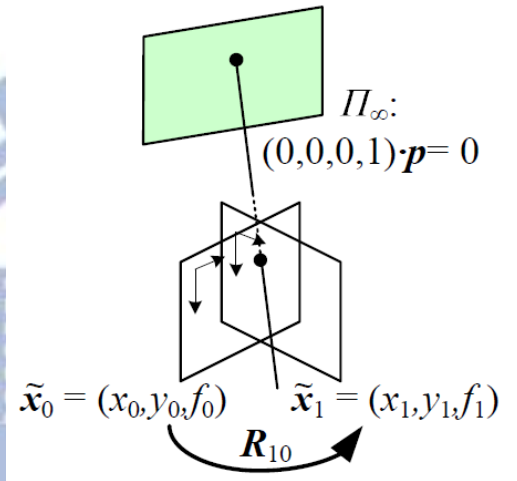


Figure 11: Homography transform for the pure rotation of the camera around its centre

# Chapter 3. Virtual View Synthesis and Chroma Keying

## 3.1 View Rendering Basics

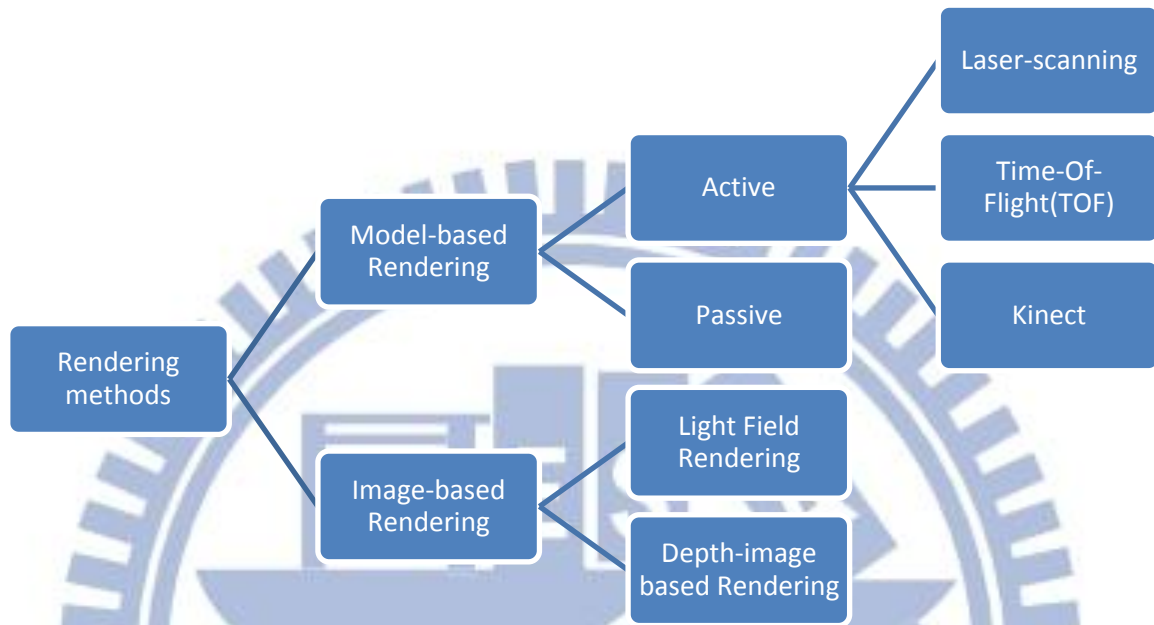


Figure 12: Rendering Methods Taxonomy

Most view synthesizing methods can be classified into two groups, **Model-Based Rendering (MBR)**, and **Image-Based Rendering (IBR)** [6]. The former tries to reconstruct the 3D model of scene or objects, and then project to the virtual view. Based on how the model is reconstructed, both active and passive methods are used. The active methods use laser-scanning, time-of-flight (TOF), or Kinect devices, to measure the 3D scene. In contrast, the passive methods aim to search the corresponding feature points like edges or corners. From the correspondence information within images, we can estimate the stereo information (depth).

While MBR tries to reconstruct the 3D model, IBR exploits the relationship among cameras to do view interpolation in the image texture domain. The most well-known methods are **Light Field Rendering** [7] and **Depth Image-Based Rendering (DIBR)** [6][8]. Light Field Rendering uses Plenoptic function to directly interpolate the virtual view; nevertheless, at the cost of a very large amount of data. DIBR, in contrast, only needs a few two-dimensional



texture images and depth images, so the amount of data is relatively small. But it often has the holes and occlusion problems.

In our work, we adopt the View Synthesis Reference Software (VSRS) as the basis framework for image synthesis, which is mainly based on the DIBR method. More detailed description of DIBR and VSRS is given in section 3.2.

### 3.2 Depth Image-Based Rendering (DIBR) and View Synthesis Reference Software (VSRS)

DIBR is one of the most well-known methods in virtual view synthesis. The basis idea is to project the texture of the known view to a virtual view, by way of view warping technique. The warping technique retains the disparity when the depth information is obtained. The disparity refers to the difference in image location of the same object in two images.

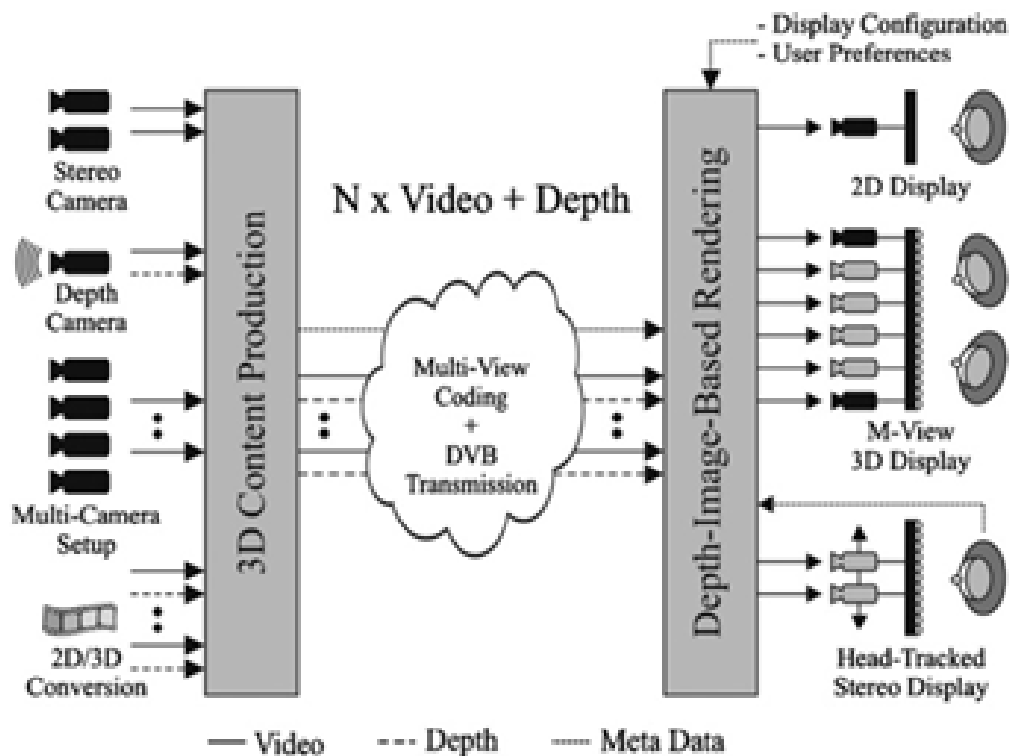


Figure 13: Proposed advanced system concept for 3DTV [1]

### 3.2.1 Texture Warping

Texture warping is an essential technique in Image-based Rendering. In the early methods, only H3x3 transformation matrix is used, with the assumption that the object in the image can only be on a plane. Recently Depth Image-based Rendering (DIBR) has been developed. DIBR uses the information of depth to project the reference texture to the virtual view, which is described as follows.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = z_r Q_r^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} + c_r \quad (9)$$

$$z_v \begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = Q_v \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - c_v \quad (10)$$

The subscription r and v stand for reference view and virtual view, respectively.  $Q[I|-C]$  (or  $K[R|t]$ ) refers to the camera parameters.  $(u, v)$  denotes the image coordinate and  $(X, Y, Z)$  is for the 3-D world coordinate. Combining eq.(9) and (10), we have,

$$z_v \begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = z_r A \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} + b, \quad A = Q_v Q_r^{-1}, \quad b = Q_v (c_r - c_v) \quad (11)$$

By eq.(11), we can project the texture from  $(u_r, v_r)$  to  $(u_v, v_v)$  with the assistance of depth information. According to the depth's reference, two kinds of warping are proposed. **Forward warping** uses the depth of reference and project to the virtual view, while backward warping uses the virtual depth image to obtain the texture of the corresponding coordinate in the reference images. Generally speaking, the backward warping works better than forward warping in consideration of pixel rounding problem and artifacts along the boundaries. These problems are illustrated in below.

### 3.2.2 Depth Image-based Rendering

We have discussed the 3D projection theory. But practically, in a complete process of view synthesis, we have to deal with several problems including **pixel rounding**, **mapping competition**, **black-hole**, **pixel-intensity accuracy problem**, etc. The key to judge whether an algorithm is good depends heavily on how it solves these problems.

When warped from the reference view, the corresponding pixels in the target view may locate in non-integer positions and some cracks in images may appear due to rounding. These cracks are rather small so they are removed by the simple median filter. However, careless uses of the filter will erode the black-hole region.

Mapping competition is caused by the inter-occlusion of the objects. This problem can also be regarded as “many-to-one” problem since it happens when multiple points are warped to the same target pixel in the virtual view. To solve this problem, we usually adopt the z-buffer method, also named as “front-most scheme”. We store all the depth values of the competitors and compete with one another each time. During each competition, we keep the one with the nearer depth value to ensure the relation of inter-occlusion can be maintained.

Black-hole problem is also due to the occlusion problem, but it is just totally the reverse of the mapping competition problem. Black-hole problem can also be regarded as “none-to-one” problem. As the baseline among the cameras is too sparse, some points in the virtual view cannot be warped by the points in the reference view, and thus a hole is formed. Figure 14 shows the problem. Since there is no information at the black-hole region, it can only be repaired by some image inpainting techniques of computer graphics. The most popular ways are to extend the background, or to use more complicated inpainting methods by the assumption of strong spatial correlation. This formidable problem drastically degrades the visual quality. To alleviate the problem, most view synthesis utilized at least two (left and right) reference views such as VSRS (shown in Figure 15).

The reference image is usually stored in the accuracy of integer-pixel. The non-ideal sampling rate may cause small pin-holes around depth discontinuity, even with perfect depth map. The problem is shown in Figure 16. “Splatting” is a well-known technique in the field of computer vision to combat such problems. With splatting, each pixel in the virtual view may refer to several pixels (by average of weighting) warped from the reference view. However, the process is a low-pass filter and will blur the texture. So in the VSRS, it detects the boundary region of depth map first and uses splatting only for these region.

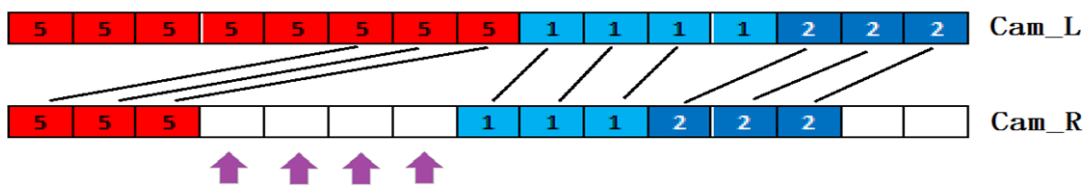


Figure 14: Occlusion problem: Each blank is a pixel filled with its disparity when warping from Cam\_L to Cam\_R. Note that the black holes are only emerged in the new view.

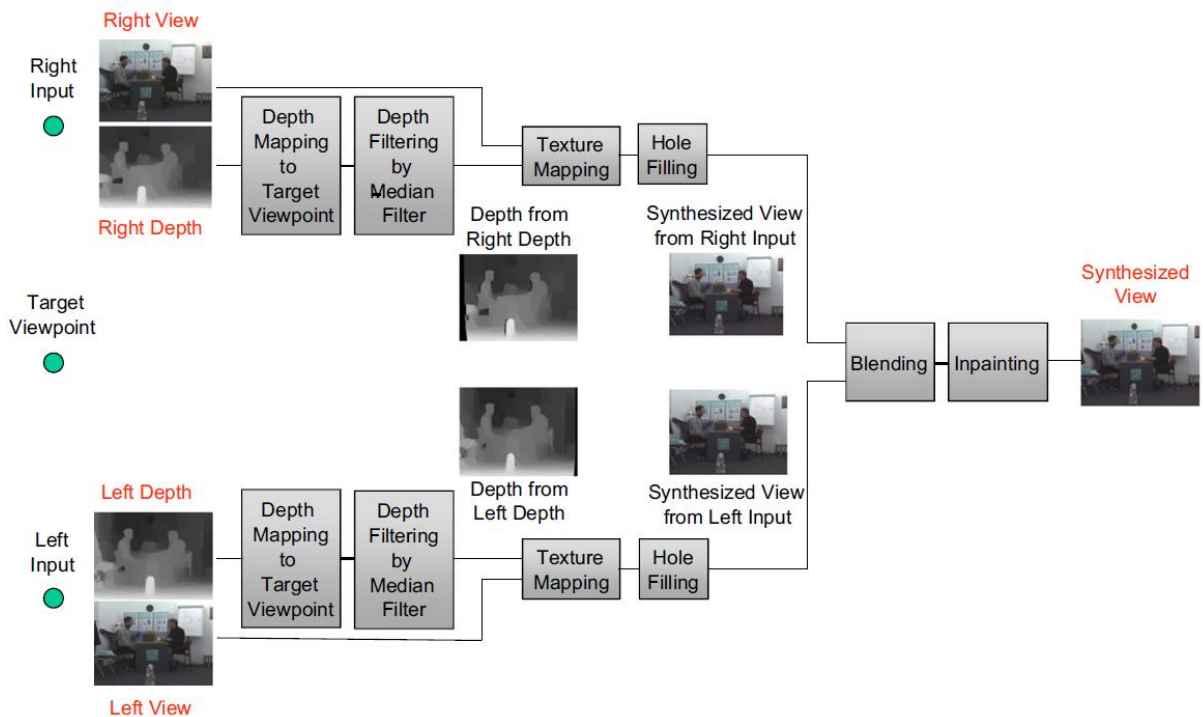


Figure 15: View Synthesis Reference Software (VSRS) [10].

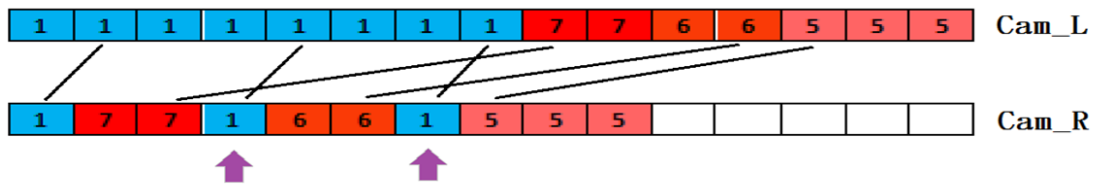


Figure 16: Pixel-Intensity problem: certain pixels are filled with wrong intensity due to the non-ideal sampling.

### 3.2.3 VSRS

For the 3D video and Free-Viewpoint TV (FTV) project, the 3D video group within MPEG (ISO/IEC Moving Picture Expert Group) [9] has developed software for depth estimation and view-synthesis: DERS (Depth Estimation Reference Software) and VSRS. We use VSRS version 3.5 as our framework previously downloaded from MPEG website, which is the final version updated at 27 August, 2009. A description of implemented algorithm can be found in [6]. The OpenCV [11] (Open Source for Computer Vision) library is required. In this thesis, slightly changes have been made for our purpose. We will explain more detail in the chapter 4.

## 3.3 Introduction to Image Composition

Thanks to search engines and social networking sites, such as Google, Flickr, and Bing, a large number of images are available on the Internet. Using these materials, we have the opportunity to create composite images. Composition of images can be created by combining parts from two or more images to create a single image.

### 3.3.1 Chroma key

Chroma key is a special technique for compositing (layering) two images or video streams together based on color hues (chroma range). It has been used in TV and movie industry for a

long time. Specifically, it has been used extensively as a post-production or special effect in newcasting, motion pictures, video games and movie industry mainly for segmentation or background substitution. Chroma key composition is also called color keying, colour-separation overlay (CSO), green screen, or blue screen, in various terms for specific color-related variants. Chroma keying can be implemented with backgrounds of any color that are uniform and distinct from the objects. Blue backgrounds are commonly used because it differs most distinctly in hue from most human skin colors. In consideration of blue color is popular in cloths like pants, green backgrounds are recently used more often.



Figure 17: Chroma keying [12]: An example for replacement of the background behind a foreground object by green-screen

### 3.3.2 Image composition

Many composite techniques combine parts from two or more images to create a single image.

Depending on the source of materials, it can be mainly divided into two classes: **pure** and **mixed composition** techniques. Pure composition is made from real images, while mixed composition is created from both real and synthetic images.

An example of pure composite technique is shown in Figure 18. The city skyline (upper) part from top-left image is combined with the lower part from the bottom-left image and their composite image is shown on the right. Another example is Photo Clip Art, which is shown in Figure 19. Its database collects thousands of images from the net. The system selects the best matching texture from the database by considering geometry and illumination (global lighting condition), and local context of the inserted object and the scene. Specifically, photo clip art computes the approximate 3D structure of the input image and is suitable for obtaining geometrically consistent composite images.

A mixed composition example is shown in Figure 20. The accurate 3D structure of the foreground objects is constructed by multiple-view videos (around the object). Once we have 3D geometry, a simulation of fluids interacting with the foreground objects can be rendered. Figure 20 shows one frame of the video where the baby is drenched with artificial water and honey.

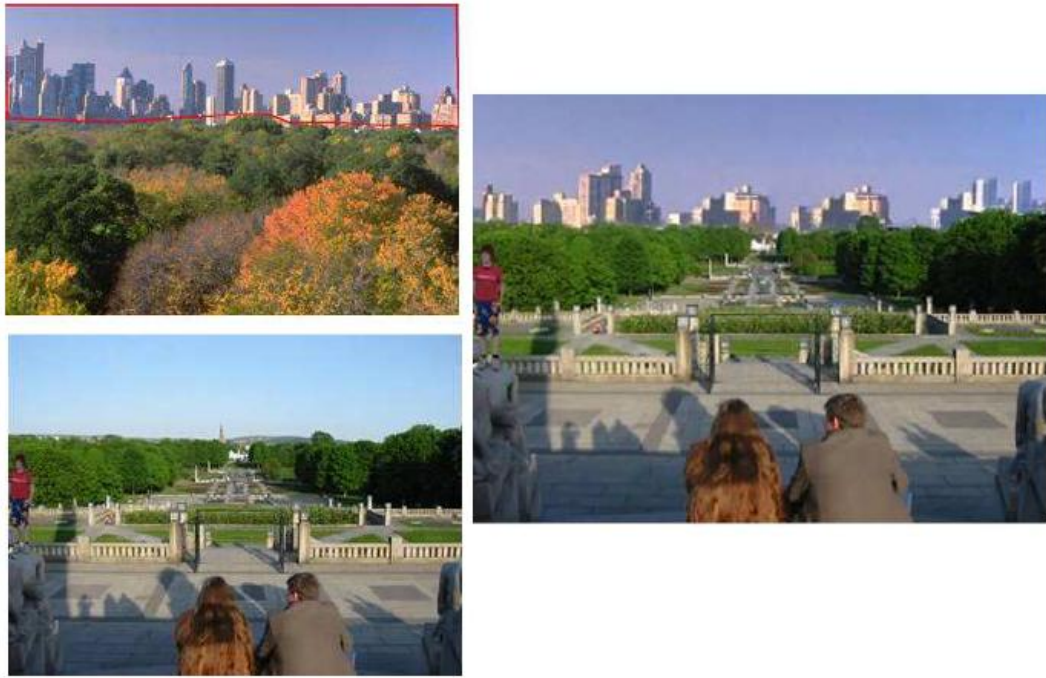


Figure 18: An example of pure composite technique [13]: Content-based Image Synthesis



Figure 19: An example of pure composite technique [14]: Photo Clip Art



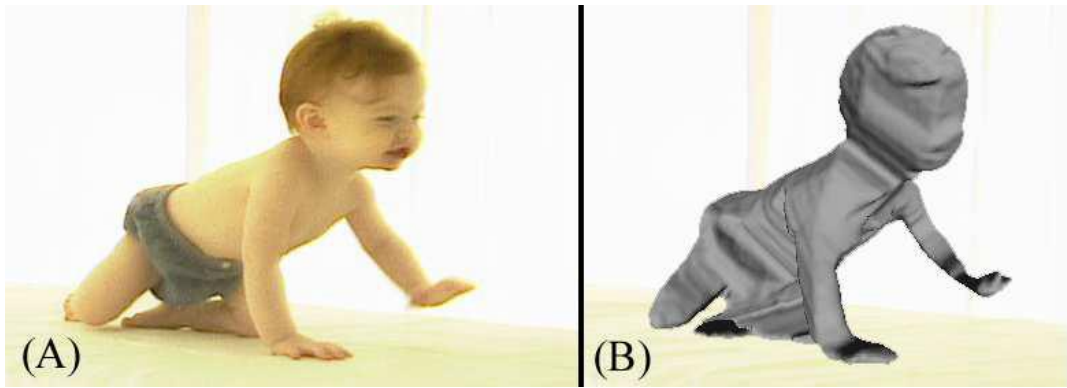


Figure 20: An example of mixed composite technique [15]: Left figure (A) shows input frame on the left and 3D geometry of the baby (foreground object) on the right (B).

### 3.3.3 Other Issues

Images can be combined in various other fashions. Some compositing techniques take specific parts from the input images. For example, they can take colors from one image and the content from another. Figure 21 shows an composite example using the color transfer technique in augmented reality application and images, where colors from one image are transferred to modify the color of another image.

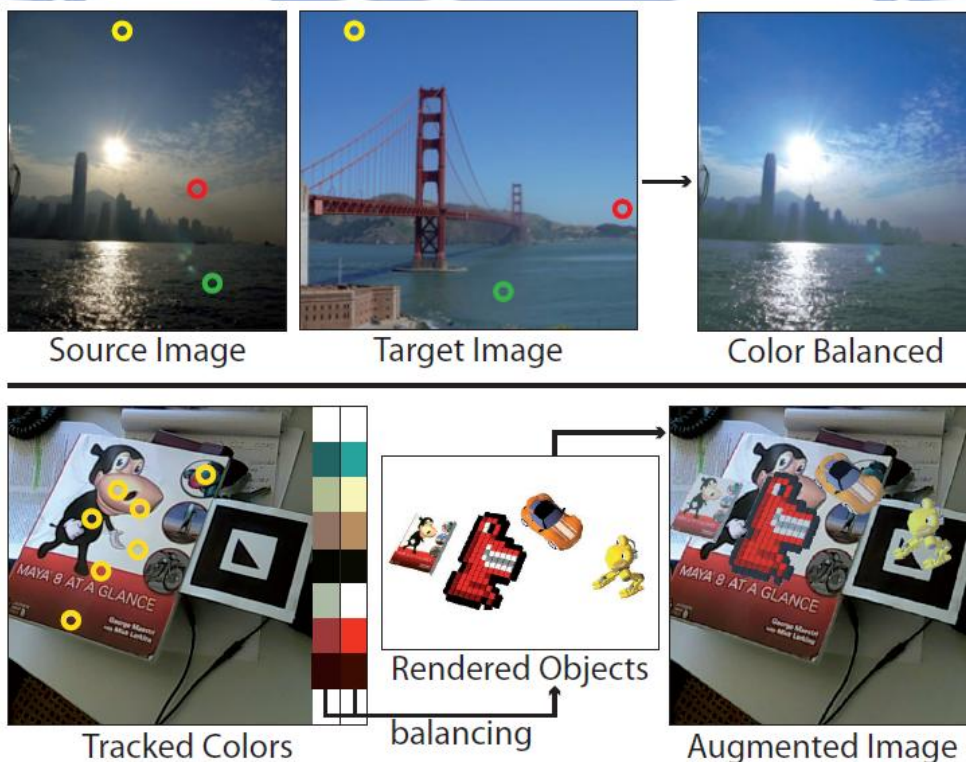
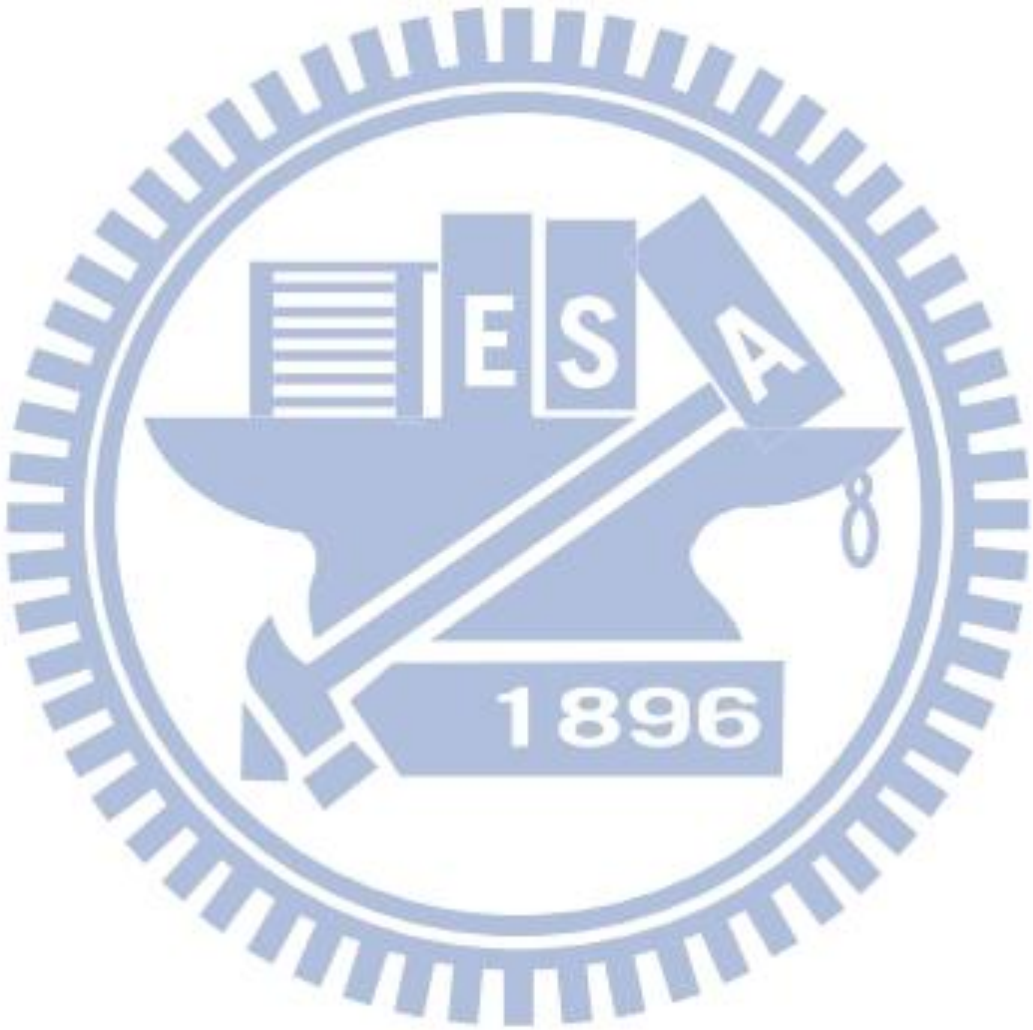


Figure 21: An example of the implicit composition using color information[16].



# Chapter 4. VSRS Framework based Stereo Video Composition System

First, we will explain the purpose and the flow of our system in sec. 4.1. Then, we are going to describe the proposed 1-D Floor-adjustment Model and depth update function in 4.2. Next, we develop the camera orientation alignment algorithm with the user-selected landing point in 4.3. In 4.4, we will explain the image rendering process in our system and the modification we made to the VSRS. Depth competition and hole-filling will be discussed in 4.5. Finally, the extension to video is outlined in 4.6.

## 4.1 Stereo Composition System Overview

### 4.1.1 Composition of Natural Scenes

As mentioned earlier, we try to merge two or more 3D contents into a single image. One similar application in 2D is augmented reality (AR). With the help of AR technology (combining computer vision and object recognition techniques), we can insert extra object information into the image of the surrounding real world. The user can interact and manipulate these artificial objects. The extra information of the environment and objects can be overlaid on the real world image.

In an AR application, we often need to calculate the camera orientation and movement so that a synthetic object (typically produced by computer graphics) can be properly inserted into a scene. In contrast, we are interested in combining two or more natural videos, where both scenes consist of time-varying natural objects. Because both scenes are not generated by computer, constructing a 3D model of natural objects based on limited views is often difficult (if not impossible). Also, 3D modeling usually requires high computational complexity and a large amount of memory.

Here, we adopt the virtual view synthesis technique to generate the background scenes (and sometime foreground objects) that match the scene geometry of the user selected viewpoint. The extension of the technique to video requires camera trajectory recovery related techniques.

#### 4.1.2 Limitation of Conventional Composition

The conventional composition such as inserting images into another image with photo realism imposes a number of challenges. The formidable challenges of composition are to satisfy **camera geometric consistency** and **photo content consistency**. Figure 22 shows a simple composition where we paste a segmented horse (extracted from a scene) into another street scene without any adjustment. There are several noticeable defects in the composite picture. Despite the lack of horse shadow, the first problem you may notice is the size of the horse, which seems to be too small. In addition, the slope of the grass (the horse originally stands on) differs from that of the road. The standing pose of the horse does not seem naturally to match the street floor. Furthermore, if we look at the 3D picture, the depth of house does not match the depth of the ground where it stands.

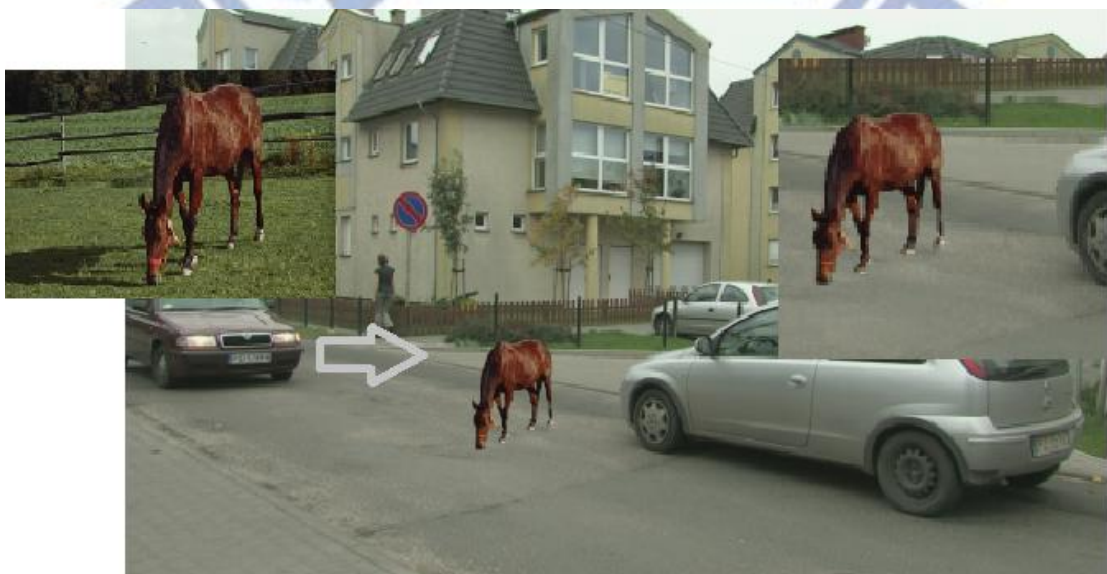


Figure 22: Problems of composition without geometry

There are many issues in producing a composed picture. In the conventional composition methods, a firm creator may need to spent lots of efforts to adjust the size, or specify an appropriate affine transformation on the background so that the horse stands on the floor naturally. When the camera parameters, the true object size, and orientation are not all available, these labor-intensive adjustments may not match the real scene.

Few researches have combined depth information in 2-scene composition. Dimitropoulos et al. [6] proposes an approach for 3DTV synthesis. Their work employs the chroma key technique to decompose a scene into foreground and background. They estimate the depth map using two views. Then, they can generate a new representation of the scene by combining the foreground objects with any background, given two sets of color images and their depth maps. However, this direct composition does not consider the mismatch between the background scenes and the foreground scene. And the new scene creation is limited by the straightforward background replacement.

The difficulty of 3D video composition mainly lies on the mismatch of the two different scenes. This mismatch may result in artifacts, causing imperfect perception in the compositing images. The mismatch type can be mainly classified into “**Camera Mismatch**” and “**Scene mismatch**”. Camera mismatch refers to the mismatch of the capturing devices. It can be further divided into “**Camera Parameter Mismatch**” and “**Orientation Mismatch**”. Examples of the former are the camera intrinsic parameter such as focal length, resolution, principal point offset, signal-to-noise ratio[17], etc. The latter comes from the differences in the camera coordinates, baselines, and orientations, including the initial camera positions and the following movements (zooming, rotation and translation).

The scene mismatch is mainly responsible for defective visual perception. The visual reality is impaired as it differs from the way we are familiar with in the daily life. One example is the color temperature mismatch. For instance, the background scene is captured in

a sunny day, but the foreground object is pictured in a cloudy day. Another example is the light source, which leads to different shadow or reflection of objects. Our main objective in this study is the adjustment of camera mismatch in two 3D scenes. Table 1 lists the mismatch types we tackle in the system:

Table 3: Mismatch type under consideration

Mismatch source	Type	Explanation/Cause
Focal length Resolution Principal point offset Skew factor Baseline distance...	Camera parameter	Usage of different capture devices
Initial position of camera centre (relative to the assigned landing location)	Camera orientation	The user may assign the location of the object in background scene at his/her will.
Camera behavior	Camera orientation	The two scene may be captured under different camera trajectory (zooming is not considered).
Color temperature	Scene mismatch	Scene may be recorded under different light sources.
Floor-slope	Camera orientation Scene mismatch	The target object may not be captured with the same orientation (relative to the slope of the floor it lands on) of the background scene.

### 4.1.3 System Definition

Since there are few related references on studying the above issues, here we propose some terms:

- The “Target scene” or “Foreground scene” is composed of the target object with camera moving (zooming has not been taken into account). Hence, camera motion recovery (CMR) is needed, assuming there is no fiducial markers or special calibration objects. The format can be MVD or non-compressed stereo pair.

- “Target object” (“Foreground object”): A segmented moving object we are interested in, which lies in the target scene.
- “Target view” (“Foreground view”): The camera captured the target scene.
- The “Background scene” is in the form of MVD images or non-compressed video.
- Merging is used to produce a new scene consisting of the target object embedded in the background scene. The view is determined by the target view. Specifically, the background should follow when the target view changes.
- Eventually, the system outputs a stereo-pair created as a new 3D image/video.

To express our works in an easily digested manner, we will use these terms throughout the thesis.

#### **4.1.4 Overall System Flowchart**

The flowchart in Figure 23 shows our proposed 3D composition procedure. The right-side remarks annotate the mismatch types to be tackled by each stage (block). At the moment, we assume a person (the creator) is needed to assign key parameters in this procedure such as marking the floor and picking up the landing points (in the background scene) of the foreground objects.

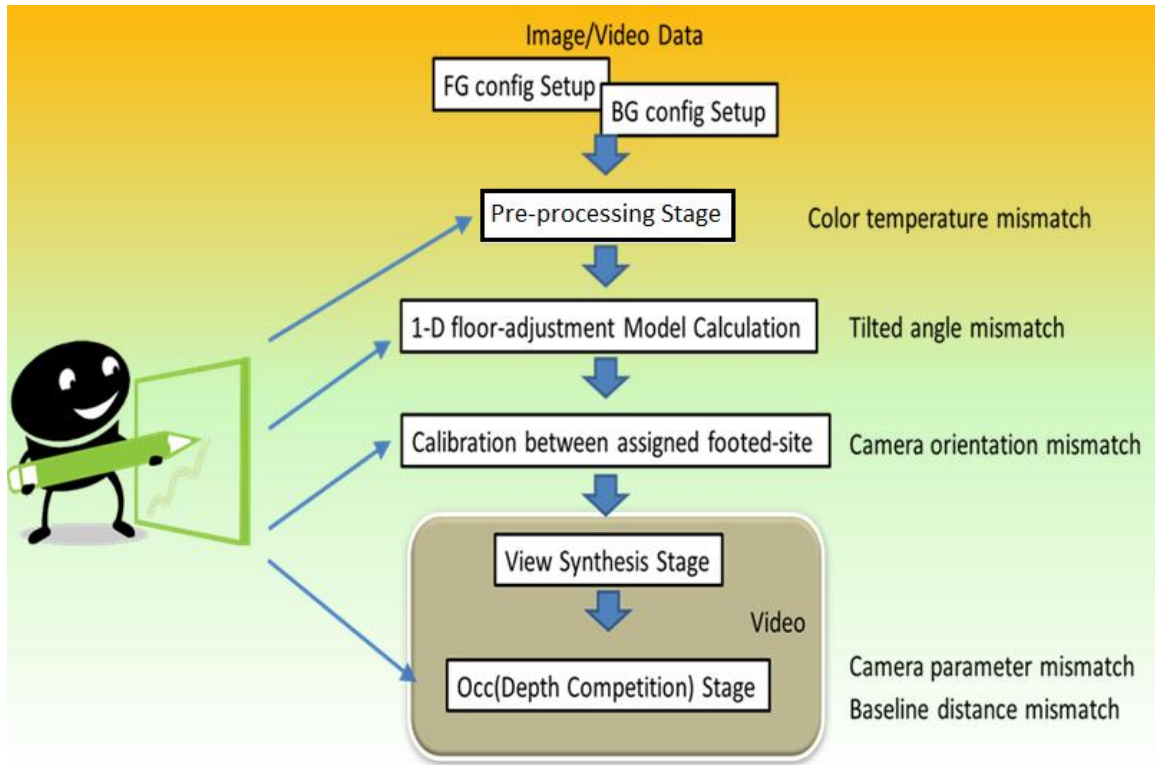


Figure 23: Overall system flowchart.

The processing stages are Input Data Stage, Pre-processing Stage, Floor-adjustment Stage, Assigned Footed-site Alignment Stage, View Synthesis Stage and Depth Competition Stage. This chapter will discuss the operations of each stages in the system. Since our main objective in this study is the adjustment of camera mismatch in two 3D scenes, the preprocessing stage (**matting** and **color grading stage**) for the system input is illustrated in the appendix.

When looking at Figure 22, we can hardly notice the key objects in the scene. When shooting a film, usually we focus more on the foreground object. Based on this assumption, we use the camera view capturing the foreground scene as the primary view.

## 4.2 Camera Orientation Adjustment by 1-D Floor-adjustment

### Model

#### 4.2.1 Relative Works of Orientation Mismatch

Figure 24 shows the problems of the orientation mismatch, where the car does not seem to



rest on the floor. To solve this problem, the geometric information of objects is a must to adjust the orientation of the camera (with respect to the ground-floor). However, in most cases, the orientation is unknown. Lalonde et al. [14] try to use some computer vision techniques to estimate the camera height and pose under certain assumptions. They tackle this problem by proposing an automatic algorithm to estimate the camera orientation with respect to the ground plane in each of the images. Assume that the object footing is visible and is not tilted from side to side and roughly orthogonal to the ground, then the camera pose can be estimated based on objects in the image with known heights. For instance, they set an object class (human) with a known height distribution, so that they can estimate the probability distribution of camera pose if the image contains a human.

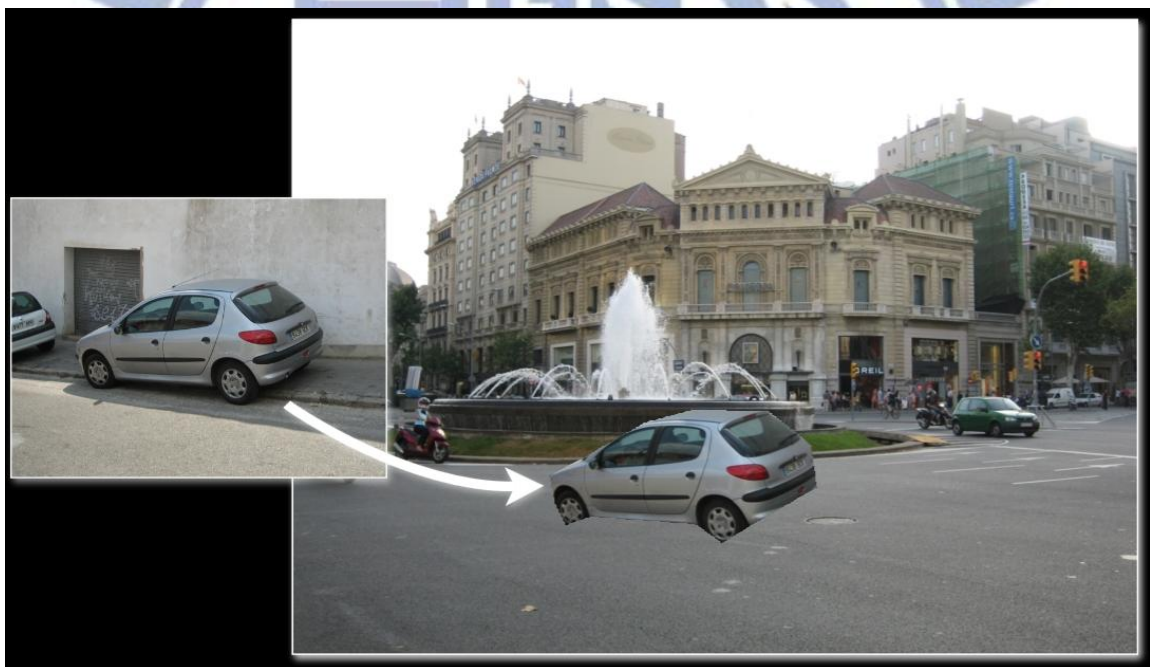


Figure 24: Problems of the Orientation Mismatch [14].

## 4.2.2 Proposed 1-D Floor-adjustment Model



Figure 25: User marks lines on the floor (green line on the left and red line on the right) for calculating the 1-D floor model. The left and right images are Lovebird1 and Poznan\_Street, respectively (MPEG test sequences).

Figure 25 shows two scenes captured by two sets of cameras. Their camera orientations are not identical. If we use two persons in the left picture (Lovbird1) as the Target Objects, how do we adjust the orientation of the right picture camera to match that of the left camera? We assume the ground (floor) is our horizontal reference. Figure 26 shows the geometric relationship between the floor and the camera orientation. Let the optical axis of the camera be horizontal, and the ground plane has an angle  $\alpha$  relative to the optical axis.  $H$  represents the distance from optical center to the floor, perpendicular to the optical axis. For each pixel on the marked 1-D line (the green and red lines in Figure 25),  $z$  refers to its linear depth value,  $\theta$  stands for the angle of a pixel on the image plane relative to the optical axis, and  $X$  is the row index along the vertical X-axis on the image plane, whose origin is at the same row of the principal point on the image plane. For example, in Figure 26, the principal point has the same row index as the green pixel.  $X$  of the three pixels from top to bottom (green-red-blue) is 0, 1, and 2, respectively.

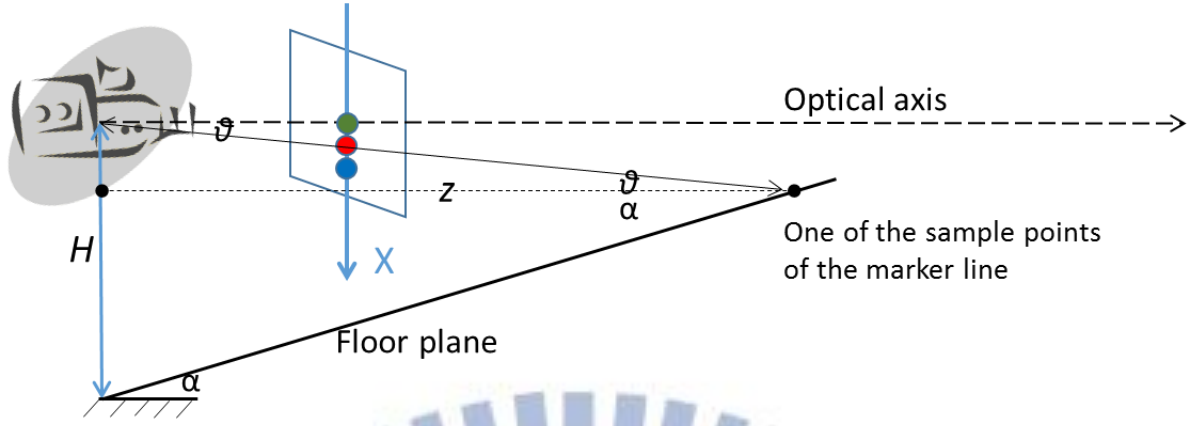


Figure 26: 1D camera orientation vs. floor model.

In Figure 26, we can then observe that

$$z = \frac{H}{\tan \theta + \tan \alpha} = \frac{H}{\left(\frac{X}{f}\right) + \tan \alpha} \quad (12)$$

In this case, the nonlinear regression problems can be moved to a linear domain by a suitable transformation. Thus, we define the target  $t$  as inverse of  $z$ .

$$\frac{1}{z} = \left(\frac{1}{fH}\right)X + \left(\frac{\tan \alpha}{H}\right) \quad (13)$$

and

$$t = \frac{1}{z} = \Phi \omega = [1 \quad X] \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} \quad (14)$$

If the depth  $z$  and pixel index  $X$  are given, based on the marked 1-D lines on the floor, we can estimate the vector  $\omega$  by the maximum likelihood regression in both the foreground and the background scenes, respectively, which is achieved by eq.(15).

$$\omega_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t, \quad (15)$$

where  $\Phi$  is the design matrix<sup>3</sup>, and  $\Phi^+ = (\Phi^T \Phi)^{-1} \Phi^T$  is known as the Moore-Penrose

<sup>3</sup> Design matrix is a defined matrix formed by the data (samples) projected to various bases for the regression problem.

pseudo-inverse of the matrix  $\Phi$ .

Finally, the angle between the floor and the optical axis can be recovered by

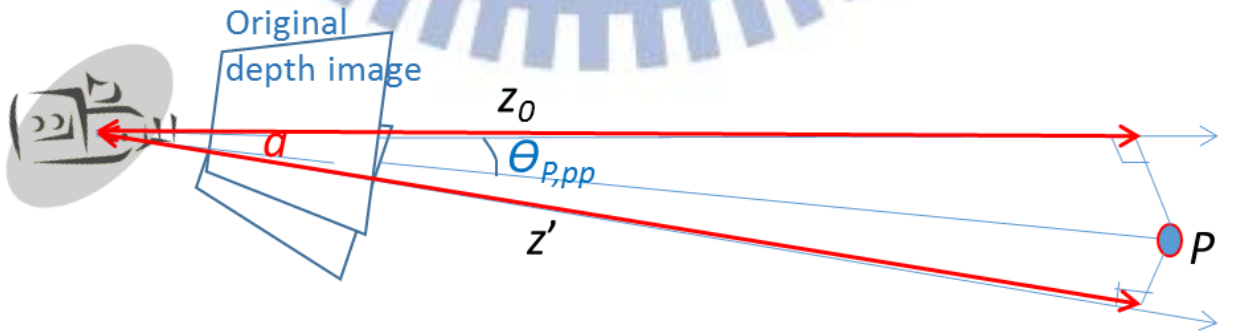
$$\alpha = \tan^{-1}\left(\frac{\omega_0}{\omega_1 f}\right) \quad (16)$$

Based on the 1-D floor model described in the above model, we can compensate the camera orientation mismatch in two scenes. Essentially,  $R_{fg,bg}(r_x, r_y, r_z)$  is the 3-D rotation matrix between the two camera orientations. Assume two camera are horizontally aligned, then  $r_y = r_z = 0$ ,  $r_x = \alpha_{fg} - \alpha_{bg}$ .  $R_{fg,bg}$  is the rotation around the  $X$ -axis to compensate the mismatch of the orientation based on the common ground floor plane assumption in two scenes.

### 4.2.3 Depth Correction under Rotation

However, the pure rotation changes the depth map because the depth value is the distance projected to the axis. To maintain the fidelity of the depth image of the virtual view, we have to correct the depth values. Figure 27 shows that the original depth value ( $z_0$ ) of a particular 3D point<sup>4</sup>  $P$  changes after a rotation transform  $R_{fg,bg}(a, 0, 0)$  is applied, which gives

$$z' = z_0 \cdot f(\theta, a) \quad (17)$$



<sup>4</sup> In this paper, uppercase  $P$  stands for a 3D point, and the lowercase  $p$  for a 2D point, in the homogeneous coordinates.

Figure 27: Geometry relation of pure rotation

$$f(\overrightarrow{\theta}_{P,pp}, a) = \frac{\cos(\|\overrightarrow{\theta}_{P,pp} - \begin{bmatrix} 0 \\ a \end{bmatrix}\|)}{\cos(\|\overrightarrow{\theta}_{P,pp}\|)}, \quad (18)$$

where  $a$  is the rotated angle,  $z'$  represents the new depth value, and  $z_0$  is the original depth value.  $\overrightarrow{\theta}_{P,pp} = [\theta_y \ \theta_x]^T$  is a two-dimensional vector consisting of the angles of the pixel  $P$  with respect to the original camera principal point/axis (' $pp'$ ') in the  $y$  and  $x$  direction. The correction is achieved by multiplied with a scalar, which is decided by both the optical axis (principal point offset) and the location of the target pixel. The depth correction function  $f$  in eq.(18) can be interpreted as two steps. First, the denominator is to convert the original linear depth value to the distance from the camera to the point  $P$ . Afterwards, the distance is then projected to the new optical axis. Note that the depth value of each sequence should be scaled to the same unit. For the MPEG test sequences, the unit information in each sequence can be calculated from the given baseline distance.

An experiment is conducted for the validation of the 1-D floor-adjustment model by using Kinect devices. The test are shown using Figure 28 to Figure 31. In Figure 28, we mark the ground floor on the reference image (top) and the target image (bottom). The camera pose and height indeed can be regained and warped to the target view.



Figure 28: The top is the reference image, and the bottom is the target image. The lines indicated the referenced 1-D line, which is on the desktop (so-called the floor) the drink stands on.

From the proposed model, we can derive  $\alpha$  (camera pose) and  $H$  (Camera height) for each scene. In Figure 29, **Camera2** represents the target Kinect view, and **Camera1** for the reference one. If we assume the optical axis of **Camera2** be horizontal, and the floor has an angle  $\alpha_2$ . **Camera1** is not parallel and originally looks downward. After homography matrix  $R_{\text{floor}}$ , **Camera1** is rotated to be horizontal. As discussed earlier, the virtual view depth map can be calculated by the homography transform of the original depth map with the depth update formula (eq.(17)). The result of updated depth map is shown in Figure 30. The left depth image is the virtual depth rotated from the reference view, and the right is the result

with correction. The image height is enlarged to show the depth after the rotation. The origin is on the left-top corner.

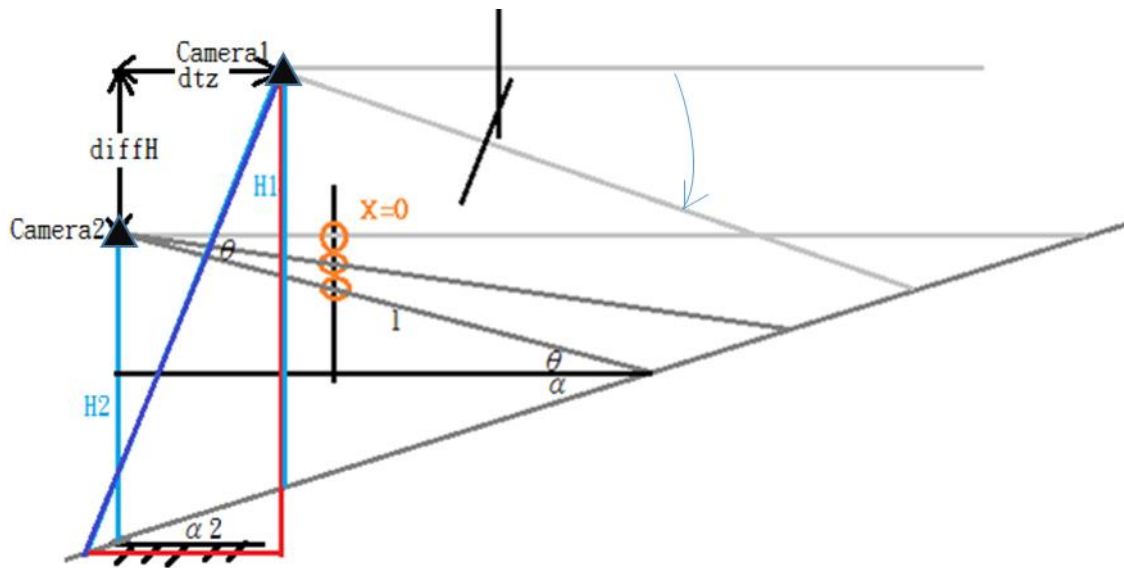


Figure 29: Calibration for 1-D floor-adjustment model.

‘▲’ stands for the camera position.

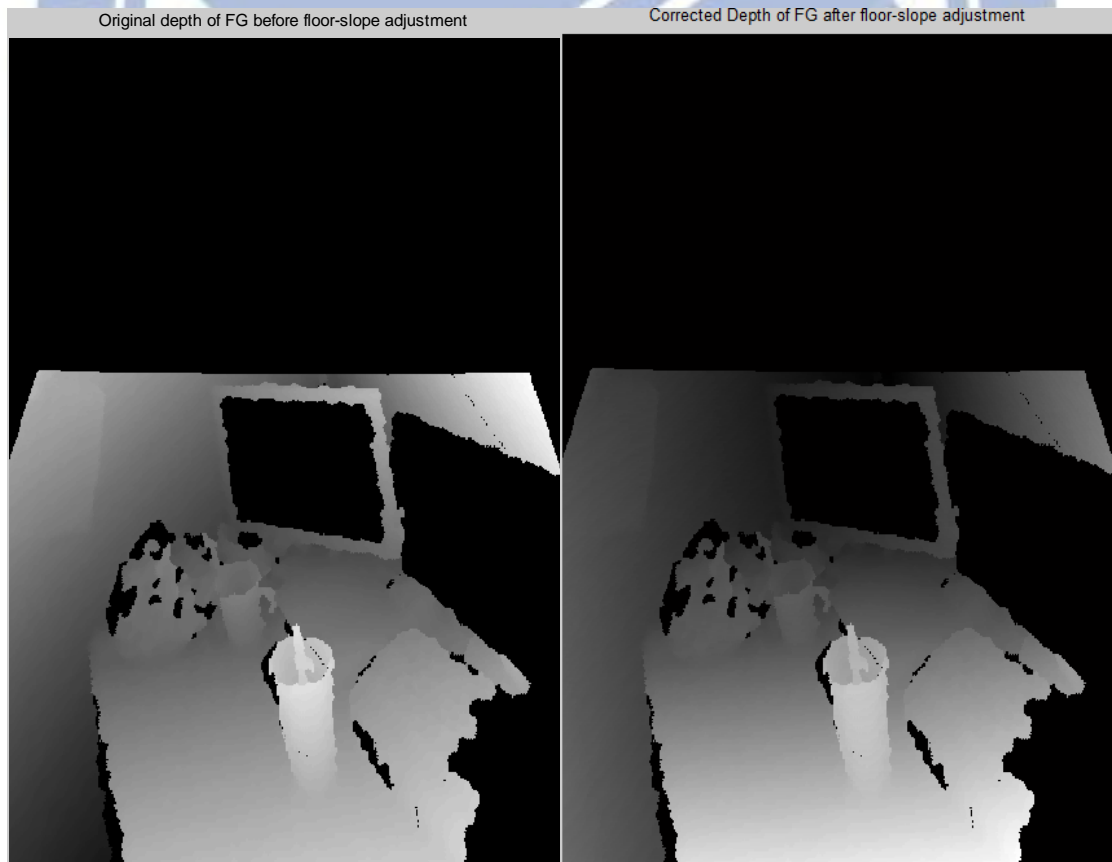


Figure 30: Before (Left) and after (Right) applying the depth map updated formula.

The difference of height is:

$$diffH = |H_2 - (H_1 \cos(\alpha_2 - \alpha_1) - H_1 \sin(\alpha_2 - \alpha_1) \tan \alpha_2 + dt_z \tan \alpha_2)|. \quad (19)$$

$dt_z$  and  $diffH$  respectively are the translations in z-direction and in y-direction. In this case, which is created on purpose, the difference of the floor angle is 37 degree and  $diffH$  is approximately 47.4cm. The result of view synthesis is shown in Figure 31. The right image is the target image, and the left one is the warped texture from the source image. Holes are shown in black and green color. The black region indicates the holes of the reference depth map, and the green region indicates the holes result from the warping process. Despite the fact that the drink appears tilted from side to side, we can see that through 1-D floor-adjustment model, we not only regain the camera pose and height, but compensate the orientation mismatch quite successfully. In the above process, we did not use any calibration process. The desktop (floor) that the drink stands on still has a rotation around z-axis. Since the model is limited to 1-D, the synthesized view has a tilt that is not compensated

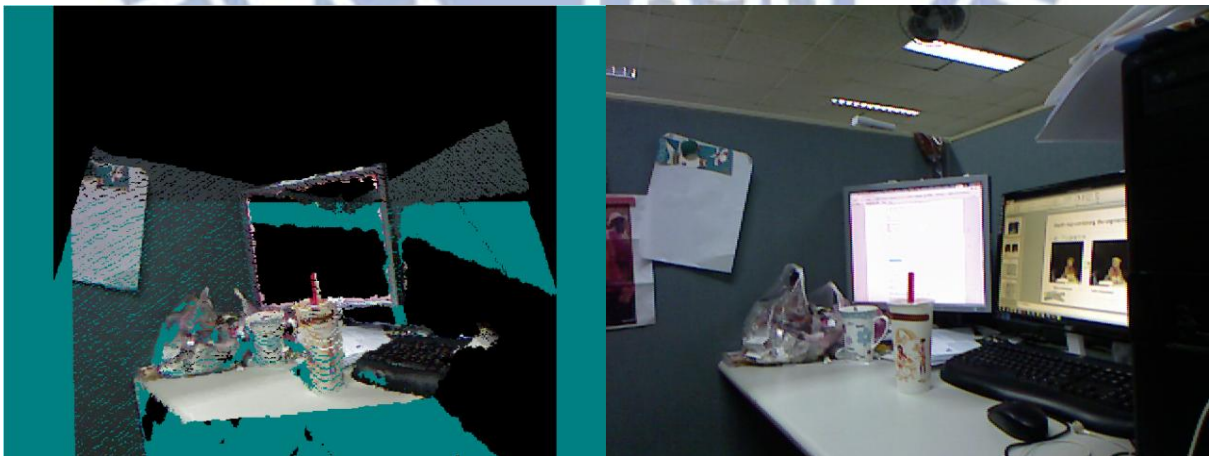


Figure 31: After warping to the identical relative position The left shows the synthesized view using the information of camera pose and height. The right shows the target view.



Also, we test our camera orientation adjustment in Figure 32. The synthesized pictures with and without the floor-modeling and adjustment are shown. Because the camera orientations of these two scenes are quite close, the compensation effect is not as obvious as that in Figure 30. And the 1-D floor-model does improve the photo realism.



Figure 32: Upper images are the stereo composition results without floor-adjustment; Lower images are with floor-adjustment stage, which appears more natural.

### 4.3 Camera Orientation Alignment with Assigned Landing Point

Given the MVD format video, it offers possibility to render the view of the background scene by the geometry information. The synthesis background view is determined by the “**Landing point**”, indicating the location where the target (foreground) object places. The following passage describes the detailed procedure.

### 4.3.1 User-assistant Landing Point Selection

The user can place the object landing point at a properly selected location in the background scene, on the reference view (left view).  $P_{toe}$  indicates the “toe” of the foreground object, which is automatically selected from the lowest pixels of the object (the rightest, in our case).

$P_{landing}$  is the target landing point in the image of the background scene, which is manually picked at one’s will.  $P_{landing}$  and  $P_{toe}$  have associated depth values (linear, not quantized)  $dr_{landing}$  and  $dr_{toe}$ , respectively. The warping parameters for the assigned landing point are given below.

$$P_0 = ((K_{fg})^{-1} \cdot p_{toe}) dr_{toe} \quad (20)$$

$$P_1 = R_{fg.bg}^T ((K_{bg})^{-1} \cdot p_{landing}) dr_{landing} \quad (21)$$

$$dt = \begin{bmatrix} dt_x \\ dt_y \\ dt_z \end{bmatrix} = P_1 - P_0 \quad (22)$$

$R_{fg.bg}$  is the rotation derived from the previous floor-model, which records the camera rotation mismatch around the x-axis.  $K$  is the camera intrinsic matrix for each scene and  $dt$  is the alignment vector, which is the translation in  $x$ ,  $y$ , and  $z$  directions. The warping parameter aligns the “toe” of the target object with the assigned landing point in the 3D homogeneous coordinate.

Figure 33 shows the result of a single view (left view) of several landing points. Once we picked a landing point, we need to adjust the background scene, so that the depth  $dr_{landing}$  matches  $dr_{toe}$ . We thus need to move the camera (of the background scene) forward. In other words, the alignment process first estimates the relative position of the foreground camera.

The synthesized camera center locates at the virtual plane parallel to the adjusted background floor. Due to the camera viewpoint shift, the new background scene is synthesized using VSRS. We can see that certain selection of landing points may degrade the image quality of the background due to the view synthesis process. The synthesized view becomes worse when it is far from the reference view.





Figure 33: Some examples of the picked landing points. The camera location needs to be

adjusted to match the assigned ground point. The top figure is the background scene. The red dots in the background are the picked landing points. The result is shown in the order of red points (landing point) **from right to left**.

### 4.3.2 Depth Prediction of the Landing Point when Occluded

When the user selects the landing point in the reference image, the interested landing point may be occluded. For example, the user wants to have the couple in the lovebird\_1 sequence stand right behind the gray car in Figure 33. In this case,  $dr_{landing}$  is estimated by the 1-D floor model derived from the previous stage to calculate a reasonable depth value. The technique enriches the freedom of composition. Specifically, derived from eq.(14),  $dr_{landing}$  is estimated as follows.

$$dr_{landing} = \frac{1}{\omega_1 X_{landing} + \omega_0} \quad ( 23 )$$

## 4.4 View Synthesis Stage

### 4.4.1 Virtual-view-based rendering by modified VSRS

The left view is used to select landing point and draw line mark on the floor plane; then the left background view is synthesized using the rotation and translation matrices derived from the previous stages. The right view of the background scene is treated as a 1-D parallel view with a baseline distance identical to the foreground stereo pairs. We use VSRS (View Synthesis Reference Software) version 3.5 as the view generation tool. From the given set of multiple views, we pick up the nearest reference left and right views for better synthesis quality. An example is shown in Figure 34.



Figure 34: An example of final stereo composition result. The background is Pozna\_CarPark [18] (MPEG test sequences).

#### 4.4.2 Virtual Depth Generation

We need to modify the software to generate the virtual-view depth map. As we have mentioned in sec 3.2.1, the virtual depth map is generated by the following backward warping.

$$\begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} = H(z_q) \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} \quad or \quad \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = H^{-1}(z_q) \begin{bmatrix} u_v \\ v_v \\ 1 \end{bmatrix} \quad (24)$$

$$H(z_q) = A + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \otimes \left( \frac{z_q}{N-1} \cdot (z_{near}^{-1} - z_{far}^{-1}) + z_{far}^{-1} \right) \cdot b \quad (25)$$

The subscript  $v$  is for virtual view and  $r$  for reference view.  $z_q$  refers to the quantized depth value (intensity from 0 to 255, totally  $N$  discrete layers).  $[z_{near} z_{far}]$  stands for the range of linear depth values. The warping process can be treated as a homography transform of the quantized depth value from the reference view. In VSRS, its backward texture warping is implemented by looking up to the pre-calculated homography table, which records  $H$  of  $N$  discrete depth layers. Then, the virtual depth intensities are warped from the two reference depth images (which are not with correct depth values). The depth intensities are used for the

index when it looks up the table. To regain the correct virtual depth (for the depth competition stage), we need to develop on our own.

In addition to the rotation  $R_{floor}$ , there is  $dt$  as well. Note that  $dt_x$  and  $dt_y$  do not affect the depth value. Similar to the depth updated function in pure rotation, but the function in terms of the virtual view image index  $X_{new}$  :

$$z' = (z_0 - dt_z) \cdot f(\overrightarrow{\theta_{X_{new}}} + \begin{bmatrix} 0 \\ \alpha \end{bmatrix}, \alpha) \quad (26)$$

,where  $f$  is the update function identical to eq(18).

## 4.5 Inter-Occlusion of the Stereo Composition View

In comparison with the conventional composition without geometry information, we need to maintain a valid inter-occlusion relation. Therefore, we will discuss the depth competition problem and solution in our system.

### 4.5.1 Depth Competition under User-assistance

The foreground object can occlude, or be occluded by the background scene. Using VSRS, we are able to regain the depth map of the virtual view. By cropping the synthesized background scene around the principal point and converting the depth value to the same measuring unit, the inter-occlusion relation can be regarded as a simple depth competition problem for each pixel. We eventually choose the scene (pixel) with the nearer depth value. However, in the camera orientation alignment stage, the landing point is arbitrarily assigned and it determines the camera orientation, and this process may contain errors. In Figure 25, the left image shows the automatic choice of the foreground (people) pixels and some pixels are missing. The lower part of foreground is cropped, resulting in the wrong estimated depth value and  $dt_z$ .

Hence, we let the user to determine the occlusion threshold in this stage. It is an interactive process. The user can see the real-time occlusion results after picking up a threshold. One instant result is shown in Figure 35.

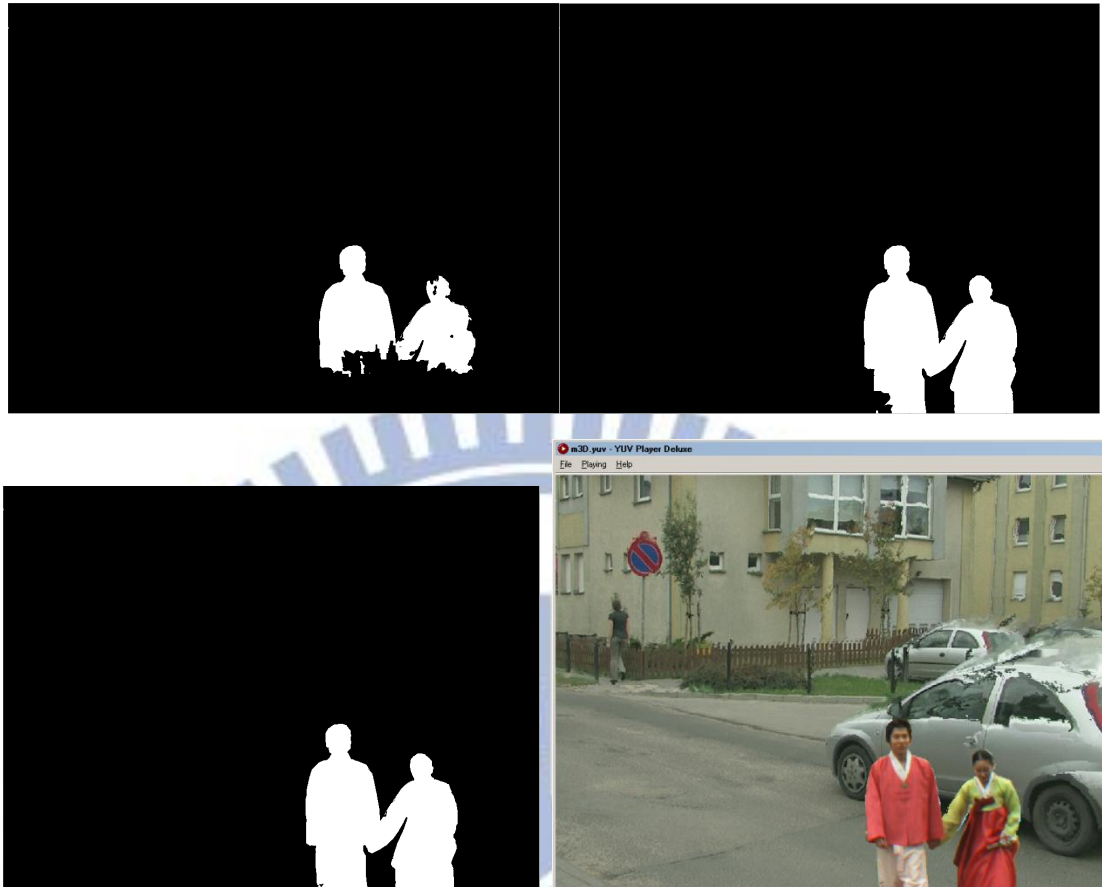


Figure 35: Depth competition map under different racing threshold (top-left:no thresholding / top-right:  $th=-2000$  / bottom-left:  $th=-2500$ ). Bottom-right is the bottom-left 's depth competition result( $th=-2500$ ).

#### 4.5.2 Hole Filling

Practically, the depth image of virtual view contains holes due to the warping process (the reason is described in 3.2.2 D). An example is shown in Figure 36. There are several existing algorithms for hole-filling. However, our case is slightly different. Most existing hole-filling techniques aim for better quality in texture backward warping, while our depth image is for the depth competition.



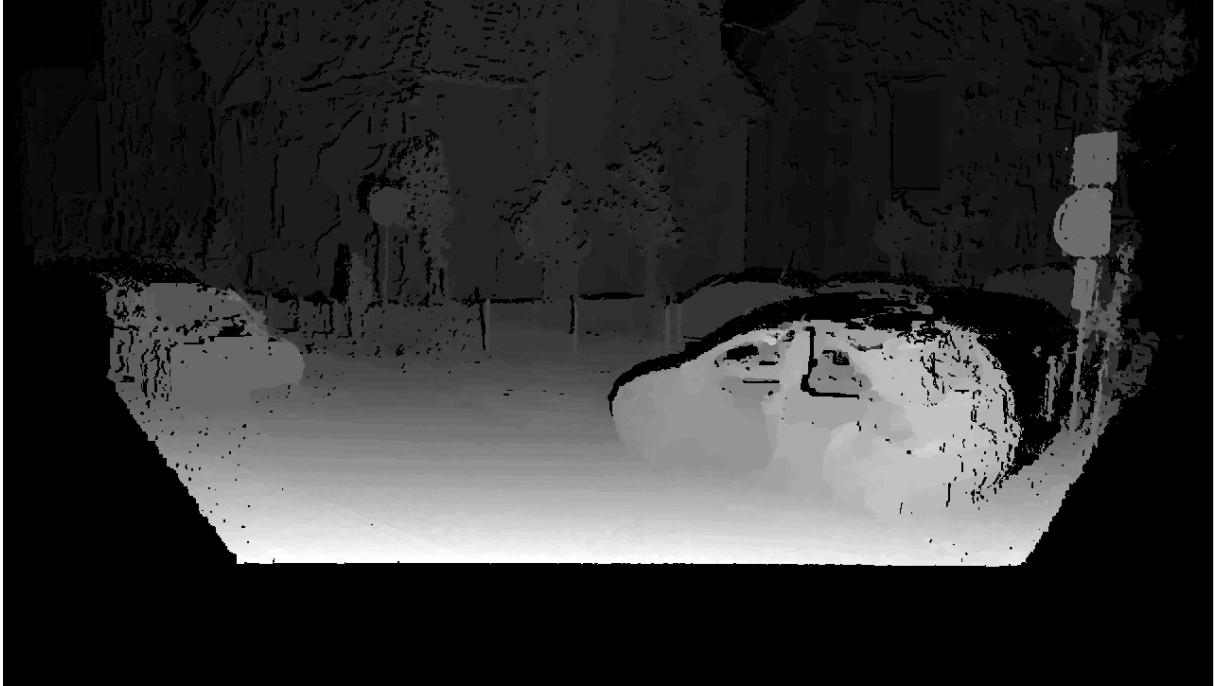


Figure 36: Large holes appear due to warping process

Our hole-filling procedure is as follows.

1. Group the black pixels together if they are connected (4-neighbors).
2. For each hole (as in the lower figure of Figure 37), mark all its 4-neighbors (Pixels around the hole).
3. Calculate the histogram.(as in Figure 38).
4. Eliminate those bins (of the histogram) whose values are too small.
5. Select the depth value with the smallest (furthest) value.

We examine the histogram of pixels in a hole. If the pixel number of the same value is too small, it may be an outlier (noise). Based on our experiments, we eliminate the bins less than 5% of the total number of a connected hole. A histogram example with outliers is shown in Figure 38. The leftiest bin is thus ignored (excluded). Afterwards, we will select the bins with the smallest (furthest) depth value since in most cases, the background region is occluded.

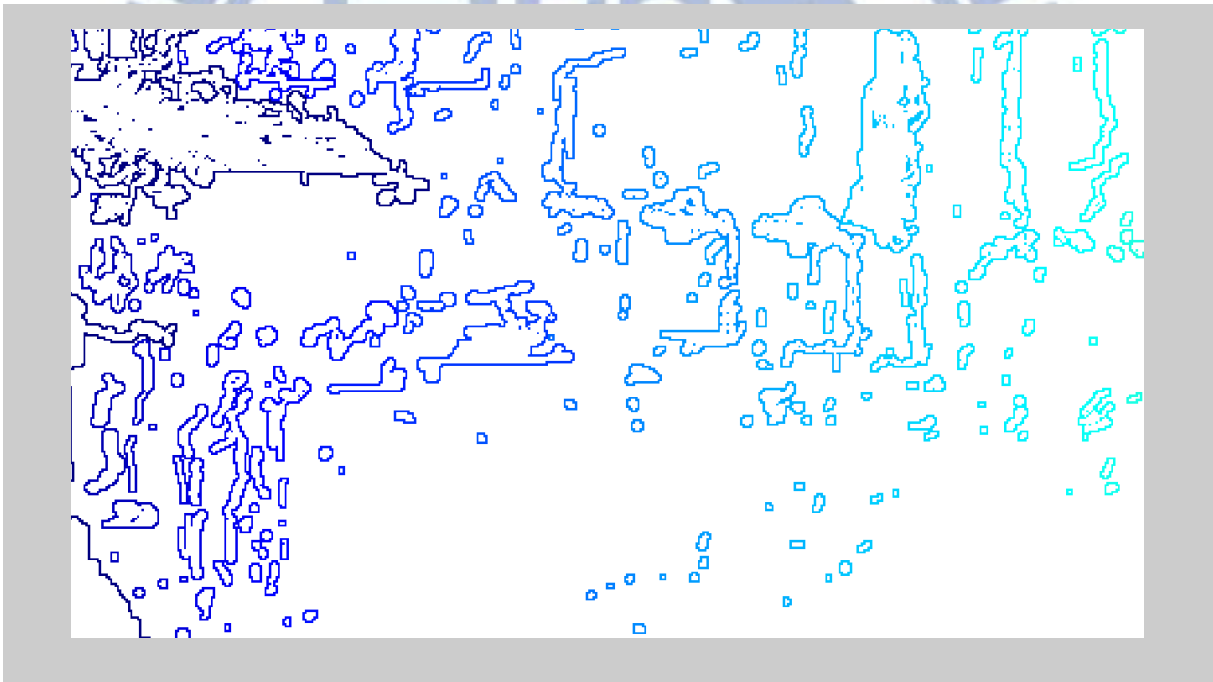
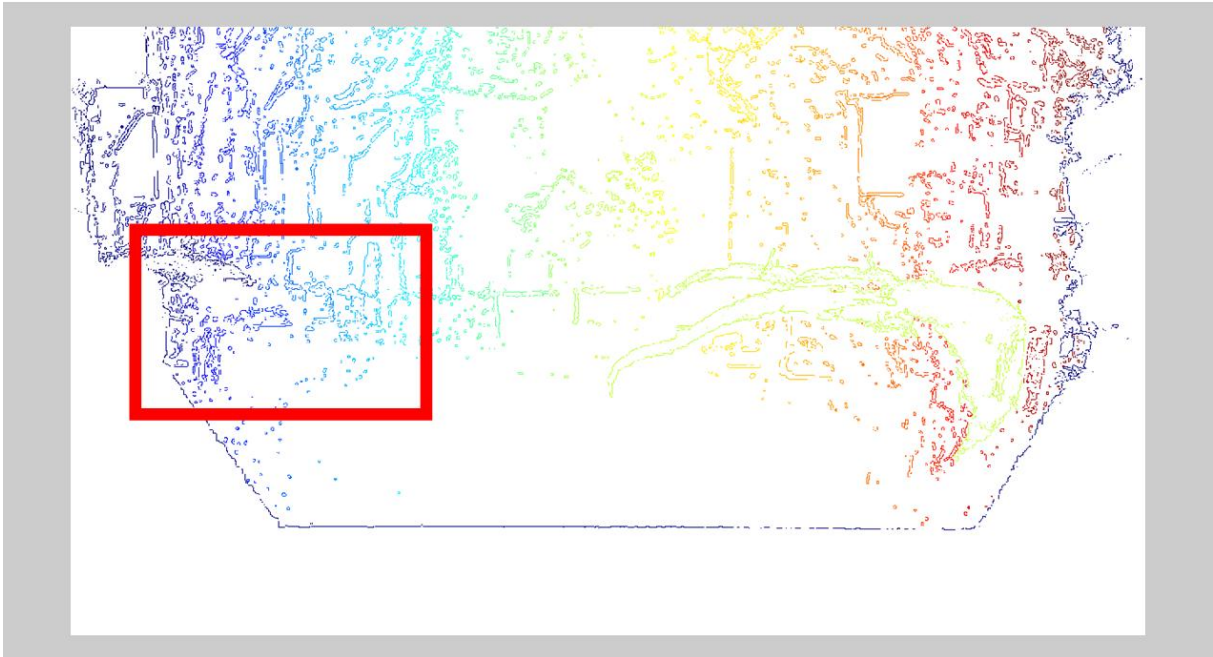


Figure 37: Top figure shows all the connected holes. The lower figure is the enlarged region marked by the red rectangle.

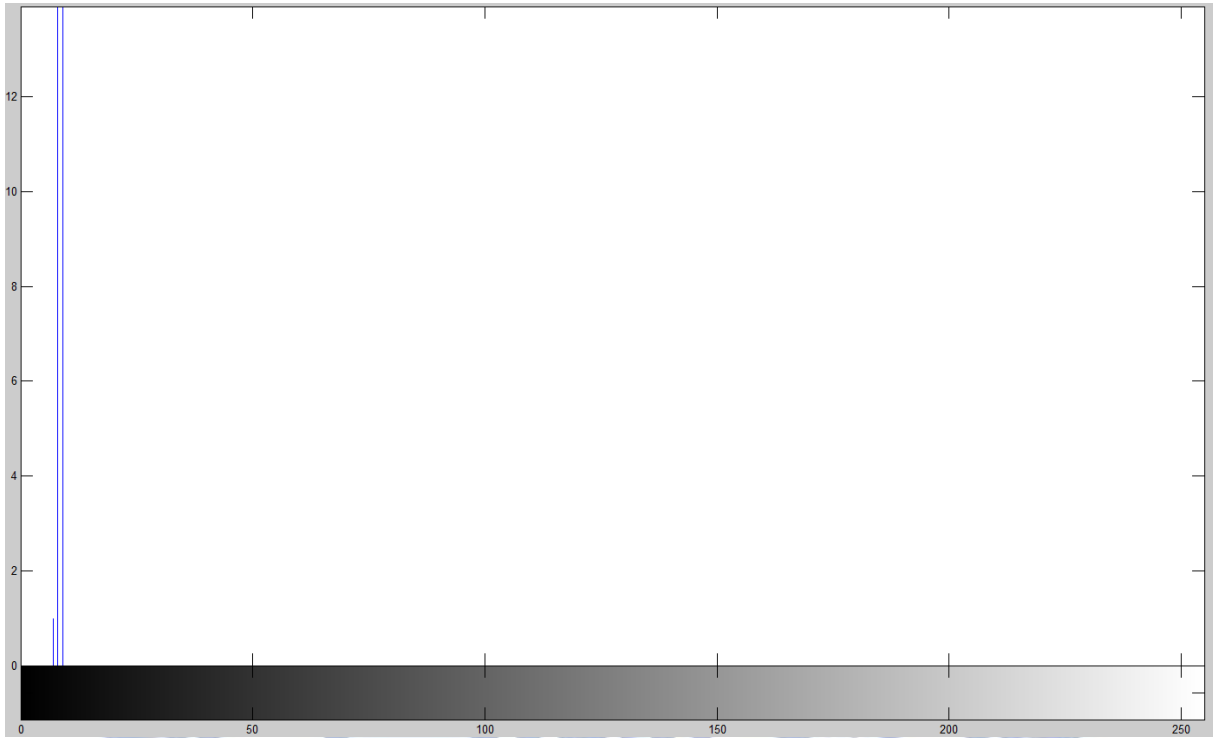


Figure 38: Histogram of the neighboring pixels of a hole. The horizontal axis is depth value from 0~255. The vertical axis is the pixel number.

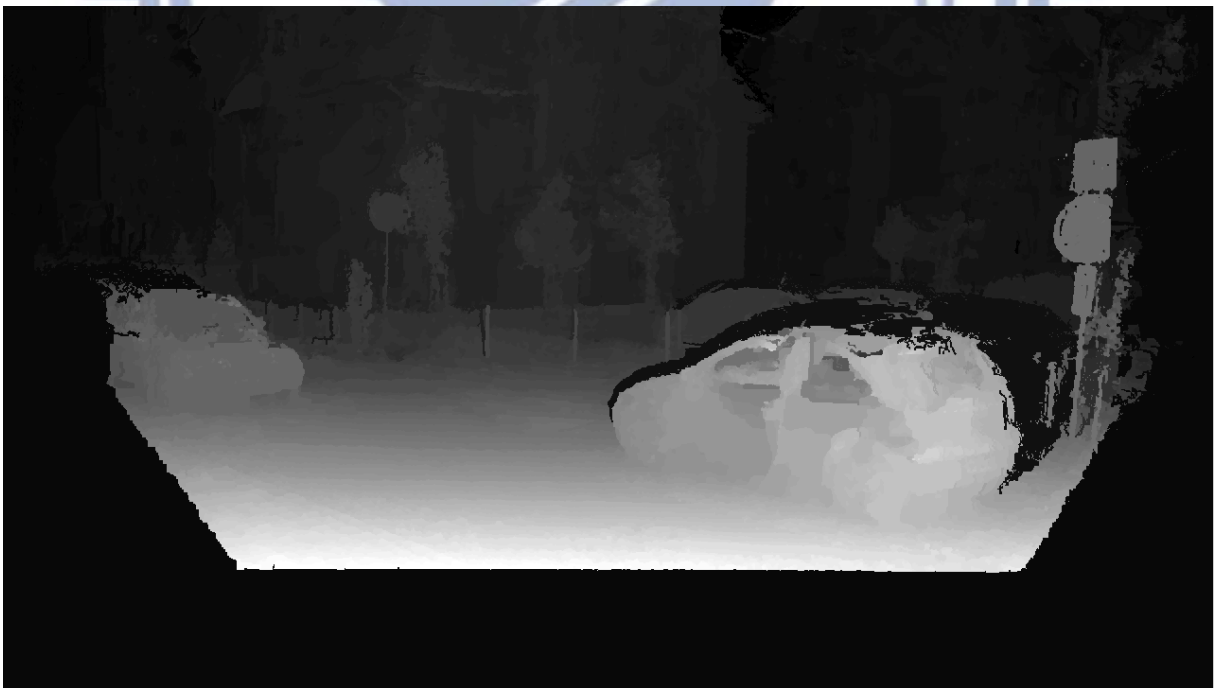


Figure 39: Result after applying our hole-filling techniques

Essentially, we choose the furthest depth values of the neighbors of a hole pixel except for the outliers. We observe that a large hole at the lower part of the image is filled with a

small value (8 in Figure 39) as it connects to the background. This kind of hole results from the rotation and translation in  $z$ -direction. It is not correct but it is fine because the region usually belongs to the floor or the scene's boundary. They are often occluded.

## 4.6 The Extension to Video

The 3D image compositing techniques can be further extended to video. One key is to maintain the orientation relationship in the temporal domain. For example, if the foreground camera is moving, the synthesized background should follow the camera movement. Another issue is that the background is not captured by a static camera. In short, the composite video shall jointly consider the behavior of both foreground and background cameras. We first try to recover the camera trajectory and use it in our system.

### 4.6.1 Camera Matchmoving

This stage tries to recover the camera motion for the foreground view and/or the background view. So far, we have estimated all the translation and merging parameters for matching two scenes. Hence, if we can estimate the camera matchmoving, we can update these setting along time axis from the first frame and use it in the following frames.

Camera matchmoving is also called ego-motion, camera motion recovery, or camera trajectory recovery. Sometimes it is also called image registration or structure from motion (sfm) studied from a different perspective (but shares similar ideas). There are mainly two approaches. First, we convert the motion problem to a stereo problem and the correspondence between a number of feature points (e.g., corners) in the image at time  $t$  to the image at time  $t+dt$ . Second, we can compute the optical flow and use its inter-frame shifts to deduce three dimensional information about the scene and the motion. The first approach, which leads to a sparse 3D structure, is known as the matching methods, and the second approach, which leads to a dense 3D structure, is known as the differential methods.

We favor the matching methods because of its robustness and low computational complexity. The procedure is as follows. First, a **feature detection** (Harris descriptor is used here) followed by **RANSAC** (RANdom SAmple Consensus) to find the matching points (inliers) in two frames to estimate the fundamental matrix.





Figure 40: Found putative matches from two images.

After estimating the fundamental matrix from the matching inliers, we decompose the relative projection matrix by the SVD (or SR) decomposition to obtain the relative rotation and the translation matrix. Consider a camera matrix decomposed as  $P = K[R | t]$ , and let  $x = PX$  be a point in the image. If the calibration matrix  $K$  is known, then we may apply its inverse to the point  $x$  to obtain the point in the normalized coordinate  $\hat{x} = K^{-1}x$ . Then  $\hat{x} = [R | t]X$ . The camera matrix  $K^{-1}P = [R | t]$  is called a normalized camera matrix. Then, we consider a pair of normalized camera matrices  $P = [I | 0]$  and  $P' = [R | t]$ . The fundamental matrix corresponding to the pair of normalized cameras is called essential matrix. It has the form

$$E = [t]_{\times} R = R [R^T t]_{\times}. \quad (27)$$

In fact, essential matrix follows the relationship with fundamental matrix as follows:

$$E = K'^T F K \quad (28)$$

The essential matrix  $E$  has only five degrees of freedom: both the rotation matrix  $R$  and the translation vector  $t$  have three degrees of freedom, but there is an overall scale ambiguity.

These steps as well as the extraction of camera parameters from the essential matrix can be found in [5].

In this process, the reconstruction ambiguity problem occurs when we estimate the translation vector. The problem is illustrated by Figure 41. Assume two images with a known matching point. We can see that the two views can be in pair of {position1, position2} or {position1, position2'}. In the latter case, the object is located at  $O'$ . This problem (reconstruction ambiguity) is shown in Figure 42. We can interpret the problem in this way: The box in the pictures may be a gigantic one, or it can be a tiny one. We have no exact idea about the scale of each sequence's translation vector, so usually we only obtain the normalized translation vector.

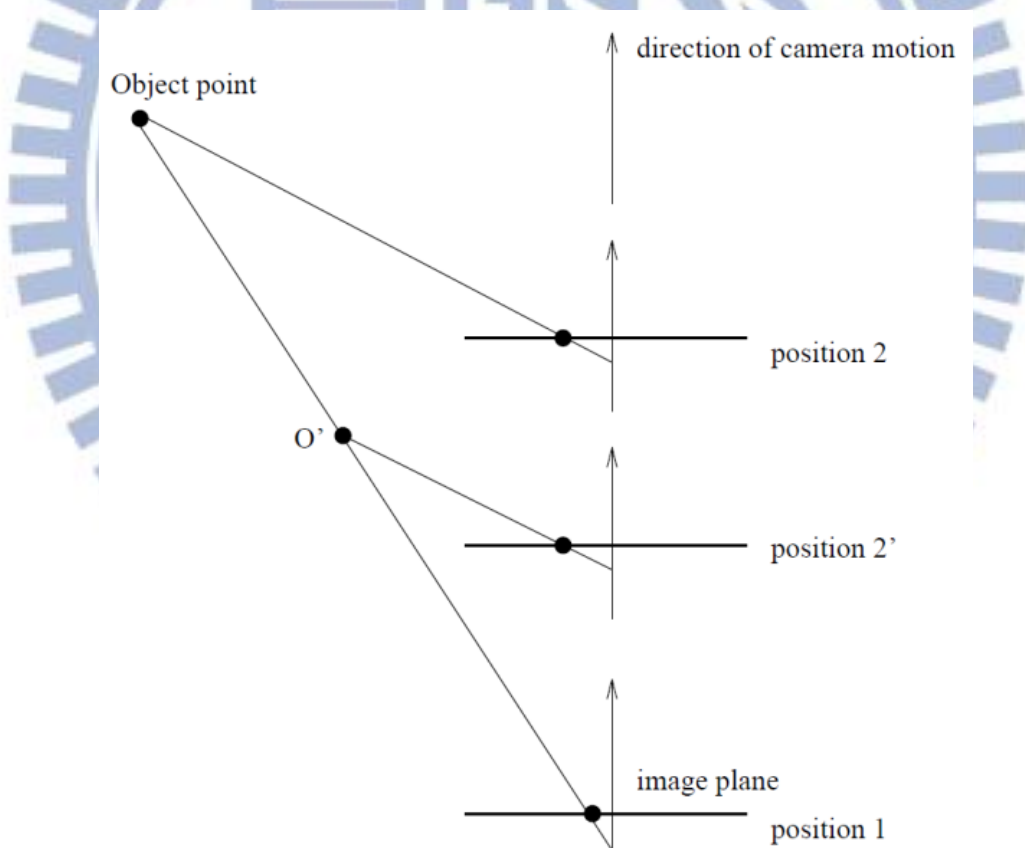


Figure 41: Reconstruction ambiguity [5].

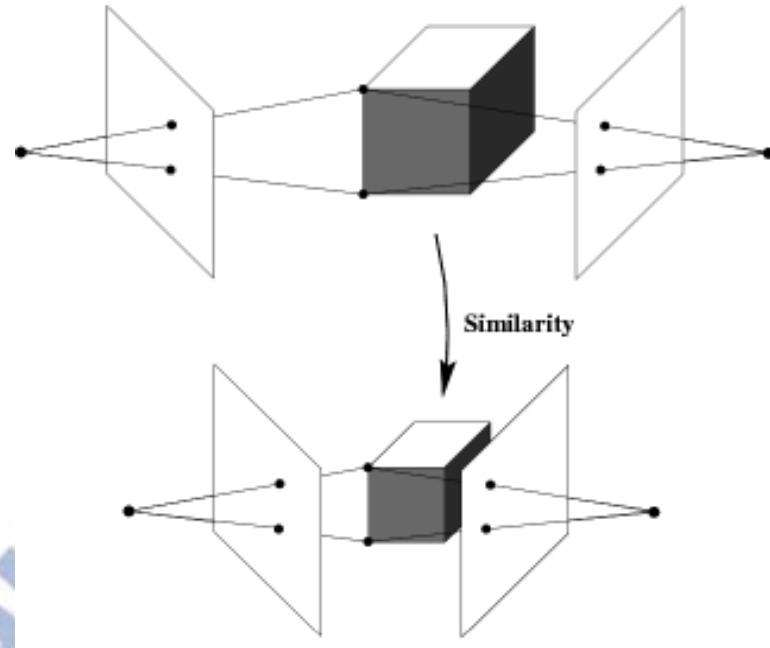










Figure 42: Reconstruction ambiguity even in the calibrated case: the actual 3D structure may be under a similarity transform.

The disparity warping for parallel and identical intrinsic matrix is as follows:

$$disparity = \frac{f_x \Delta t}{z_r}. \quad (29)$$

$f_x$  is the focal length and  $\Delta t$  is the translation vector. Since the matching inliers can offer the disparity information, and the depth value  $z_r$  is also known from the input depth map. We can estimate  $\Delta t$  by the maximum likelihood regression technique for minimum squared error (MSE). To reduce estimation noises, we only take disparity of one of the direction component ( $x$  or  $y$ ), depending on which one is dominant. Eventually, the scale for translation vector is determined by the dominant component of the estimated  $\Delta t$ . The result of motion compensation by view synthesis is shown in Figure 43. However, the frame-by-frame motion estimation is somewhat noisy. Hence, the compensated video looks shaking when playing back.



Frame Number	Original Sequence	Motion Compensated Sequence
1		
21		
41		
61		

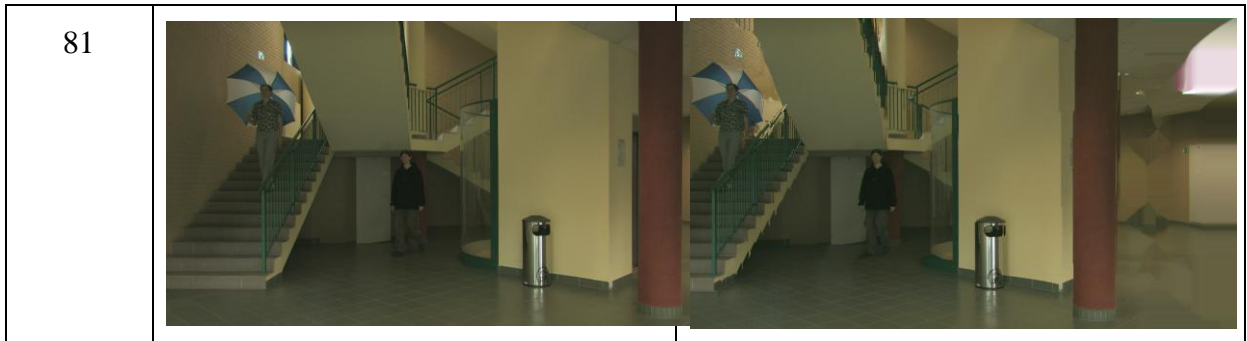


Figure 43: Motion compensation result of Poznan\_Hall2 sequence



Figure 44: Composition result by combining static foreground scene and moving background scene. The top is the stereo pair in frame #1. The bottom is the stereo pair in frame #51.

## Chapter 5. Conclusion and Future Work

### 5.1 Conclusion

The goal of the proposed VSRS based stereo video composition system is to produce good visual quality composite 3-D contents based on the available and estimated geometry information. The challenge is to resolve various types of mismatches between two sets of original 3D scenes. In this study, our focus is on the geometric relationship of the camera and the scenes. We construct a 1-D floor model to aid the camera orientation adjustment. We derive the geometric transforms for creating the new virtual viewpoint so that the new background scenes can be synthesized to match the foreground objects.

### 5.2 Future Work

There are still several important future tasks on this subject. Some of them are listed below.

- Friendly GUI interface (with 2-D display) or HCI with depth sensor (with 3-D display).
- Video matting may be used for better extracting the foreground. Both the scene depth and texture information can help in foreground segmentation.
- Use key frames for motion estimation instead of using the previous frame. This may increase the motion estimation accuracy.

# Appendix

Our main focus in this thesis is the geometry consistency in composite scenes. To achieve this aim, we also work on issues such as segmentation and photo realism (color). We describe these tools in Appendix A & B.

## A. Segmentation and Matting

In firm working, the image composition system takes as input of segmented foreground objects using chroma keying techniques. However, limited to the MPEG 3dvc dataset, we have to do segmentation / matting by ourselves. We use the tool of Learning Based Digital Matting method proposed by Yuanjie and Kambhamettu [20] to extract our target objects. We will briefly introduce how matting works and illustrate our methods.

### A.1 Matting

Digital matting refers to the process of extracting a foreground object image  $F$  along with its opacity mask  $\alpha$  (typically called “alpha matte”) from a given digital image  $I$ , assuming that  $I$  is formed by linearly blending  $F$  and a background image  $B$  using  $\alpha$  :

$$I = \alpha F + (1 - \alpha)B \quad (30)$$

The input and output of matting is shown in Figure 45. More detail can be found in [20]



Figure 45: Supervised image matting by using trimap. The left figure is the reference image.

The middle figure shows the trimap as input, which labels some region that is definitely foreground (white) or background (black). The right figure shows its output (alpha matte),

which is usually followed by a compositing process to create a new image by linearly blending the extracted foreground object image and a new background image with the output alpha matte.

## A.2 Local Thresholding

Considering that we have depth information, which can assist the supervised segmentation problem. And since the target is video but image, we need an automatic method. Ting et al. [19] proposes that tri-map can be automatically generated by assistance of depth-map as shown in Figure 46. The user first draws a rectangular on the depth image. The edge (contour) is then labeled, followed by morphological methods to automatically generate a tri-map.

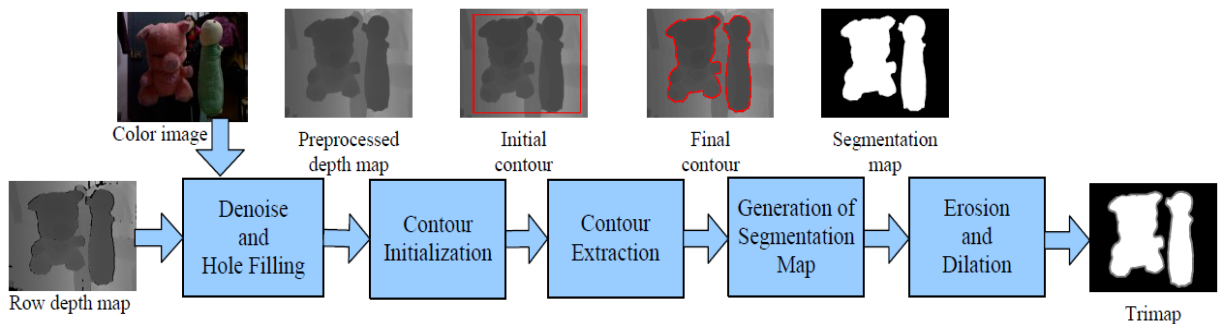


Figure 46: Flowchart of tri-map generation

However, there are challenges in our cases. The first problem is that when we try to use thresholding methods for the depth image, the threshold value automatically generated by Otsu's method usually fails in telling the foreground and background.

Since the depth values of the foreground object nears the toe get closer and closer to that of ground-floor, as shown in Figure 47, we cannot tell the contour simply by the depth image.

Our proposed row-by-row thresholding algorithm is described below. For the different region of depth image:

- Upper part: Check if there are larger depth difference, if no, no binarization is done.
- Lower part: Check if there are larger depth difference, if no, no binarization is done.

- The last row: Check if there are larger depth difference, to determine whether the foot is cropped.

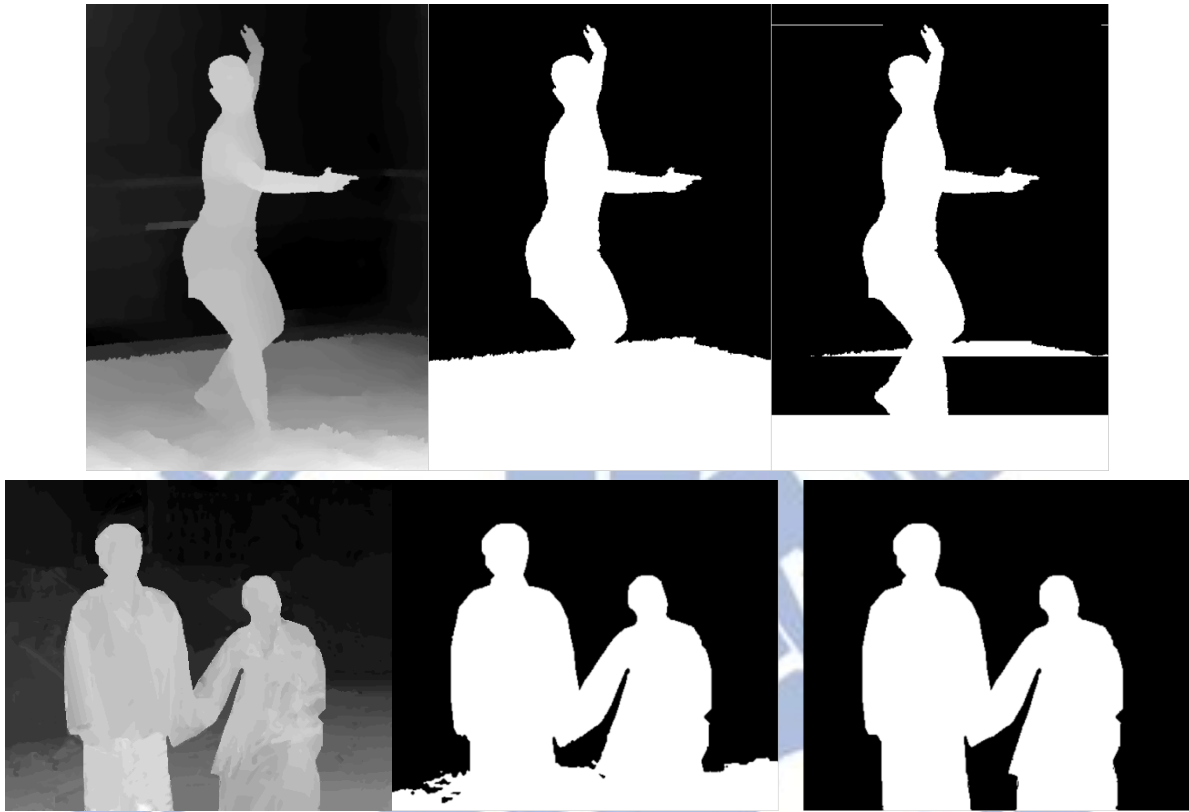


Figure 47: Two examples using proposed row-by-row thresholding: The left image is the selected original depth; the middle is the result using Otsu's threshold; the right is the result using our method.

### A.3 Automatic Tri-map Generation

Here the algorithm branches for two conditions: the object's lowest part is non-cropped and cropped. It depends on if large depth difference exists for object and background. For example, in Figure 47, we show two cases that the ballet dancer is not cropped, while the lovebird's couple are cropped.

For non-cropped mode (e.g., ballet dancer),

- Edge detection: Use sobel filter for grey channel defined as

$$H_x = [-1 \ 0 \ 1; -2 \ 0 \ 2; -1 \ 0 \ 1], \quad H_y = [1 \ 2 \ 1; 0 \ 0 \ 0; -1 \ -2 \ -1]; \quad (31)$$

Followed by thresholding. (Figure 48)

- Use binarized depth(Figure 47) as a mask. (Result in Figure 49: Left).
- Median filter (Figure 49: Middle).
- Morphological methods with hole-filling (Figure 49: Right).
- Erode and dilate the filled mask to produce the tri-map. (Figure 50).

For cropped mode (e.g., lovebird1), Depth information is reliable, we simply use eroded and dilated binarized depth for tri-map. An example is shown in Figure 51 and Figure 52.



Figure 48: Result of edge detection by sobel filter and thresholding.

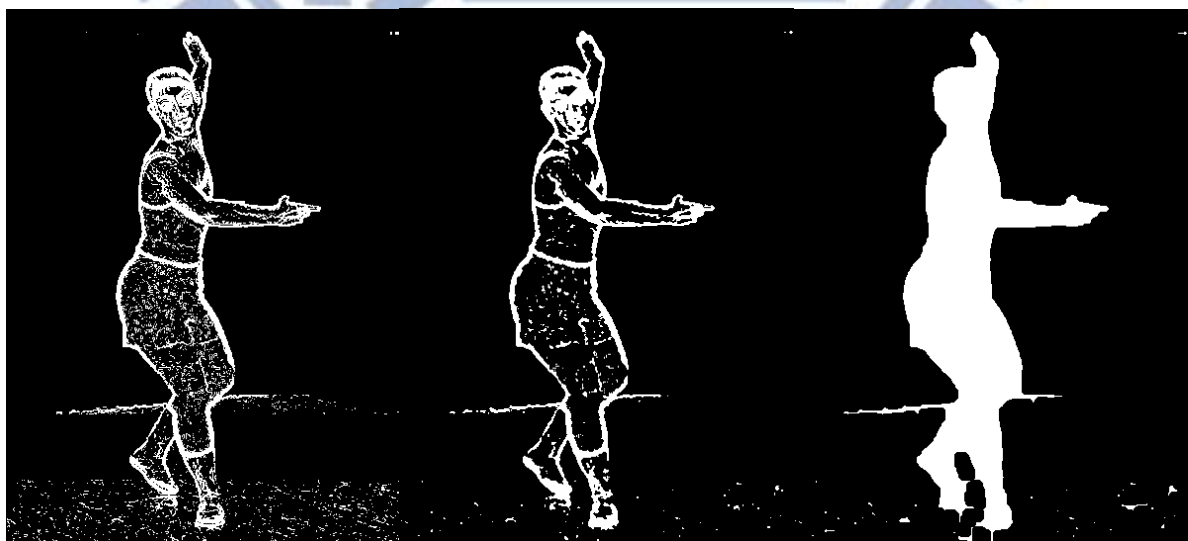


Figure 49: Temporary result of automatic tri-map generation.

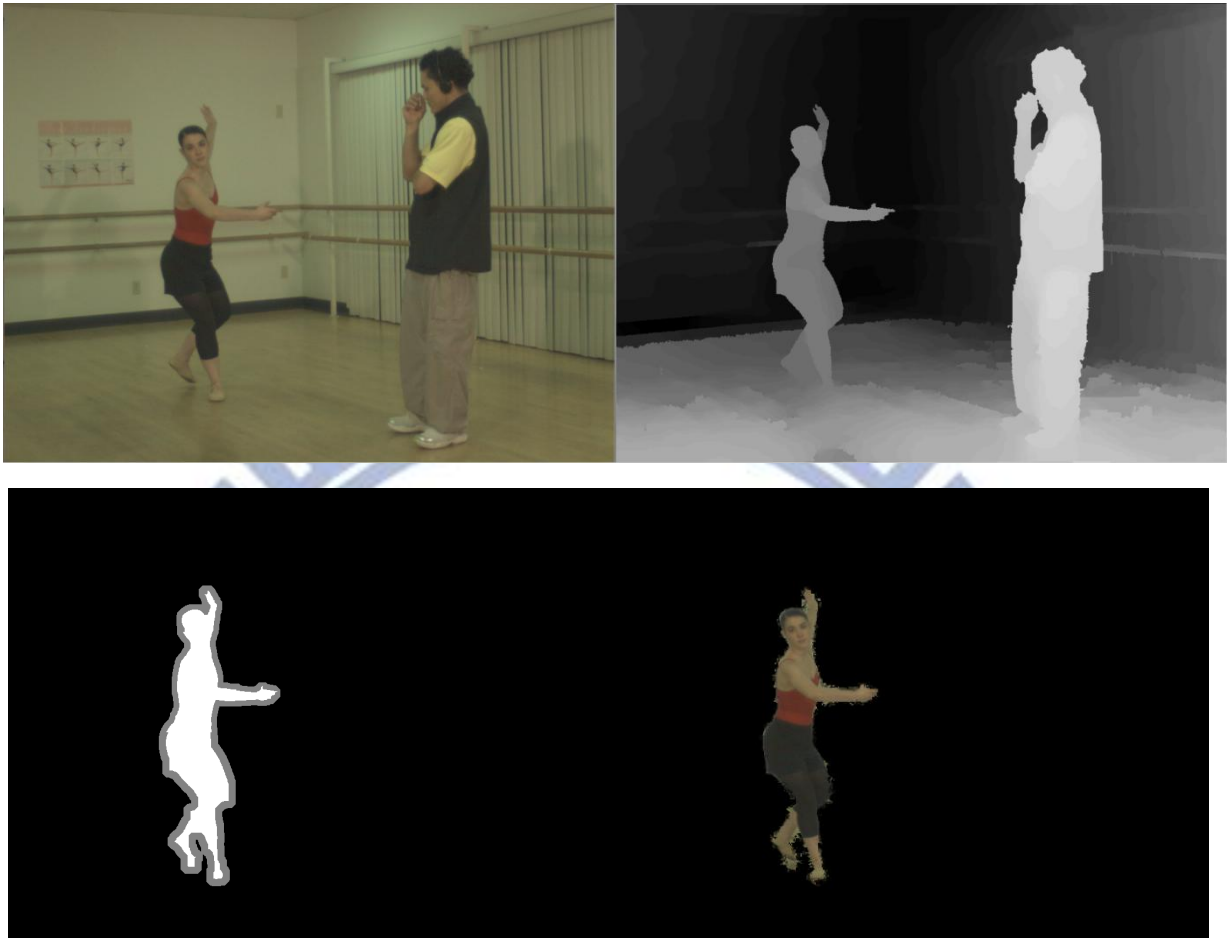


Figure 50: Result of ballet sequence(non-cropped mode)





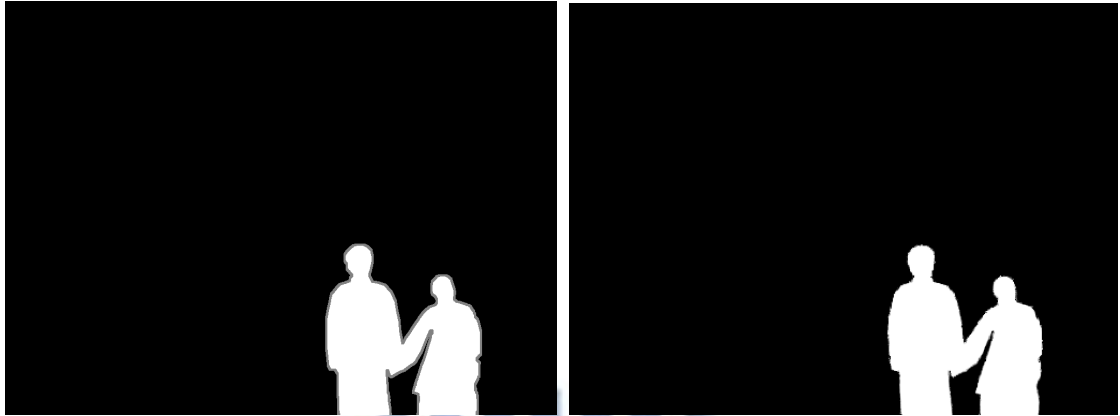


Figure 51: Temporary result of row-by-row thresholding of cropped mode (Lovebird1): The top left is the depth image; the top right is the binarized depth after row-by-row thresholding; the bottom left is the tri-map and the bottom right is the output alpha matte.

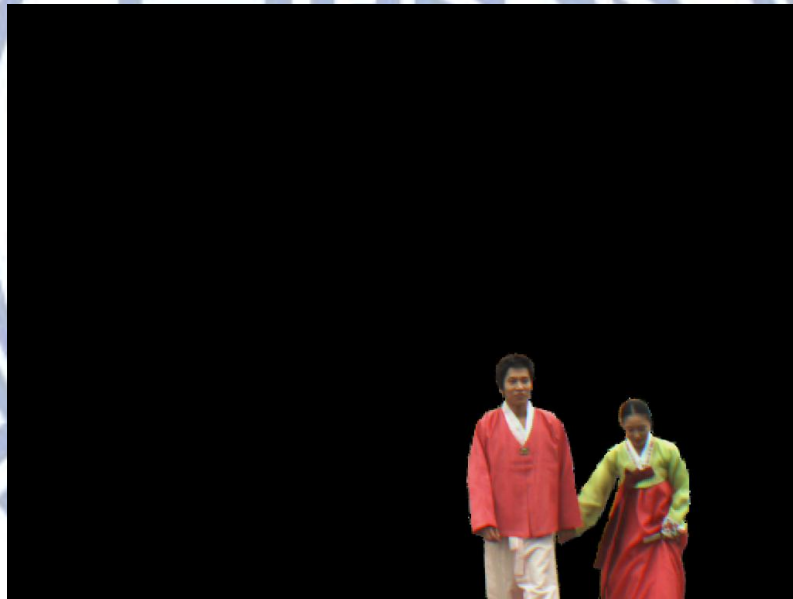


Figure 52: Linear blending result with black background.

However, in ballet sequence, we found that the dancer moves around drastically, which makes the rectangle chosen by the user throughout the whole video too large. So in our experiment, we only use Lovebird1 sequence for better demonstration.

## B. Color Grading

Here we try to solve the color temperature (or light condition) mismatch problem

discussed in chapter 4.1.2 Limitation of Conventional Composition. The behind model of the light condition can be regarded as a black box which is hard to control and predict. So usually we fix the digital photography by post-processing algorithms. This problem is known as color grading or color balancing. With today's tools, like Photoshop, this process still requires considerable human efforts, and is hard to adjust to a satisfied color as the dimension of color space is three.

We follow the idea of the algorithm proposed by Oskam et al. [16]. Despite the fact that the color temperature in the scene is unknown, we try to mimic the color tone of another scene to tune our foreground scene. In the following, we'll describe how it works.

### B.1 Vector Space Color Balancing

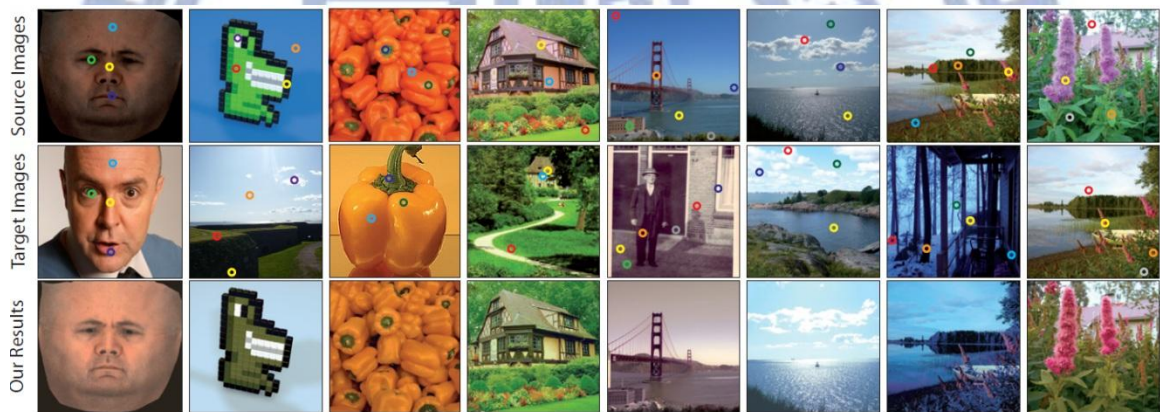


Figure 53: Color balancing[16].

Given a **sparse** set of color correspondences, the goal is to transform the color gamut of an image such that

- (i) colors that have references get as close as possible to that point in the color space.
- (ii) colors, for which no references are available, are transformed in a plausible way.

We interpret the given color correspondences as vectors in the color space, as shown in Figure 54. To achieve the two goals above, the assigned correspondence colors act as a role of support vectors, as in Figure 55, and other color tuning can be achieved by interpolation of these support vectors. Assume there are given  $n$  color correspondences. For a given pair of

color  $(c_i, d_i)$ , in the three-dimensional CIE Lab space, we define  $c_i$  as the support point. Vector  $v_i = \|d_i - c_i\|$ , and  $\phi_i$  is the basis function for each  $v_i$ . Those support vectors of  $v_i$  is annotated in  $w_i$ . The function is shown below.

$$v(e) = \frac{1}{\sum_{i=1}^n \phi_i(e)} \sum_{i=1}^n \phi_i(e) w_i \quad (32)$$

, where the basis function

$$s_i(c_j) = (1 + \|c_i - c_j\|)^{-\varepsilon} \quad (33)$$

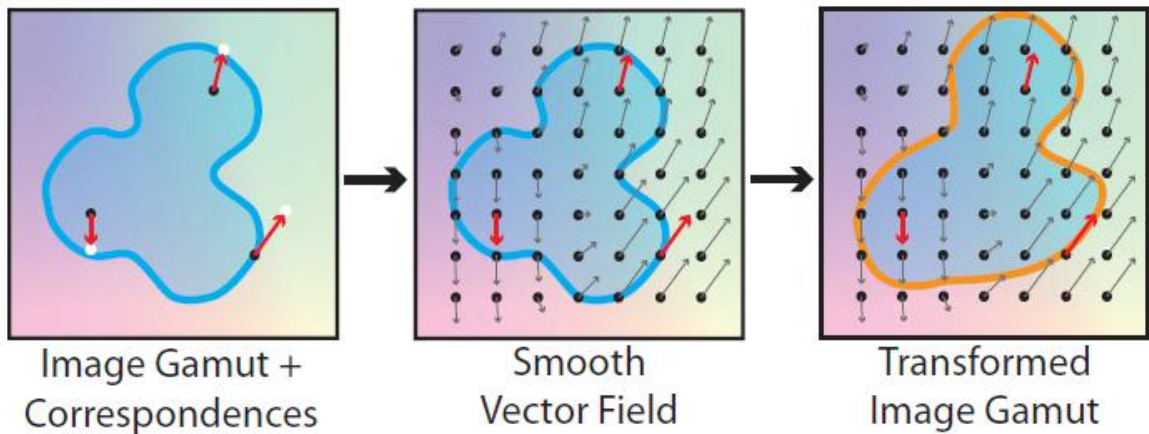


Figure 54: Color gamut: each color is viewed as a vector in 3-D space.

In their works they show that the best basis function is normalized Shepard function with  $\varepsilon = 3.7975$ . The result of implementing their algorithm is shown from Figure 56 to Figure 57. We can see that the result is plausible even in two different scenes and it let the color grading as easy and effortless for the user to adjust the color.

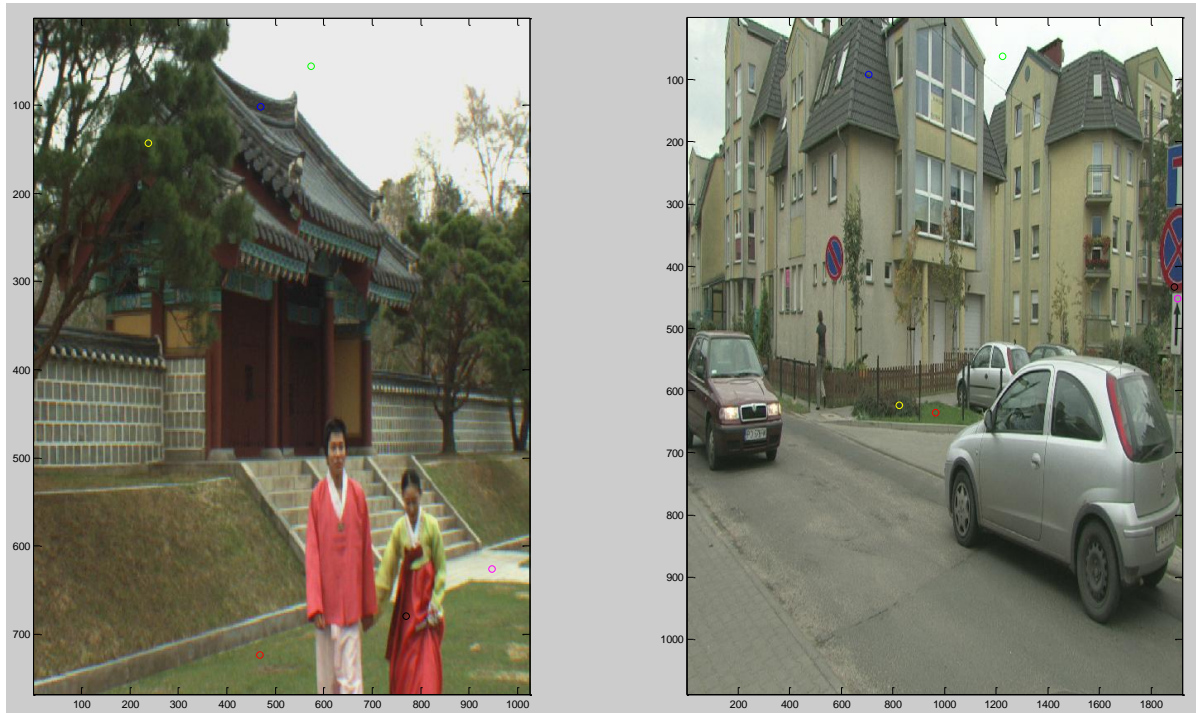


Figure 55: Marked color correspondence (marked as the circles with the same color by users).



Figure 56: The left shows the original image; the right shows the tuned result after the vector space color balancing.

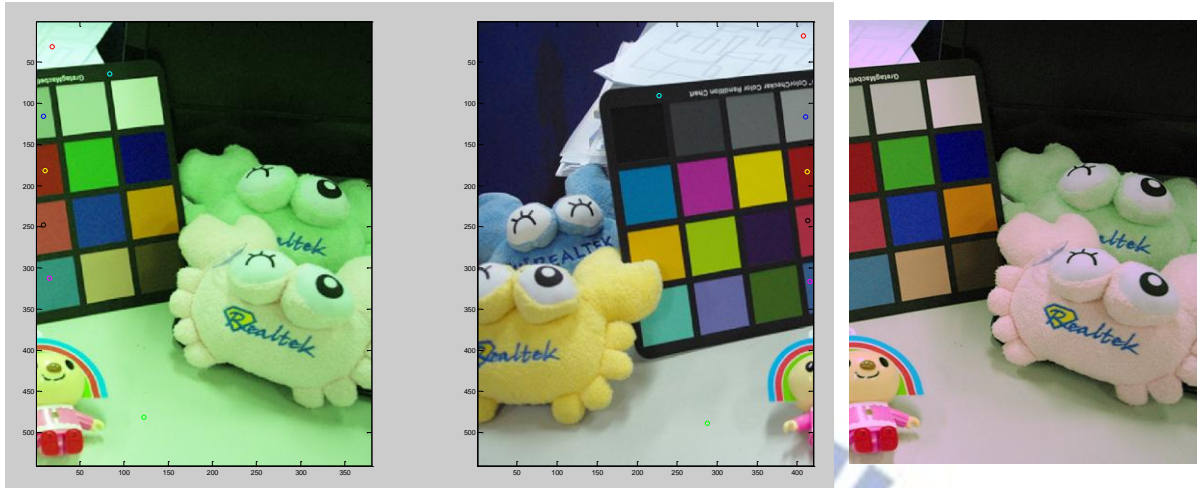
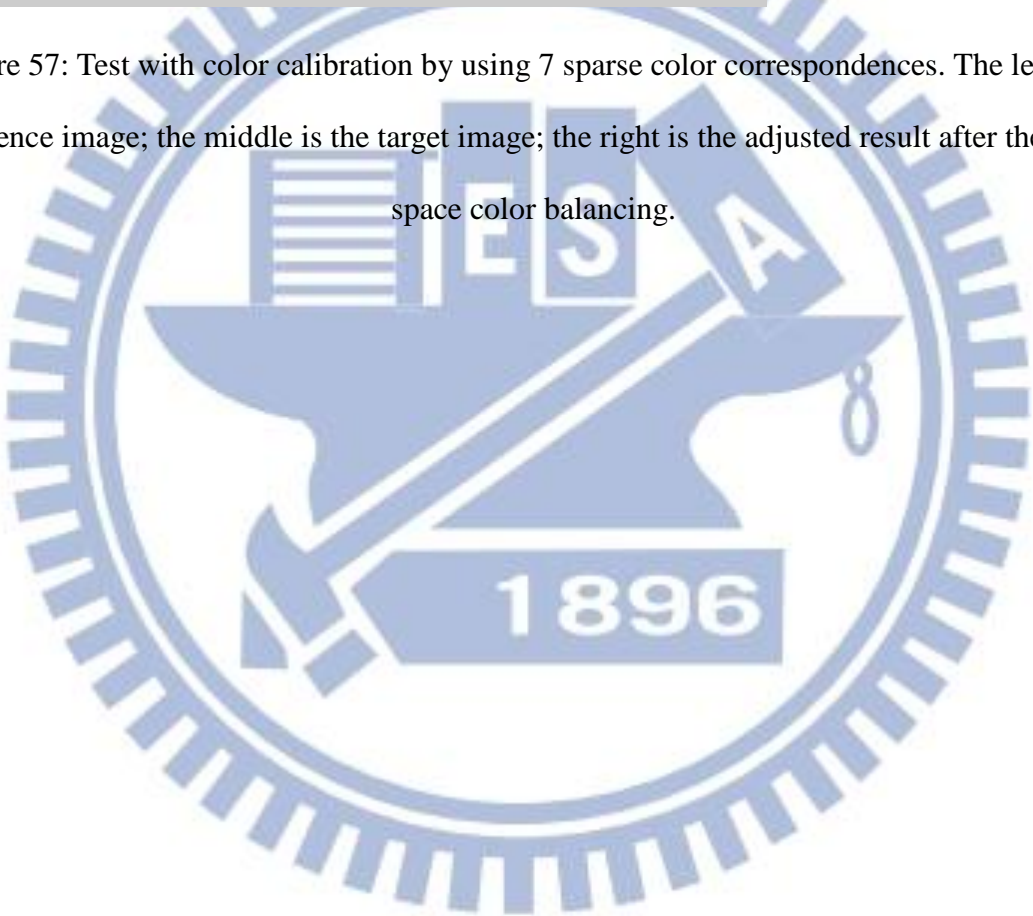


Figure 57: Test with color calibration by using 7 sparse color correspondences. The left is the reference image; the middle is the target image; the right is the adjusted result after the vector space color balancing.



## Bibliography

- [1] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process: Image Communication, Special Issue on 3DTV*, pp. 217-234, Feb. 2007.
- [2] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview Imaging and 3DTV," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 10–21, Nov. 2007.
- [3] M. Tanimoto, "Free Viewpoint Television (FTV)," *Digital Holography and Three-Dimensional Imaging (DH)*, Vancouver, Canada, June 18, 2007.
- [4] [http://www.youtube.com/watch?v=a21\\_WMiTAVE](http://www.youtube.com/watch?v=a21_WMiTAVE)
- [5] R. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2th edition, 2003.
- [6] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process: Image Communication, Special Issue on 3DTV*, pp. 217-234, Feb. 2007.
- [7] G. Chen, Y. Liu, and N. Max, "Real-time view synthesis from a sparse set of views," *Signal Process.: Image Commun.*, 22, (2), pp. 188–202, 2007.
- [8] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View Synthesis for Advanced 3D Video Systems," *EURASIP Journal on Image and Video Processing, Special Issue on 3D Image and Video Processing*, vol. 2008.
- [9] Video, "Report on Experimental Framework for 3D Video Coding," ISO/IEC JTC1/SC29/WG11 Doc. N11631, Guangzhou, China, October 2010.
- [10] M. Tanimoto, "FTV: Free-viewpoint Television," *Signal Processing: Image Communication*, Volume 27, Issue 6, July 2012.
- [11] <http://opencv.willowgarage.com/wiki/>
- [12] [http://en.wikipedia.org/wiki/Chroma\\_key](http://en.wikipedia.org/wiki/Chroma_key)

- [13] N. Diakopoulos, I. Essa, and R. Jain, "Content Based Image Synthesis," *Conference on Image and Video Retrieval (CIVR) 2004*, Dublin, Ireland, pp. 299-307, July 2004.
- [14] J. Lalonde, D. Hoeim, A. A. Efros, C. Rother, J. Winn, and A. Criminisi, "Photo Clip Art," *ACM Transactions on Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, August 2007
- [15] V. Kwatra, P. Mordohai, S. Kumar Penta, R. Narain, M. Carlson, M. Pollefeys, M. Lin, "Fluid in video: Augmenting real video with simulated fluids," *Computer Graphics Forum 27*, pp. 487-496, 2008.
- [16] T. Oskam, A. Hornung, R. W. Sumner, M. Gross, "Fast and Stable Color Balancing for Images and Augmented Reality," *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT) Second International Conference on*, pp. 49-56, 13-15, Oct 2012.
- [17] G. Klein and D.W. Murray, "Simulating low-cost cameras for augmented reality compositing," *IEEE Trans. Vis. Comput. Graph.*, 16(3):369-380, 2010.
- [18] M. Domanski, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, and K. Wegner, "Poznan Multiview Video Test Sequences and Camera Parameters," ISO/IEC JTC1/SC29/WG11 M17050, Xian, China, Oct. 2009.
- [19] T. Lu, S. Li, "Image matting with color and depth information," *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp.3787-3790, 11-15 Nov. 2012.
- [20] Y. Zheng, C. Kambhamettu, "Learning based digital matting," *Computer Vision, 2009 IEEE 12th International Conference on*, vol., no., pp.889-896, Sept. 29 2009-Oct. 2 2009.

## 自傳

張鈞凱，民國七十八年生於台北市。民國一〇〇年畢業於國立交通大學電機資訊學士班，同年進入國立交通大學電子研究所攻讀碩士學位，承蒙杭學鳴教授的指導，進入通訊電子與訊號處理實驗室(CommLab)，主要研究方向為虛擬視角合成、電腦視覺與擴增實境。論文題目為「基於虛擬視角的立體影片合成」。於民國一〇〇二年八月取得碩士學位。

