Université Catholique de Louvain

Ecole Polytechnique de Louvain
Département d'électricité

Virtual Image Reconstruction in a Multi-View Camera Network

Shruthi Narasimhe Gowda

Thesis submitted to Louvain School of Engineering
in partial fulfilment of the requirements for the degree of
Master [120]: ingénieur civil électricien, à finalité spécialisée

Promoter: Christophe De Vleeschouwer
Jury : Cédric Verleysen
P. Dutre

Louvain la Neuve, 2011-2012

Acknowledgements

I would like to express my sincere thanks to my promoter Prof. Christophe De Vleeschouwer for giving me an opportunity to work on this project. His guidance and motivation has helped me in the successful completion of this project.

I would like to thank Cédric Verleysen for his immense support and guidance all through the project. His experience in the area and knowledge proved highly beneficial and made this a great learning experience for me.

I wish to express my gratitude to the whole team of ICTEAM at UCL, whose feedbacks and ideas made a difference in the way the final work has shaped out.

I would take a moment to thank my family, for being my constant strong support system and encouraging me in all my endeavours.

Big heartfelt thanks to all my friends who have always been there for me, motivated me in the time of need and have made this journey worthwhile.

Last but not the least, I convey my thanks to all the readers who have taken time off to be interested in this project work.

Abstract

Virtual Viewpoint rendering, a popular research topic in computer vision, is synthesizing new views in a multi view camera network. One of the most attractive applications of virtual viewpoint synthesis is to enable an audience to view events from any arbitrary viewpoint. Most events happen occur in a large space, so object area is large and since viewers get to select any viewpoint according to their choice, the synthesized images should be a result of smooth transition between the images of the two cameras.

The proposed method is an attempt towards reconstruction of dynamic regions (players) of a basketball match in a virtual viewpoint using the images from two cameras that have a wide baseline (the line segment between the cameras). A novel synthesis method, which is a mixture of a weak image model based approach and the transfer based approach is used for view synthesis. The player is modelled in 3D with a very simple model, namely a plane; and the color information is used to synthesize the texture of these 3D models. The transfer based approach, which used projective geometry between neighbouring views, is used for estimating player positions. The efforts for camera calibration are reduced as cameras do not need to be strongly calibrated for projective geometry; hence, the proposed method can be easily applied to dynamic events in a large space.

First part of this report is an overview of the projective geometry (epipolar geometry) between views. The noticeable observation here is the high reliance on color information when finding the best correspondence between the regions in both views. This proves as a challenge because different cameras possess different color gamut and produce images of the same scene with color dissimilarity. This is solved effectively by performing camera color calibration which is discussed in the second part of this report. Finally, the last part deals with selecting best model (plane) for the player. We investigate the reconstruction of the player in accordance with our model using probability maps.

CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Virtual Viewpoint Rendering, a popular topic in computer vision area, is a method of synthesizing new views in a multi-view system. Virtual viewpoint synthesis in a multi-view environment is one of the most attractive video applications. Recently there has been great deal of interest in making a system that enables an audience to view sports event from any arbitrary viewpoint. The research aims at generating a virtual viewpoint image from the multiple view videos (e.g. in a sport). The chosen free-viewpoint may not only be selected from the available multi-view camera views, but also any viewpoint between these cameras. It will be possible to watch a soccer game or a concert from the user preferred viewpoint, where the viewpoint can be changed at any time.

The rapid development of networks and computers will soon make it possible to provide a form of real-time video that allows the viewer to select any desired view of the action, but just the video processing technology is not enough. The acquisition and rendering of visually realistic images of real objects and scenes has been a long standing problem in both computer graphics and computer vision. Where computer graphics deals with the complex modelling of objects and simulation of light interaction in a virtual scene to generate realistic images, computer vision offers the opportunity to capture and render such models directly from the real-world with the visual realism of conventional video images.

Several methods are used to visualise an object or a scene from any desired viewpoint and viewing angle. Virtualized reality, a pioneering project in this field, has achieved such virtual movements for dynamic scenes using computer vision technology (T. Kanade 1997). Three-dimensional models of objects in the target scene are reconstructed from multiple view images, but the target area in these methods is some localized area with dense camera arrangement. This is the **Model based rendering** which synthesizes images using computer graphics.

More recently, there has been research to realize the virtual view synthesis in an actual sporting event held in a stadium. The entire scene including the players, field, and stadium is a reconstruction target, i.e., the object area is very large. Moreover, the image at virtual viewpoint should also be synthesized using captured scenes rather than computer-generated model. So, intermediate viewpoint is generated from the real images obtained by the camera views. This is the **Image based rendering** technique which synthesizes the new view through view interpolation. Also, because the

viewers can get to choose the viewpoints to their choice in real time, the intermediate images should be a result of smooth transition between the images of the two cameras.

These criterions give rise to a lot of **challenges**. The difficulty arises when the baseline (the line segment between two cameras) is too big. The solution would be to practically place a large number of cameras and generating intermediate images between them, but this decreases the cost efficiency. The Model based rendering methods which generate an accurate 3D model of the scene require a lot of images (or cameras) for a large area (like a stadium). The basic idea of most depth image based rendering (DIBR) rendering methods is to perform 3D warping to the virtual viewpoint using texture and depth information of the reference cameras(A. Smolic 2008). But lesser number of cameras implies a high baseline and hence acquiring the depth information becomes almost impossible. So, the novel approach is to perform rendering without the depth map/distortion map information.

The goal is to reduce the number of cameras required, but yet synthesize intermediate images at viewpoints of choice from the real captured images from nearby cameras and also provide a smooth transition between the viewpoints. This has an evident trade with respect to the quality of reconstruction in the new view.

1.2 Related Work

In the field of computer vision, the techniques for synthesizing virtual view images from a number of real camera images have been studied since the 1990s. These techniques can be categorized into three groups, ***model based approach***, ***transfer based approach***, and approach using the ***plenoptic function***.

In the *model based approach*, a 3D surface model is constructed for a scene and texture maps are extracted from the images(T. Kanade 1997). The quality of the virtual view image then depends on accuracy of the 3D model. As a large number of video cameras or range scanners are typically used to construct an accurate model, this approach requires large amounts of calculation and information. Furthermore, camera calibration (Tsai 1987) is usually required to relate 2-D coordinates in images to their corresponding 3-D coordinates in object space. The camera calibration is computationally complex and also proves very expensive on a large area. Once the 3D model has been constructed, however, it is possible to flexibly change the viewpoint to present another view.

Transfer based approach on the other hand synthesises novel views directly from the original images rather than through explicit reconstruction of scene geometry. The arbitrary viewpoint image is synthesized without an explicit 3D model; instead, is synthesized by transferring of correspondences between real view images. View Interpolation(Williams 1993) or View morphing (Dyer 1996) can be used. Shashua and Avidan (Shashua 1998) have employed trifocal tensor for image transfer.

Depth image based rendering method requires depth information of the given cameras. Zitnick (C. L. Zitnick 2004) provide a free-viewpoint rendering algorithm which is based on a layered depth map presentation. These results are extended in the work of Smolic and Muller(A. Smolic 2008),

where three layers of depth are identified and ranked according to their reliability. The warping results are then obtained for each layer and merged. These methods generate the high-quality virtual view by assuming dense camera arrangement. The drawbacks of such research are that it is necessary to perform strong calibration, which estimates the projective relation between world coordinate system and camera coordinate system. Therefore, it becomes more difficult as the target space becomes large (large baseline).

As regards approach using the *plenoptic function*, which describes all the radiant energy that is perceived by an observer at any point in space and time, it is possible to create novel views from a collection of sample images. This allows a user to arbitrarily pan and tilt a virtual camera and interactively explore his/her environment. In its most general forms, the plenoptic function is a seven-dimensional function. Due to its high dimensional nature, data reduction or compression of the plenoptic function is essential. The light field of Levoy and Hanrahan (Hanrahan 1996) have simplified the function with four dimensions. However, it is inadequate for large-scale events because it is impossible to describe all the radiant energy.

In order to reduce difficulty in strong calibration of cameras or using large number of cameras in close proximity, the technique (S. Y. Saito 2004) of generating a virtual view is proposed using the 3D model estimated from the weak calibration, which estimates only the epipolar geometry between multiple cameras.

Naho Inamoto and Hideo Saito (N. I. Saito 2007) have experimented in generating a virtual viewpoint image, by applying IBR based on this Transfer Based Approach to the whole soccer scene. In its method, after classifying a soccer scene into dynamic regions and static regions, the appropriate projective transformation is applied respectively. This is easily applicable also to the sport scene in large-scale space like soccer, since it is only required to perform weak calibration, which estimates a projective geometry between multiple views. This method is based on dividing a scene according to the geometric character, generating a virtual view image for every domain, and merging them. By taking such scene dividing strategy, it has succeeded in generating a virtual view image of the whole soccer scene.

1.3 Proposed Method

The objective of this work is to reconstruct dynamic regions of a basketball match (players) in a virtual viewpoint in between two cameras having a wide baseline (the line segment between the cameras). Since the movements of each player are complex, it's almost impossible to reconstruct an accurate 3D model of the scene. Furthermore, the full camera calibration that is necessary to compute 3D positions is very difficult in a real stadium. Here we use a weak model based approach and also use projective geometry of the transfer based approach.

Images of a basketball match by two different cameras are considered. The simplest model is used for the players, namely planes. Epipolar geometry is used to find correspondences between players in these two images. This information should be used to reconstruct the players in a virtual viewpoint in between the cameras.

Firstly, the projective geometry between multiple views has to be understood to help transfer correspondences between images. Many different correspondences exist between two views. Here, we mainly focus on two important mappings: Fundamental matrix and Homography. The players are detected and labelled and the labelled regions of the same player in the neighbouring views are corresponded through homographic transformations of the appropriate player planes. To obtain perfect correspondence between the players, the planes defined for each player to estimate the homography has to be ideal. So, the goal has to be to select the best possible plane for each player, and this can be selected based on comparisons between the correspondences from both views for different planes. This comparison is based on color only, and the plane that gives the best correspondence is chosen as a rough 3D model of the player.

This necessitates color uniformity between the images of the two cameras. This cannot be guaranteed because different cameras have different color gamuts and can produce images of the same scene with some color discrepancies. Hence, color correcting techniques should be implemented before proceeding to the estimation of player probability map. Three different methods of color correction are discussed in this work, two of which are applied on the images after being received by the cameras, while the third method is directly implemented on the camera itself and hence is called the camera color calibration.

Once the color correction is achieved, the best plane is chosen for each player. Player correspondences are determined for this model, and then the focus is shifted to reconstruct this player on a chosen viewpoint in between the cameras. For a good model (the best plane selected), the correspondences should yield a perfect representation of the player. This is verified by creating probability maps. To decide if a particular point in the virtual view belongs to the player or the background in 3D reality, probability map/decision is used, which aids in a good reconstruction of the player.

The different topics in this report are organised as follows. The following chapter, i.e. the second chapter gives a brief overview of Epipolar geometry. The projective geometry is implemented on the given images to get a better understanding and also to establish the importance of color uniformity between images of the two views. As mentioned earlier, before proceeding to the player reconstruction, color correction needs to be implemented. The different color correction methods are discussed in the third chapter. After attaining color uniformity between images of the two cameras, projective geometry is revisited to continue with finding correspondences between players. The player reconstruction is discussed in the fourth chapter.

Chapter 2

Epipolar geometry

2.1 Introduction

Virtual viewpoint rendering, as mentioned in the previous chapter, is synthesizing a new intermediate view in a multi-view system. The transfer based approach used to achieve this makes use of projective geometry between the views. Epipolar geometry is the most important tool used in this approach.

Epipolar geometry is the intrinsic projective geometry between two views (Zisserman 2000). The main goal in virtual viewpoint reconstruction using transfer based approach is to obtain correspondences between the two given view points (the two camera images).

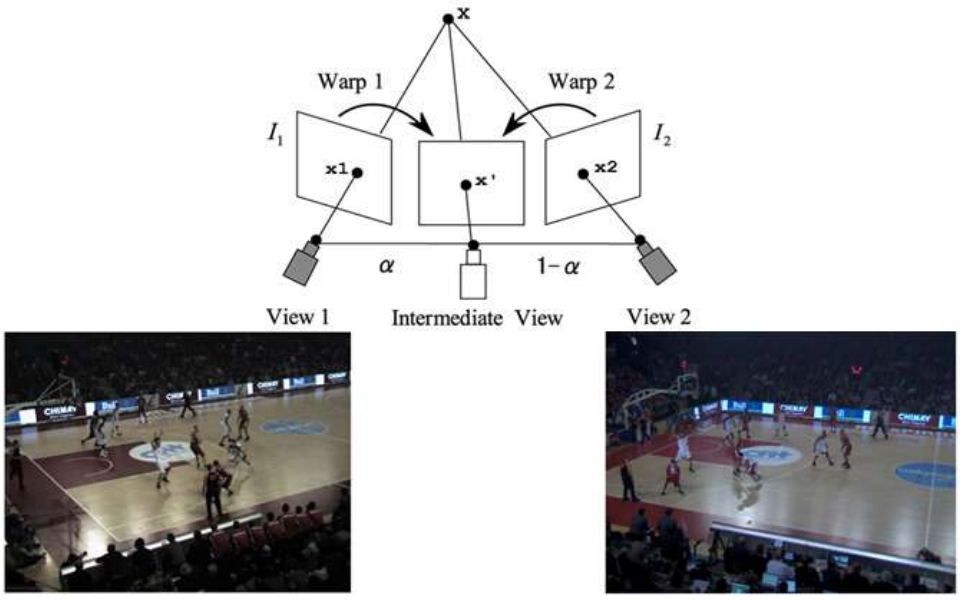


Figure 1: Virtual viewpoint synthesis between two cameras

Figure 1 shows two cameras capturing a same scene, in this case, a basketball match. So there are two images of the same scene but from different viewpoints. As explained before, the goal is to reconstruct the scene in any virtual viewpoint in between the two cameras with a very smooth transition between the views. This is possible by finding dense correspondence between the two obtained images.

The Epipolar Geometry helps in achieving the correspondence between two views and has the additional benefit of being totally independent of the scene being captured. The cameras work on the basic projective ray geometry and the projections between the cameras help provide these needed correspondences. It is the general projective geometry between any two views and depends only on certain parameters of the camera and their relative positions.

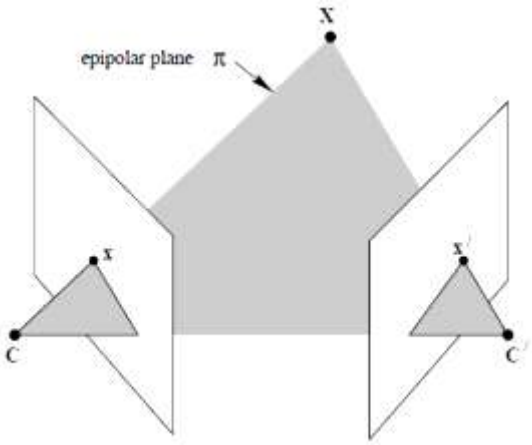


Figure 2: Epipolar geometry between two views

Consider Figure 2, C and C' are the two cameras (the camera centers. i.e. the center of the lens in the physical sense) and X is a point in 3D space. The 3D point X is imaged by the two cameras as x and x' respectively. The idea is to find some constraint on x'(second view) when x (first view) is known.

Note: All through the report, the capital letters (X) denotes the 3D point (on the scene) and the small letters (x and x') denote the 2D points (on the respective image planes).

Many correspondences exist in between the two views. But this work mainly focuses on two correspondences. These are,

- Point to Line correspondence: For any point on the first view, there is a corresponding line in the other.
- Point to Point correspondence: For any point on the first view, there is a corresponding point in the other view.

Both these correspondences are discussed in this chapter. Before following on with the correspondences, a brief knowledge about the working of a camera is required. The projective geometry can be more easily understood by knowing the camera and its parameters, which are explained in the next segment.

2.2 Camera

To understand projective geometry between multiple cameras, learning about the working of a camera is of paramount importance. The process of image formation, namely the formation of a two-dimensional representation of a three-dimensional world, helps deduce about the 3D structure of what appears in the images. A simple camera model and its parameters (which establish a definite relationship between the camera coordinates and the world coordinates) are discussed in this section.

2.2.1 Projective Camera Model

A very simple model of the camera is considered for this research work. There are different kinds of cameras but we restrict it to a basic one which can be defined as a mapping between the 3D world and a 2D image. A general projective camera works on ray geometry. Consider a ray from 3D world point through a fixed point in space, the centre of projection. This ray will intersect a specific plane in space chosen as the image plane. The intersection of the ray with the image plane represents the image of the point (Faugeras 1993).

This model is in accordance with a simple model of a camera, in which a ray of light from a point in the world passes through the lens of a camera and falls on a film or digital device, producing an image of the point. Ignoring such effects as focus and lens thickness, a reasonable approximation is that all the rays pass through a single point, the center of the lens (Zisserman 2000). The center of projection is the camera center (Figure 3). The line from the camera center perpendicular to the image plane is called the principal axis or principal ray of the camera and the point where the principal axis meets the image plane is called the principal point.

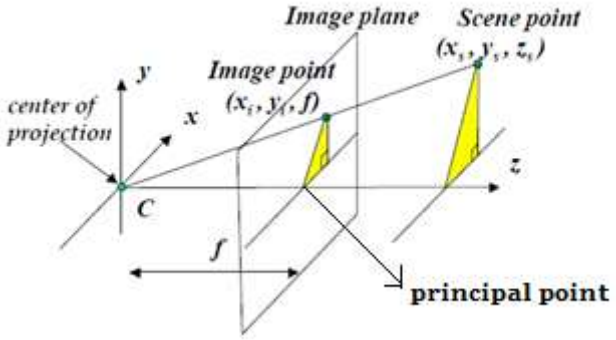


Figure 3: The camera center, its image plane and the 3D scene

Homogeneity

A point in Euclidean 2-space is represented by an ordered pair of real numbers, (x, y). An extra coordinate can be added to this pair, giving a triple $(x, y, 1)$, that can be declared to represent the same point. This seems harmless enough, since we can go back and forward from one representation of the point to the other, simply by adding or removing the last coordinate. These are called the homogeneous coordinates of the point.

Euclidean space (or Cartesian space) describes the 2D/3D geometry well, but they are not sufficient to handle the projective space. Imagine a point at infinity, in non-homogeneous coordinates, there is no numerical representation for a point at infinity; (∞, ∞) is meaningless. From the point of view of projective geometry, points at infinity are not any different from any other points. In homogeneous coordinates, such a point has its third component equal to zero i.e. it has the form $(x, y, 0)$ (Zisserman 2000).

The equations for different transforms in projective geometry are non-linear when expressed in non-homogeneous coordinates, but are linear in homogeneous coordinates. It provides one of the

main motivations for the use of homogeneous coordinates, since linear systems are symbolically and numerically easier to handle than non-linear ones.

Homogeneous coordinates have a range of applications, including computer graphics and 3D computer vision, where they allow affine transformations and, in general, projective transformations to be easily represented by a matrix.

$$\begin{pmatrix} kx \\ ky \\ k \end{pmatrix} \quad = \quad k \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

k is a non zero constant;

Hence, all the coordinates used in projective geometry in this work are represented as homogenous coordinates.

The camera model can be further explained based on its intrinsic and extrinsic parameters.

2.2.2 Intrinsic Camera Parameters

The set of intrinsic camera parameters govern how 3D points in the scene of reference are mapped to 2D coordinates in the images plane.

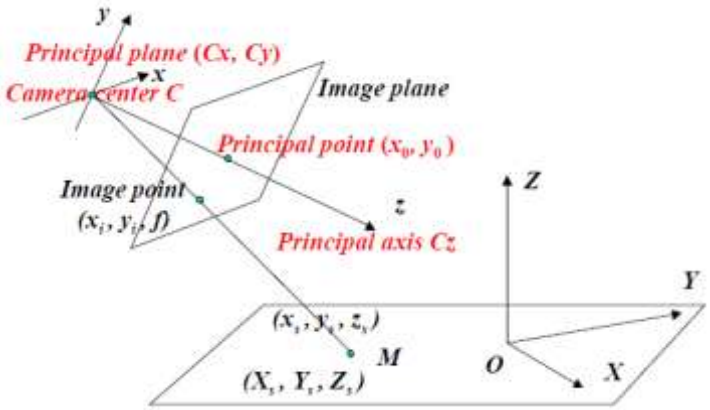


Figure 4: Camera coordinates and world coordinates

When the world point (x_s, y_s, z_s) and image point (x_i, y_i, f) are represented as homogenous vectors, a linear transformation can be generated between the two. The focal distance is denoted as f.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

$$x_i = \frac{u}{w}\,; \quad y_i = \frac{v}{w}\,;$$

This equation assumes that the origin of the coordinates in the image plane is at the principal point. In practice it may not be so.

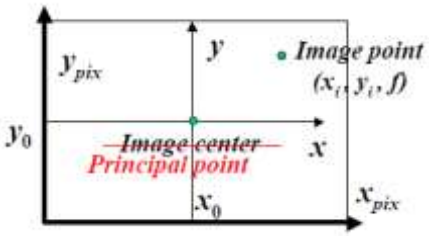


Figure 5 : Image center and the principal point offset

There might be an offset for the principal point.

The equation now becomes,

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

(x_0, y_0) are the coordinates of the principal point (Figure 5).

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K[I_3|O_3] \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

K: The Camera Calibration matrix

$'K'$, the calibration matrix is a 3×3 upper triangle matrix and has five degrees of freedom. It is an essential parameter that compensates for the principal point offset in the image plane.

2.2.3 Extrinsic Camera Parameters

The extrinsic camera parameters relate the camera's position with respect to the world coordinate reference frame. Hence it is simply the translation and rotational transformation from the world origin to the camera's position. These are often represented as a 3D translation vector T and a 3x3 rotation matrix R.

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ O_3^T & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

T: Translation vector; R: Rotation matrix

Combining the two parameters of the camera model, we obtain a linear transformation from World coordinates to image coordinates.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K[I_3|O_3] \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = K[I_3|O_3] \begin{bmatrix} R & T \\ O_3^T & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

$$\boldsymbol{x = PX}$$

$X = [X_s, Y_s, Z_s, 1]^T$: World coordinates of a scene point

$x = [{}^{u}/_{w}, {}^{v}/_{w}, 1]^T$: Projection of the scene point on the image plane

$P = K[I_3|O_3]\begin{bmatrix} R & T \\ O_3^T & 1 \end{bmatrix}$: is the $3x4$ matrix that maps scene to image and is called projection matrix of the camera or the camera.

The mapping of the camera from 3D scene to the 2D image plane has been explained briefly. Some relevant topics like lens distortion have not been explained as it is out of scope for this work. The next goal is to use this information to project from one view to another to establish a relation between the images of the two cameras.

2.3 Point to Line Correspondence: Fundamental Matrix

The first correspondence to be discussed is the point to line correspondence which is encapsulated by the fundamental matrix (Zisserman 2000).
Figure 6 shows the same model as before, with two cameras, 3D scene and the two image planes. The mapping function (Fundamental matrix) which provides this point-to-line correspondence is explained.

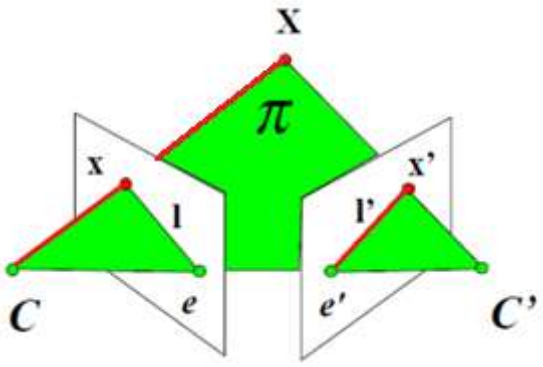


Figure 6: Epipolar geometry between two views

Property: The baseline (the line segment between two camera centres), the 3D point (X) and its image on the two image planes (x and x') lie on the same plane (coplanar).

This is called the epipolar plane and in Figure 6, this is defined as the plane π. Consider we know x, the ray defined by this point along with the baseline form the plane π. But from property 1 (coplanar), the ray corresponding to any point x' in second view should also lie on the same plane π. So this point can be any point on the intersection between the plane π with the second image plane. This intersection results in a line $l'$and is shown in Figure 6. So, the correspondence of point x in first view is now restricted to a line in the second view (from the whole image plane!).

This line l' is called the epipolar line of point x. So the epipolar line can be defined as the intersection of epipolar plane with the image plane of the camera. The points of intersection of the baseline with the image planes are the epipoles (e and e') and lie on the epipolar line.

The epipolar line in the second view is actually the image (or projection) in the second view of the ray back projected from X (the ray from point X through the camera centre C). The red line in

the Figure 6 represents this. This line l' extends from x' till the epipole e'. So the image of camera centre C in second view is actually the epipole e'!

The existence of some mapping between a point (x) in the first view to the epipolar line (l') in second view has been established.

$$\boldsymbol{x \rightarrow l'}$$

This mapping is provided by the fundamental matrix,

$$\boldsymbol{l' = Fx}$$

F is the Fundamental matrix.

The point x' in the second view lies on the epipolar line l',

So ${x'}^T l' = 0$,

but $l' = Fx$

$\Rightarrow {x'}^T Fx = 0$

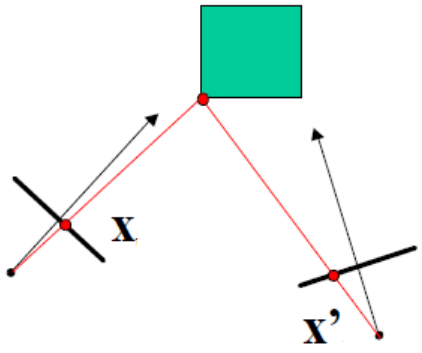


Figure 7 : Point to point correspondence between two views

So, for any pair of corresponding points $\boldsymbol{x \rightarrow x'}$ in two images (Figure 7), the fundamental matrix satisfies the condition,

$$\boldsymbol{{x'}^T Fx = 0}$$

2.3.1 Fundamental matrix: Estimation

The following formulation is from (ZHANG 1996). From topic 2.2.3, the relation between the 3D scene and the 2D image plane is given by,

$$\boldsymbol{x = PX}$$

and $\boldsymbol{x' = P'X}$

$\boldsymbol{P}$: Projection matrix of camera $\boldsymbol{C}$ (first view).

$\boldsymbol{P'}$: Projection matrix of camera $\boldsymbol{C'}$ (second view).

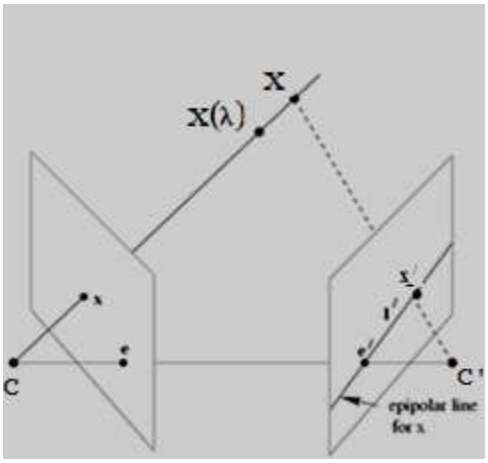


Figure 8 : Point to line correspondence (1)

The ray back projected from x by P can be solved by the equation,

$$X = P^{-1}x$$

P^{-1} is the pseudo inverse of P such that $PP^{-1} = I$ (Identity matrix)

The one parameter solution is given by,

$$X(\lambda) = P^{-1}x + \lambda C$$

C, the camera center (Camera1), is a null vector such that $PC = 0$.

The ray is parameterised by λ. There are two points on this ray (red in Figure 9),

1. $X = P^{-1}x$ at $\lambda = 0$;
2. C, the camera centre at $\lambda = \infty$

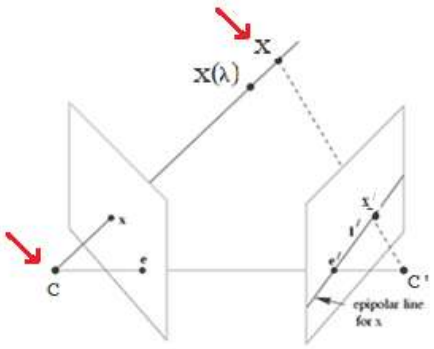


Figure 9 : Point to line correspondence (2)

The images of these two points in the second view are (green in Figure 10),

1. X is imaged at x' in the second view: $x' = P'X$

Substituting $X = P^{-1}x$

$$\Rightarrow x' = P'P^{-1}x$$

2. C is imaged as the epipole e' in the second view,

$$\Rightarrow e' = P'C$$

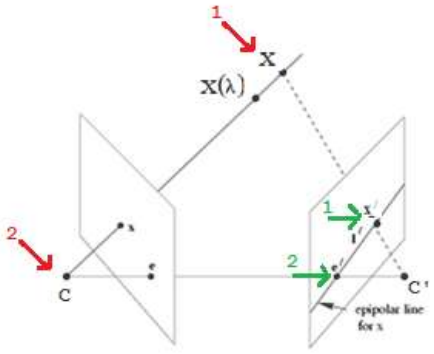


Figure 10 : Point to line correspondence (3)

The epipolar line l' passes through these two points, x' and e'.

$$l' = (e') \times (x')$$

$$l'=[e']_x \, P'P^{-1}x \quad ; \; [e']_x = \begin{bmatrix} 0 & -e'(3) & e'(2) \\ e'(3) & 0 & -e'(1) \\ -e'(2) & e'(1) & 0 \end{bmatrix}$$

$$l'=[P'C]_x \, P'P^{-1}x$$

$$l' = Fx$$

$$\boldsymbol{F = [P'C]_x \, P'P^{-1}}$$

F from the above equation is the fundamental matrix of the camera pair (P, P'). F is a unique 3x3 homogenous matrix of rank 2 and 7 degrees of freedom.

- For **canonical cameras** with world origin at the first camera(Zisserman 2000),

$$P = [I|0] \; ; \quad P' = [M|m]$$

$$\boldsymbol{F = [e']_x \, M}$$

$$e' = m$$

$e = M^{-1}m$ (are the epipoles)

- For **stereo-rig cameras**(Zisserman 2000),

$$P = K[I|0] \; ; \quad P' = K'[R|t]$$

$$\boldsymbol{F = [e']_x \, K'RK^{-1}}$$

$$e' = K't$$

$$e = KR^T t$$

For the given projection matrices for the two cameras, F is calculated. This is illustrated in Figure 11. For a point selected in the first image (basketball field center), the corresponding line in the second image is viewed.

Figure 11: Point to line correspondence between the two images

For a point in one image, the corresponding point in the second image is now restricted to just a line. To further find the exact corresponding point on this line, homography is introduced.

2.4 Point to Point correspondence: Homography

Two correspondences between any two views (in a multi view system) were considered for this work. The first relation was covered in the previous segment, where, the fundamental matrix defines correspondence between a point in one view to a line in another,

$$\boldsymbol{x} \rightarrow \boldsymbol{l}'$$

and the corresponding point can be anywhere on this line. In this segment, the point-to-point correspondence is discussed.

$$\boldsymbol{x} \rightarrow \boldsymbol{x}'$$

Images of points on a plane in one view are related to corresponding image points in another view by a planar homography using a homogeneous representation. Homography plays an important role in the geometry of multiple views (Zisserman 2000).

Importance of Depth

Figure 12 is the basic representation of the two camera centres, their image planes and the 3D points that we have already seen before.

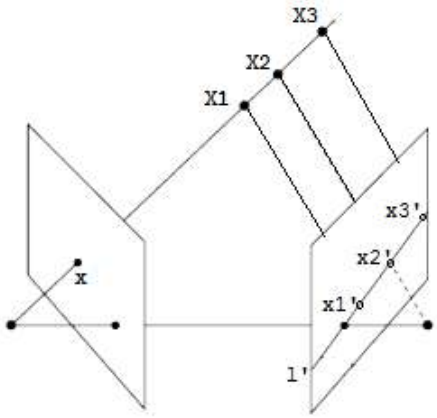


Figure 12: Point to point correspondence between two views

The point x in first camera can be projected to any point on the line l' in the second image plane. Consider the ray extending from x meeting the 3D points $X1, X2$ and $X3$, now the point x can correspond to $x_1{}', x_2{}', x_3{}'$ when this point is projected back on to the second camera. To know the point correspondence of x on the line l', we need to know the depth information in 3D.

The whole idea of homography is based on the world plane introduced at the particular depth needed. Imagine, if a world plane is introduced at $X1$ in the 3D scene, the ray from a point on the world plane ($X1$) projects to x in the first image plane and the point $x_1{}'$ in the second. So by forcing a plane the constraint can be pushed from anywhere on the epipolar line to a single point on it.

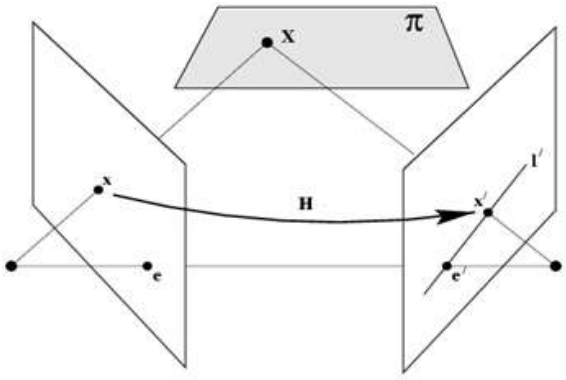


Figure 13: Homography between two views induced by the plane π

Figure 13 shows the homography induced by a plane. A world plane π has been defined. The ray extended from point x in the first view meets the world plane at a point x_π and this point is projected on to the second view, where it meets the second image plane at x'. This establishes the point-to-point correspondence between two views. This mapping between x and x' is the homography induce by the plane π.

$$\boldsymbol{x \rightarrow x'}$$

This mapping can be visualised algebraically. There is an independent relation between the world plane and the two image planes. There is a relation between the world plane and the first image plane,

$$x = H_{1\pi}X$$

Similarly, exists a relation between the world plane the second image plane,

$$x' = H_{2\pi}X$$

Combining both the relations, $x' = H_{2\pi}H_{1\pi}^{-1}X$

$$\boldsymbol{x' = Hx}$$

H: homography between the two image planes.

This is a pure projective relation since it only depends on the intersection of planes with lines. The homography transfers points from one view to the other as if they were images of points on the plane.

An inherent advantage of homography is that the relationship is independent of the scene structure. The homography depends on the intrinsic and extrinsic parameters of the cameras used for the two views and the parameters of the 3D plane. It does not depend on what the cameras are looking at and the homography holds regardless of what is seen in the images. Another important property of homography is that it is unique to every plane.

- For **canonical cameras** with world origin at the first camera,

$$P = [I|0] \;\; ; \;\; P' = [M|m]$$

Consider a plane $\pi = [v^T, 1]^T$ for a vector v in the scene. If the plane is defined as $ax + by + cz + d = 0$; (d is taken as 1 here), then vector $v = [a, b, c]$.

The ray corresponding to x meets this plane at X, which projects to x' in the other image.
For the first view $x = PX = [I|0]X$ and so any point on the ray $X = (x^T, \rho)^T$ projects to x, where ρ parameterizes the point on the ray. Since X lies on the plane π,

$$\pi^T X = 0$$

Solving for ρ, provides $X = (x^T, -v^T x)^T$

The homography is given by(Zisserman 2000),

$$x' = P'x = [M|m]X$$

$$x' = \left(\mathrm{M} - \mathrm{mv}^{\mathrm{T}}\right)x$$

But, we know that $x' = Hx$

$$\mathbf{H} = \mathbf{M} - \mathbf{mv}^{\mathbf{T}}$$

2.4.1 Homography: Estimation

A plane in 3D can be specified by three points and homography can be estimated by finding the corresponding image elements that specify the plane (Dubrofsky 2007). Explicit method uses three point correspondences to estimate homography while the implicit method considers the epipole correspondence also along with the three points. Alternatively point and line correspondence can also be used for estimating homography.

Four matching point estimation: The implicit method using four point correspondences is a basic methodology using the DLT algorithm (more correspondences, the better estimation) (Dubrofsky 2007).

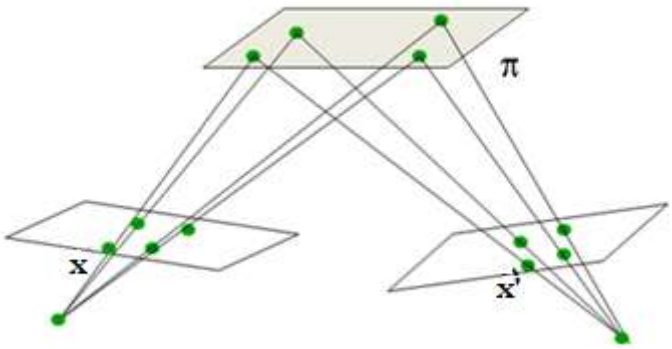


Figure 14: four point estimation of homography

Consider images of three non collinear points in both views and the fundamental matrix. The fourth point is the epipole in the both views. Four points make the basis for the projective plane and is shown in Figure 14.

The homography H induced by the plane formed by these points can be estimated knowing these four points.

$$\boldsymbol{x}' = \boldsymbol{H}\boldsymbol{x}$$

The Direct Linear Transformation (DLT) Algorithm

Consider a single point correspondence between x and x'. All the coordinates are expressed in homogenous coordinate system. The homography H, can be defined as a 2D 3x3 matrix,

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = {}^{u}/_{w} \;\; ; \;\; y' = {}^{v}/_{w}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$(x', y', 1)^T$ represents $x'$and $(x, y, 1)^T$ represents the point x.

In the equation from,

$$-h_{00}x - h_{01}y - h_{02} + (h_{20}x + h_{21}y + h_{22})x' = 0;$$

$$-h_{10}x - h_{11}y - h_{12} + (h_{20}x + h_{21}y + h_{22})y' = 0;$$

These two equations can be written in the matrix form,

$$Ah = 0$$

$$\text{A: } \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & x'x & x'y & x' \\ 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \end{pmatrix}$$

$$\text{h: } (h_{00} \quad h_{01} \quad h_{02} \quad h_{10} \quad h_{11} \quad h_{12} \quad h_{20} \quad h_{21} \quad h_{22})$$

(So, each correspondence provides two equations!)

Given 4 points in first view: $\{(x_i, y_i)\}_0^3$ and the 4 corresponding points in the second view: $\{(x'_i, y'_i)\}_0^3$.
The restriction is that no 3 points can be collinear. 4 points will yield a matrix A of size 8x9. The SVD of A is determined to obtain the solution for $Ah = 0$ and the homography h is estimated.

Now that the basic idea of homography has been established, let us perform homography on the given images for a simple plane and observe the results.

2.4.2 Homography with ground plane

The homography is explored for the images of the two cameras. The basic homography is to consider only the ground plane. The ground plane is a xy plane (z=0), and the homography induced by this plane can be easily estimated. So the world plane is a z=0 plane as shown Figure 15.

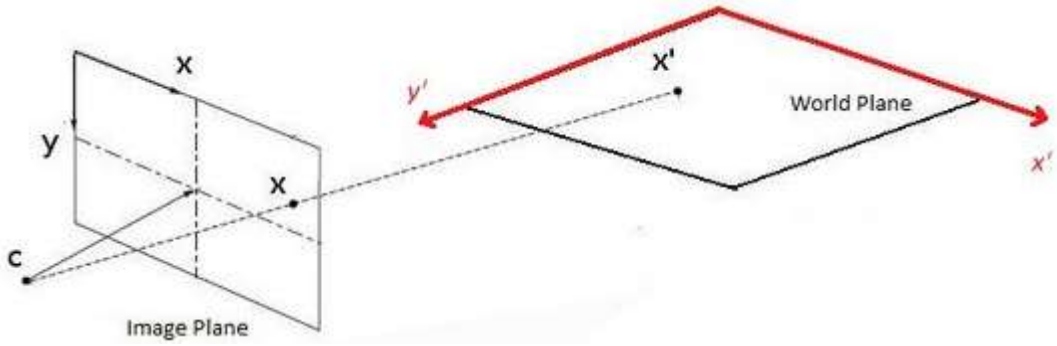


Figure 15: Homography induced by a ground plane

This plane is considered to estimate homography between the two images given. The ground plane selected is actually the basketball field in the world scene. So only the points on the field in both images can be related. All the other points in the scene like the audience, the players who do

lie on the ground get projected onto some points on the second image that will not correspond to the 3D reality.

Figure 16: Example of ground plane homography in between the two images

In Figure 16, the centre point of the field is mapped to the exact point in the second image. But, the point on the player in the first image cannot me mapped onto the same player in second because the mapping (homography) is estimated for ground plane (field region) only. So every point gets projected onto the ground and hence the corresponding point to the player actually gets projected onto ground.

To further continue experimenting with homography, first, only the ground regions are chosen in both the images and all other regions are eliminated.

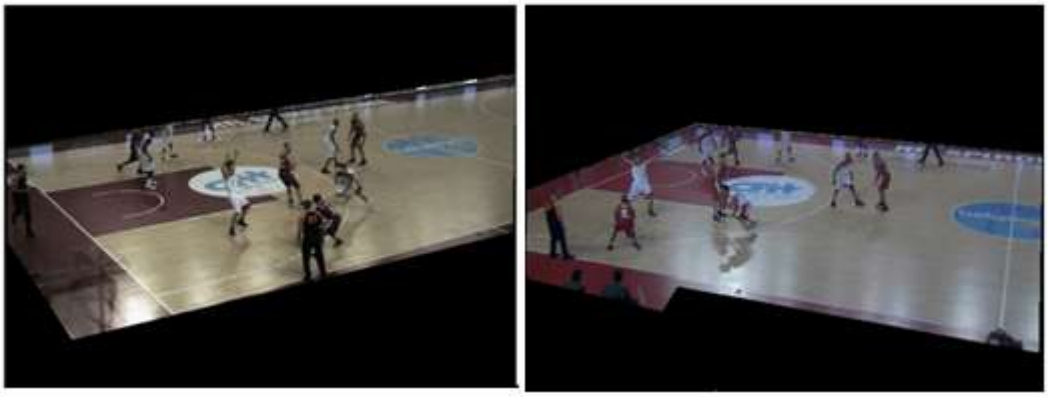


Figure 17: Ground regions of both images

The homography estimated for the specified world plane is applied to every pixel on the first image (only ground) to obtain the corresponding pixels in the second image. Figure 18 represents the points in second image that correspond to the points in the first image.

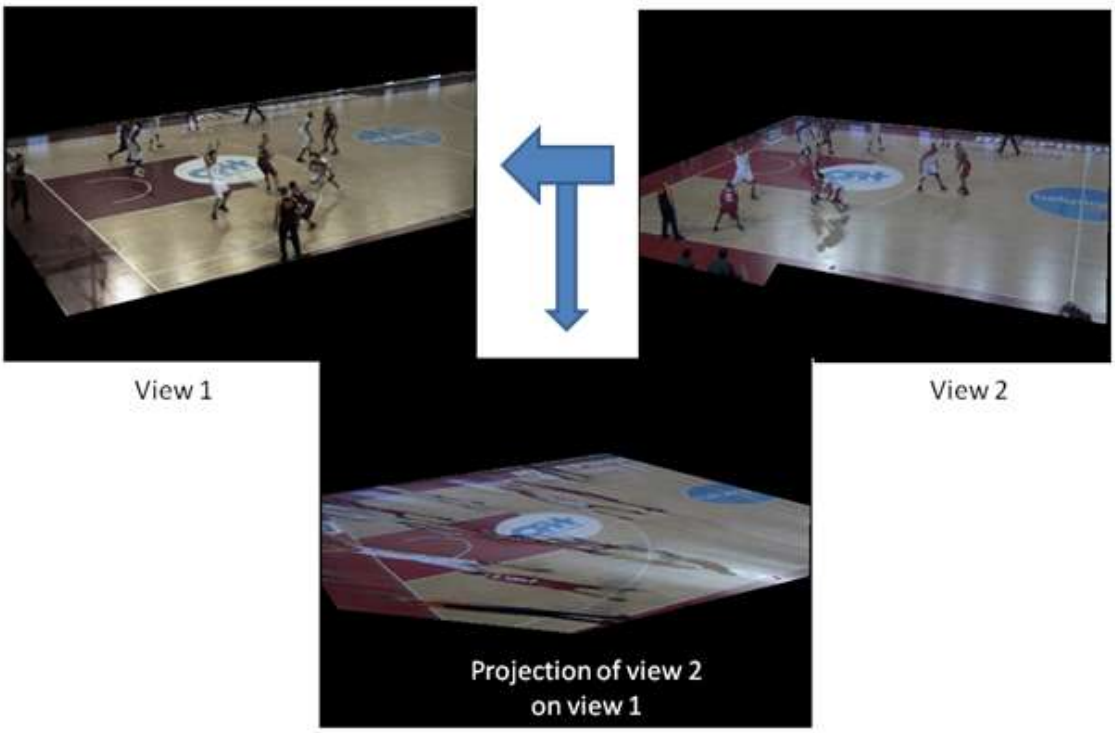


Figure 18: Homography projection of view 2 on view 1

An important observation is about the dynamic regions or the players. The players appear to be projected on to the ground (they appear to have flattened onto the field instead of being vertical). This is because of the depth of the plane. The importance of depth was discussed in segment 2.4. The depth is fixed for the z=0 (ground plane) plane. So, the plane is being forced onto the world scene and all the projections are considered from this plane only. The plane induced homography projects all the points at this depth only. This homography is applied to the whole scene irrespective of the players (who do not belong on the ground plane, except for their feet). Hence all the points are projected onto the ground plane, even the players. This gives the appearance of players lying on the ground.

Comparison

The ground plane homography verified that, the homography is plane dependent only. The entire ground region is reconstructed properly while the points not on the plane (players) are projected as they are on the ground. As seen in Figure 18, the first image and the projection of second on the first image plane are perfectly aligned to each other.

The first two images in Figure 19, actually represent the same scene; and hence they should exactly be the same (ignoring players). If this is true then the difference between these two images should yield a very low value (it should be zero but there will be differences in the player regions).

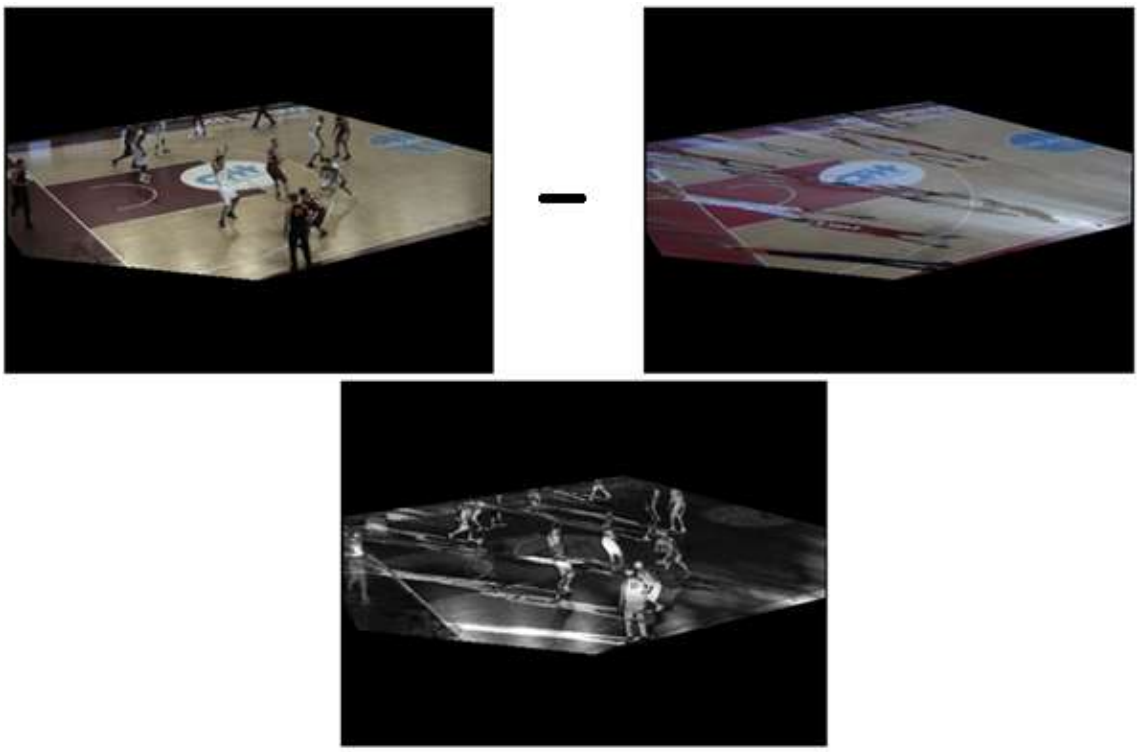


Figure 19: Difference map between the ground region in view1 and the corresponding homography in view2

The pixel by pixel subtraction should yield 0 (black) for all the ground regions and 255 (white) for the player regions. But this is not the case. The ground regions do not give a zero difference even though both images represent the same scene. Figure 19 shows the difference map for these two images. It is the normalized pixel to pixel difference for the images and it can be observed that, it is not black (0) for all the ground areas but has some value greater than 0.

Problem

The same area from the two views provide a difference map having values different than zero thus proving the fact that both possess different color information. The color information is the only way to check if the plane induced homography provides good correspondences between the images. The color uniformity is highly important for texture mapping. Trying to find best correspondences for players (for player reconstruction) now proves difficult.

2.4.3 Challenge

The biggest challenge now is to eliminate the color un-uniformity between the two images. The difference in color can be so clearly seen in the two images from the two cameras (Figure 20). Even though both the cameras are capturing the same scene, the images received by both (of the field, the players, all of the regions of the same scene) appear to be of different color.

Figure 20: The images from the two cameras

The whole basis of player reconstruction in a virtual viewpoint lies in devising a best 3D model for each player by comparing the best correspondence of the players between the two views. The color information is used to synthesize the texture of the models. So, if the color information is different, the reconstruction will be highly erroneous.

So, before proceeding towards player homography and reconstruction, this main problem of color difference between different camera images has to be solved. This is addressed in the next chapter where different color correcting methods are introduced to obtain color uniformity.

Chapter 3

Color calibration

Color uniformity across the camera images is important to ensure good reconstruction results in computer vision applications working with multiple cameras. As mentioned in the previous chapter, the inability to find efficient correspondence between the two camera images is due to the color difference in the two images. Even though the two cameras are imaging the same scene, significant discrepancies between the luminance components as well as the chrominance components of the different camera views are observed. This is because different cameras have different color gamut and different calibration. The color un-uniformity is visually noticeable in the images from the two different cameras in Figure 21.

View1 **View2**

Figure 21: Images from the two cameras

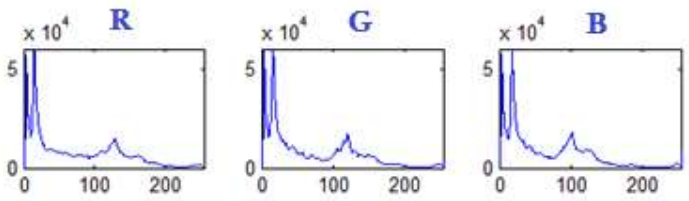


Figure 22: R, G, B histograms of view1

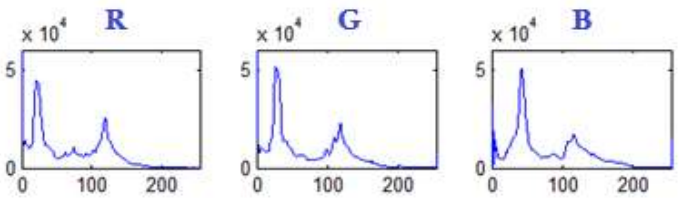


Figure 23: R, G, B histograms of view2

The color un-uniformity is further highlighted in the difference in the histograms in Figure 22 and Figure 23 respectively.

There can be two ways of correcting these camera images.

1. Post processing: color correction of each image obtained from different cameras.
2. Pre processing: camera color calibration.

Post processing refers to methods that are implemented on the acquired images. The images from all the cameras present in the multi view environment are post processed to attain color uniformity between them. For the post processing, each and every frame received has to be color corrected as and when it is received by the camera. Moreover this has to be done to frames obtained from all the different cameras present in the system. This does prove disadvantageous in real time scenarios like sports video broadcasting, where there will be an added delay because of this post processing step.

Pre processing, on the other hand, makes sure images obtained from the cameras are already color corrected. So the pre processing actually happens on the camera. It proves more advantageous than post processing color correction because the frames obtained by all the cameras are already color corrected and hence eliminates the additional delay. But this method requires all the cameras to be color calibrated before using (or mounting) it to record any event.

3.1 Post processing

Two different techniques are presented in this segment of post processing color calibration. The first method is the Color balancing, where the effects of source lighting on the camera are taken into consideration and an effort is made to reduce these effects. The second technique discussed is the Histogram Matching, which as the name suggests, tries to equalize the histograms of all the images captured to one particular reference image to obtain histogram matched images.

3.1.1 Color balancing

Color balancing is the adjustment of the intensities of colors. The main goal is to make colors have the same appearance in the image as is present in the real scene. The dominant part of color balancing is to reproduce the neutral colors (white/gray/black) perfectly and hence, this method is also called White Balancing.

White balance is the process of removing unrealistic color casts from digital images, mostly due to the acquisition condition (in particular lighting geometry and illuminant color)(Simone Bianco 2007).

Most light sources are not 100% pure white but have a certain color temperature, expressed in Kelvin. Color temperature is a way of measuring the quality of a light source. Normally, our eyes compensate for lighting conditions with different color temperatures but a digital camera needs to find a reference point which represents white. All the other colors are then calculated based on this reference point. Color cast is when everything in the image appears to have been shifted towards

one color or another. For instance, if a halogen light illuminates a white wall, the wall will have a yellow cast, while in fact it should be white.

Balancing in Cameras: When pointed to a reference point, certain cameras can calculate the difference between the current colour temperature of that reference point and the correct colour temperature of a white point and then shift all colours by that difference.

The reference generally used for digital cameras are a pure white object (card) or a gray card. A gray card is a flat object of a neutral gray color with flat reflectance spectrum and has 18% reflectance across the visible spectrum, and a white reverse side which has 90% reflectance.

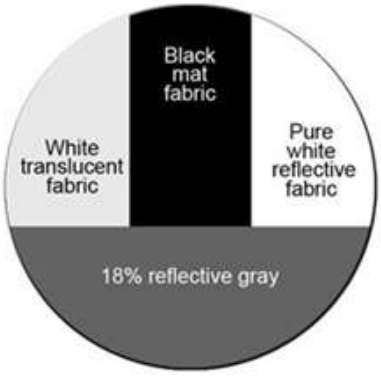


Figure 24: The reference card used by cameras for white balancing (18 % gray card)

But not all cameras are equipped with white balancing features, and hence the process has to be performed on the images acquired. The color casts on the images can be removed by performing white balancing on the images.

Von Kries hypothesis (Fairchild 2005)

The images captured with a wrong/no white balance from a camera need to be balanced. White Balancing is performed on a three component image using a 3x3 transformation matrix. Johannes von Kries, (who proposed the theory of rods and three color-sensitive cone types in the retina (Bjorn Stabell 2009)) motivated a transformation using LMS color space. LMS color space represents the effective stimuli for the Long-, Medium-, and Short-wavelength cone types that are modelled as adapting independently.

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} k_L & 0 & 0 \\ 0 & k_M & 0 \\ 0 & 0 & k_S \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix}$$

L, M, S and L', M', S' are the initial and post-color balanced cone tristimulus values and k_L, k_M, k_S are the scaling coefficients. The scaling coefficients are determined by choosing a neutral reference point.

Reference point

- **White point (RGB scaling)**: Viggiano's measure of scaling camera RGB considers white point on the image acquired as a reference point.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 255/R'_w & 0 & 0 \\ 0 & 255/G'_w & 0 \\ 0 & 0 & 255/B'_w \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

where R, G, and B are the color balanced red, green, and blue components of a pixel in the image; R', G', and B' are the red, green, and blue components of the image before color balancing, and R'_w, G'_w and B'_w are the red, green, and blue components of a pixel which is believed to be a white surface in the image before color balancing.

- **Gray world** : The gray world algorithm assumes that given an image of sufficiently varied colors, the average surface color in a scene is gray and hence the shift from gray of the measured averages on the three channels corresponds to the color of the illuminant (Simone Bianco 2007). Three scaling coefficients, one for each color channel, are therefore set to compensate this shift. A custom made ideal gray reference, which reflects all colors in the spectrum equally, and can consistently do so under a broad range of color temperatures is the reference point, and can be used to calculate the scaling coefficients in the transformation matrix.

Performing RGB scaling (white reference point) for the given images

The images are acquired from the two cameras. A reference point (a point believed to be white in the image is chosen). This point gives the R'_w, G'_w and B'_w needed for the scaling factors in the transformation T.

$$\text{T=} \begin{bmatrix} 255/R'_w & 0 & 0 \\ 0 & 255/G'_w & 0 \\ 0 & 0 & 255/B'_w \end{bmatrix}$$

This transformation is obtained to all the pixels in the image to obtain a balanced image. Figure 25 shows the camera images before and after white balancing.

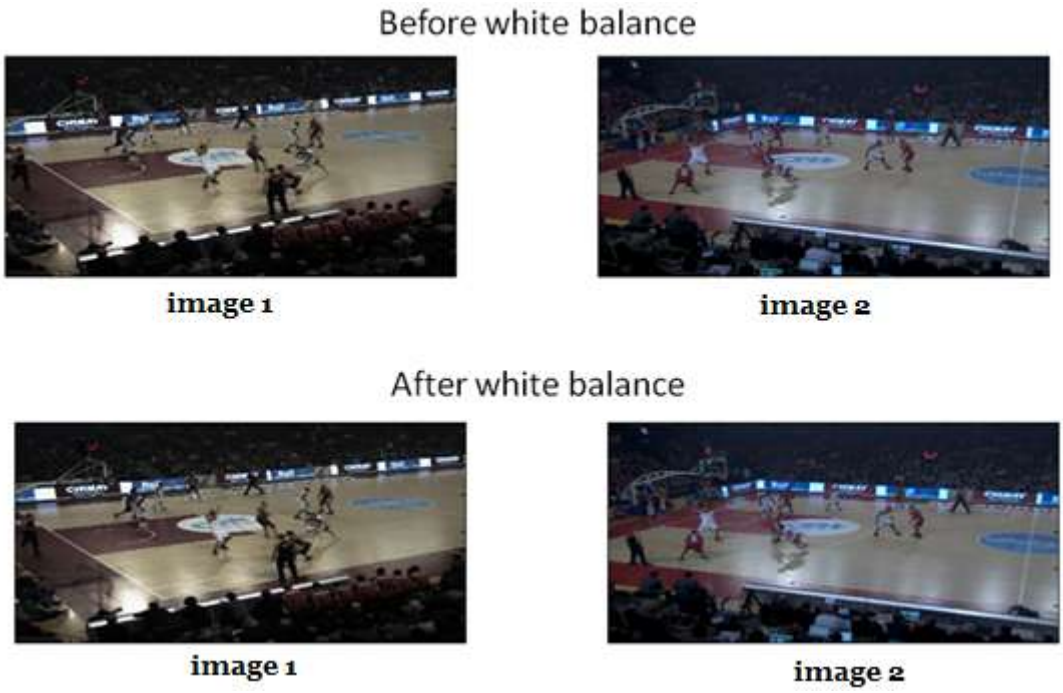


Figure 25: The images of the two cameras before and after white balancing

Visually, there is not much difference in the images before and after white balancing. The reasons for this might be: the source lighting being almost pure white does not produce any color cast on the images or because the camera or the given images might have been white balanced before. But for the images having a color cast, this post processing step of white balancing is highly effective.

[As an example to show the working of white balancing clearly, a color cast image is chosen.

Figure 26: Example of white balancing on a bluish color cast image

Figure 26 shows an image having a bluish color cast that has been removed after white balancing.]

Now returning back to the basketball images, the histograms of both the images are compared.

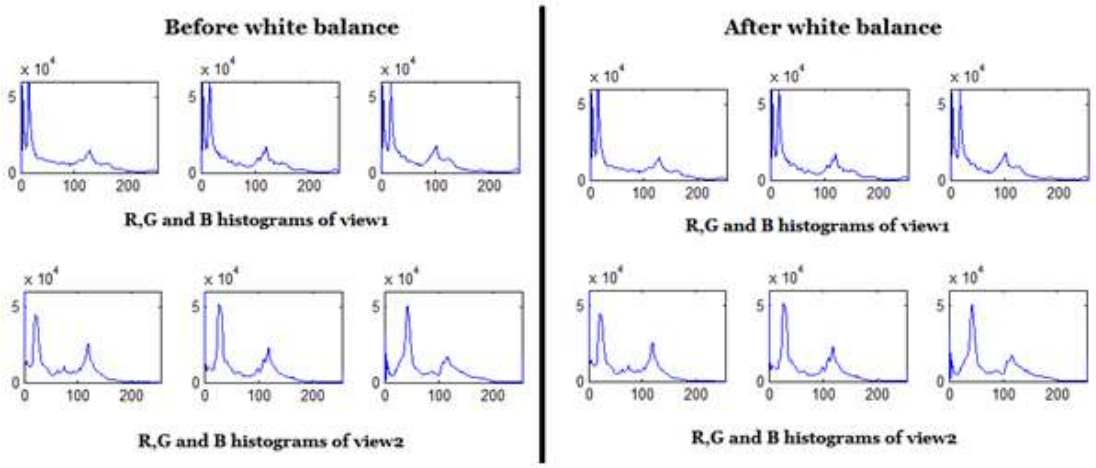


Figure 27: R,G,B histograms of the two images before and after white balancing

Figure 27 shows the histograms of the two images before and after white balancing. An important observation is white balancing does not help in matching the histograms of the two images.

The white balancing only removes unwanted color casts and compensates for the lighting and hence enhances images from each camera independently from the other. It does not bring color uniformity between the images of different cameras. Histogram matching is performed to try to get this uniformity between all the images from different cameras.

3.1.2 Histogram matching

Histogram matching, as the name suggests, tries to adapt the histogram of one image with respect to the histogram of another image. This method works well for the compensation of luminance and chrominance components in multi view images (Ulrich Fecker 2006). The method requires two images: reference image and the target image. The target image is the image from the first camera, and the reference image is the image form the second camera as shown in Figure 28.

Figure 28: Images from the two cameras with image1 as the target image and image2 being the reference image

The histogram of the reference image is tried to be adapted to that of the target image, such that in the end both reference and target images have the same histogram. In this method, the

cumulative histogram of the target image is made to adapt with the cumulative histogram of the reference image.

The procedure is exemplarily shown for the luminance component; the same is then applied for the chrominance components. First, the histogram is calculated as,

$$h_R[v] = \frac{1}{w.h}\sum_{m=0}^{h-1}\sum_{n=0}^{w-1}\delta[v, y_R[m,n]]$$

$$\text{with } \delta(a,b) = \begin{cases} 1 & if\ a = b \\ 0 & else \end{cases}$$

$y_R[m,n]$ is the amplitude of the luminance of the reference image and w and h are the width and height of the image respectively. The next step is to calculate the cumulative histogram.

$$c_R[v] = \sum_{i=0}^{v} h_R[i]$$

Similarly, the cumulative histogram of the target image is $c_T[v]$.

The next step is to create a mapping function that adapts $c_R[v]$ with respect to $c_T[v]$. The mapping is found my matching number of occurrences in the target image to the number of occurrences in reference image.

$$M[v] = u$$

$$such\ that\ \ c_T[u] < c_R[u] < c_T[u+1]$$

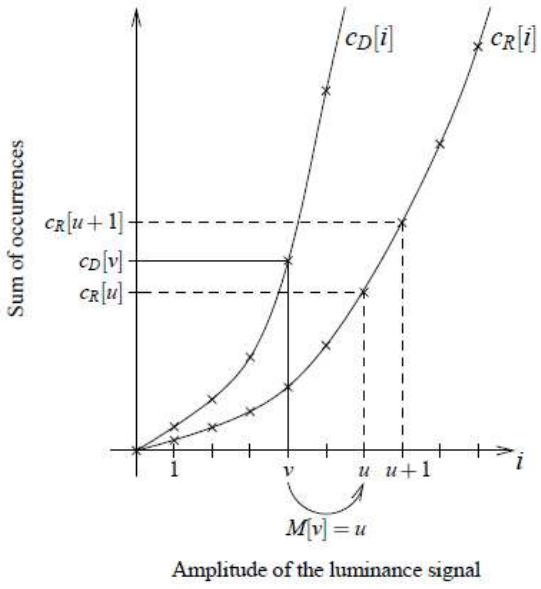


Figure 29: Mapping function used in the histogram matching algorithm

Figure 29 shows how this mapping is performed between the two cumulative histograms.

This mapping is applied to the reference image to obtain the converted reference image,

$$y_{converted}[m,n] = M[y_R[m,n]]$$

This method is applied for histograms of the both luminance and chrominance components. The cumulative histograms of all the three components of the two images are shown in Figure 30. In each case the cumulative histogram of reference is matched to that of the target by applying the mapping function.

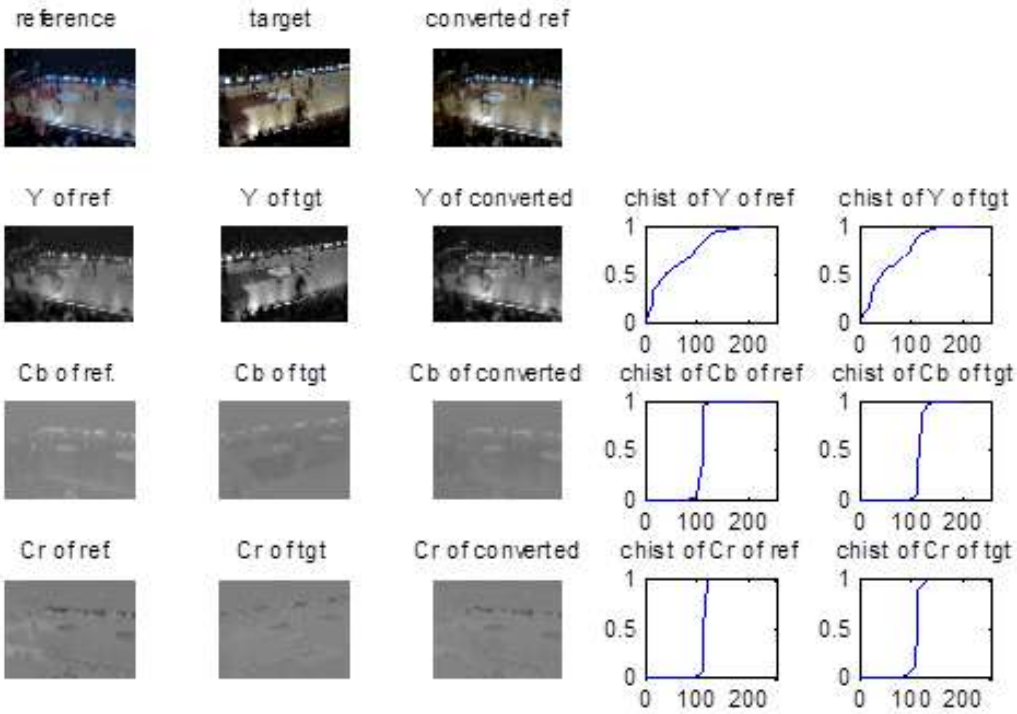


Figure 30: The luminance and chrominance histograms of the reference, target and the converted reference images

The histograms of the two images (from two different cameras) before and after applying histogram matching are shown in Figure 31. As seen, the histograms of the second image are almost similar to that of the first image.

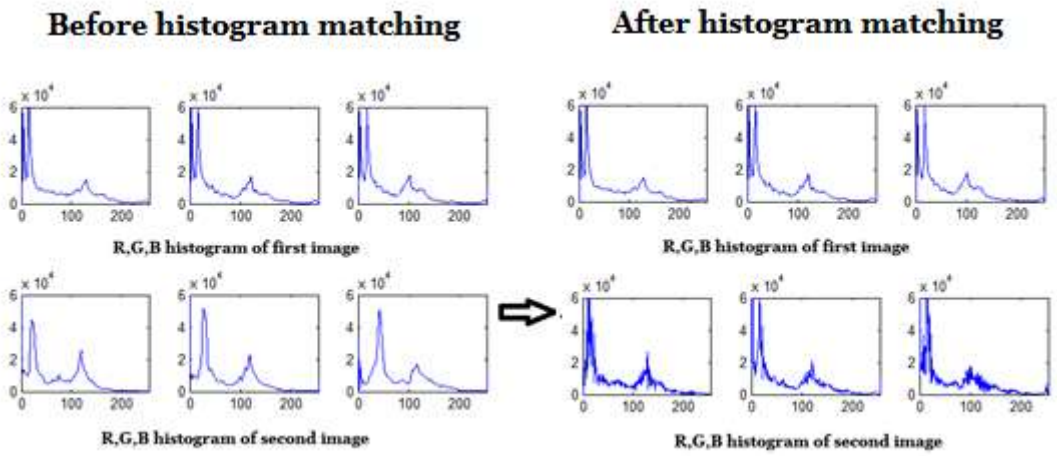


Figure 31: R,G,B histograms of the two images before and after applying histogram matching algorithm

The images are shown Figure 32, and even though the colors are not exactly the same visually, the difference in colors has been reduced. The second image has been tried to adapt to the colors of the first image and the change is quite clearly visible.

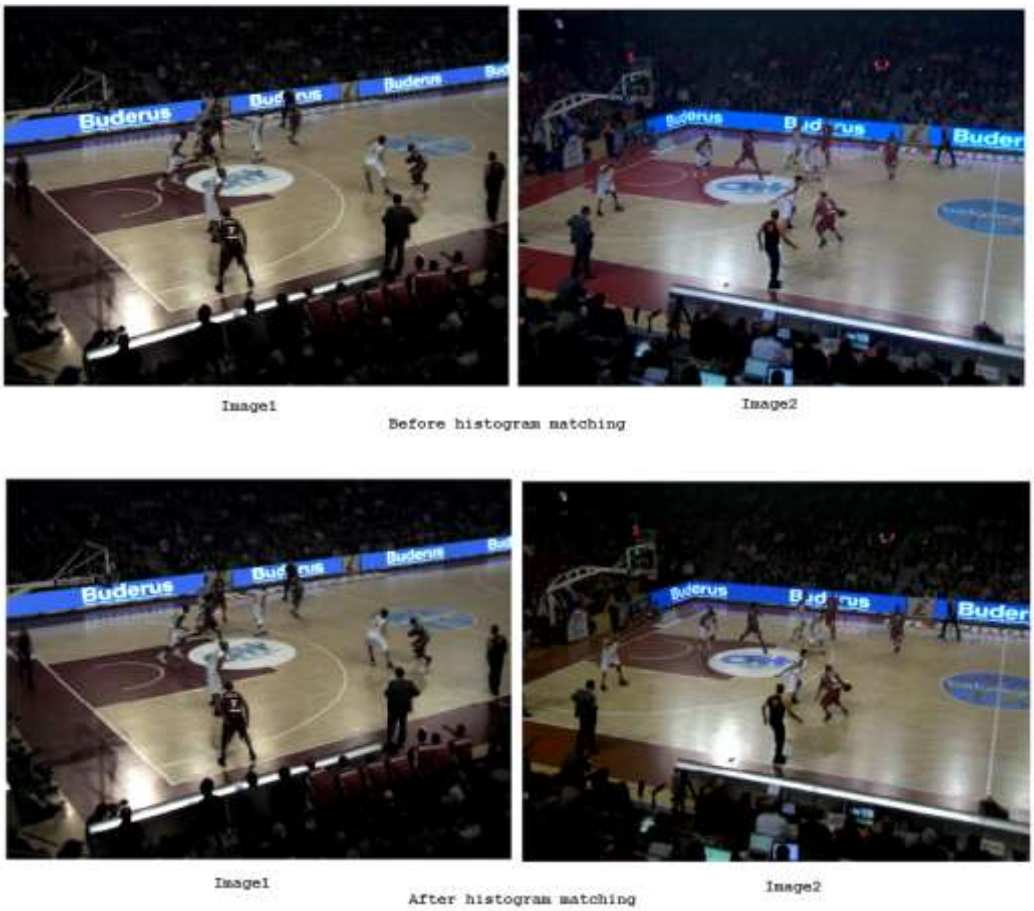


Figure 32: The two images before and after histogram matching

Computational complexity: The described algorithm can be applied to multi- view sequences in the following way: One camera view is chosen as the target view. All other camera views are corrected so that their histograms fit the histogram of the chosen target view. So, even though the computational time for a single frame is very low (in the scale of few milliseconds), the total time taken will depend on the number of frames being received. Moreover only two cameras are considered here. A multi view application can consist of a lot of cameras and the images received by all these cameras have to be histogram matched with one another which will definitely add a certain amount of delay which might affect real time applications.

Both the methods discussed here (Color Balancing and Histogram Matching) can be implemented only after acquiring the images from the camera, i.e. as a post processing step. The algorithms have to be applied to each and every frame acquired by all the different cameras in a multi view environment. This is quite a cumbersome task. A better approach would be to design cameras that can produce images without any color differences, thus giving rise to the topic of Camera Color Calibration. This can be considered as the pre processing step, because the work lies in calibrating the cameras that acquire the images and not on the images obtained.

3.2 Pre Processing: Camera Color Calibration

As discussed above, the best way to achieve color uniformity between images from different cameras would be to color calibrate all the cameras before taking the images. Different cameras, even of the same make, may have different color gamuts. One way to make images from different cameras look similar is to calibrate them so that, under a common color model, the same colors correspond to the same numerical values. Camera color calibration helps all the cameras in a multi view environment capture scenes without any luminance or chrominance distortions.

The classic approach to color matching is closed-loop calibration, where specific pieces of hardware are calibrated to each other (the camera, the printer, the scanner). A common calibration method that is used for calibrating a scanner to a printer is scanning a known target, printing it, and then scanning the printout again. A comparison is performed between the two scanned images, and different parameters are changed in the calibration loop to compensate for the difference: adjusting the scanner parameters or adjusting the printer parameters. A similar approach can be used for calibrating multiple cameras, by changing camera parameters such that all cameras will be calibrated to each other, the desired result which is highly useful for computer vision applications (Ilie 2004).

Ideally, the comparison should be performed for all the colors present in the images. To construct a lookup table which contains all the correspondences between the same colors in the two images is impractical. However, a limited number of color samples could be used and interpolated to reconstruct other colors in the color space. The trade-off between the number of samples and the accuracy of the approximation depends on the application and the devices being calibrated. A good pattern (Calibration pattern) should be used as a standard for comparing colors.

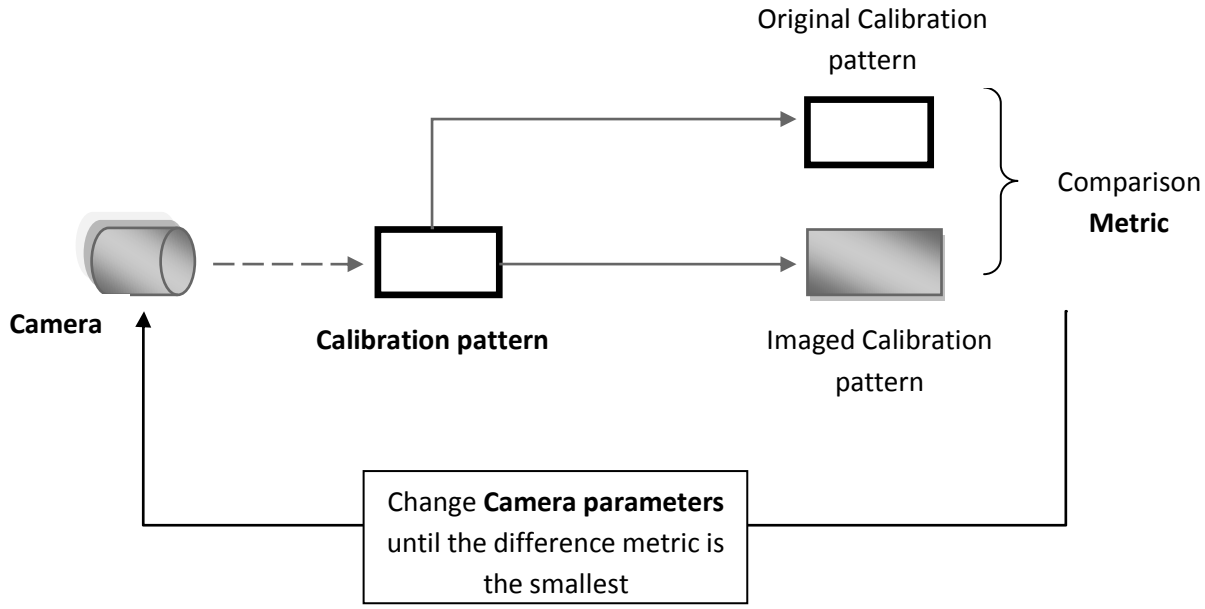


Figure 33: General schematic of camera color calibration

Figure 33 shows the general schematic of camera color calibration. So, the main methodology is to use a good color index pattern and use the camera to be calibrated to take its image. Then the

original pattern and the imaged pattern are compared and different camera parameters are changed to get the colors of the imaged pattern as similar as the colors on the original pattern. All the blocks in the schematic are explained in detail in the following topics.

3.2.1 Calibration Pattern

Calibration pattern is a color space containing all the colors to be compared. The calibration pattern should contain enough colors to provide an adequate number of samples for the computation process. The pattern shown in Figure 34 contains 63 colors with combination of R, G and B values on a linear scale on each color scale.

Figure 34: Basic color calibration pattern of 63 colors

The pattern shown in Figure 35 : Gretag Macbeth Color Checker Color Rendition Chart is the professional color target used by experts in the field of camera and photography. It is the Gretag Macbeth color chart and contains all the colors usually encountered in nature. The Color Checker Color Rendition Chart is a very well known chart with an array of 4 x 6 color patches, and is an icon of the imaging industry. It was presented in a 1976 article by C. S. McCamy and his colleagues from the Macbeth Company.

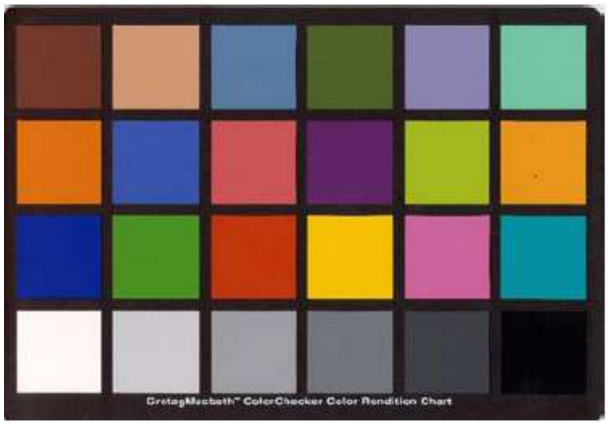


Figure 35 : Gretag Macbeth Color Checker Color Rendition Chart

The Color Checker consists of a series of six gray patches, plus typical additive (Red-Green-Blue) and subtractive (Cyan-Magenta-Yellow) primaries, plus other natural colors such as light and dark skin, sky-blue, foliage, etc. The chart's color patches have spectral reflectance intended to mimic those of natural objects such as human skin, foliage, and flowers, to have consistent color

appearance under a variety of lighting conditions, especially as detected by typical color photographic film, and to be stable over time. The degree of metamerism (the quality of some colors that causes them to appear differently under different light sources) was also very small when directly comparing the chart to the natural colors.

Index	Description	Munsell Notation	CIE xyY	Manufacturer's sRGB color values[6]
Row 1: Natural colors				
1	Dark skin	3 YR 3.7/3.2	0.400 0.350 10.1	#735244
2	Light skin	2.2 YR 6.47/4.1	0.377 0.345 35.8	#c29682
3	Blue sky	4.3 PB 4.95/5.5	0.247 0.251 19.3	#627a9d
4	Foliage	6.7 GY 4.2/4.1	0.337 0.422 13.3	#576c43
5	Blue flower	9.7 PB 5.47/6.7	0.265 0.240 24.3	#8580b1
6	Bluish green	2.5 BG 7/6	0.261 0.343 43.1	#67bdaa
Row 2: Miscellaneous colors				
7	Orange	5 YR 6/11	0.506 0.407 30.1	#d67e2c
8	Purplish blue	7.5 PB 4/10.7	0.211 0.175 12.0	#505ba6
9	Moderate red	2.5 R 5/10	0.453 0.306 19.8	#c15a63
10	Purple	5 P 3/7	0.285 0.202 6.6	#5e3c6c
11	Yellow green	5 GY 7.1/9.1	0.380 0.489 44.3	#9dbc40
12	Orange Yellow	10 YR 7/10.5	0.473 0.438 43.1	#e0a32e
Row 3: Primary colors				
13	Blue	7.5 PB 2.9/12.7	0.187 0.129 6.1	#383d96
14	Green	0.1 G 5.4/9.6	0.305 0.478 23.4	#469449
15	Red	5 R 4/12	0.539 0.313 12.0	#af363c
16	Yellow	5 Y 8/11.1	0.448 0.470 59.1	#e7c71f
17	Magenta	2.5 RP 5/12	0.364 0.233 19.8	#bb5695
18	Cyan	5 B 5/8	0.196 0.252 19.8	#0885a1
Row 4: Grayscale colors				
19	White	N 9.5/	0.310 0.316 90.0	#f3f3f2
20	Neutral 8	N 8/	0.310 0.316 59.1	#c8c8c8
21	Neutral 6.5	N 6.5/	0.310 0.316 36.2	#a0a0a0
22	Neutral 5	N 5/	0.310 0.316 19.8	#7a7a79
23	Neutral 3.5	N 3.5/	0.310 0.316 9.0	#555555
24	Black	N 2/	0.310 0.316 3.1	#343434

Figure 36: Colors of Gretag Macbeth Color Checker in CIE xyY encoding, Munsel notation and in sRGB values

Gretag Macbeth gives numeric values for the twenty-four patches (Figure 36) in CIE xyY encoding (chromaticity values *x* and *y* and the *Y* luminance value), Munsel Notation (Hue Value/Chroma), and sRGB (Unlike most other RGB color spaces, the sRGB gamma cannot be expressed as a single numerical value.)

The color chart chosen for the experiment is Gretag Macbeth Color Checker Color Rendition Chart of size 8.25 x 11 inch.

3.2.2 Camera

The camera used for the experiment is the Basler Ace GigE Camera (Basler acA640-90gc image format).

It is an economical Gigabit Ethernet camera. The camera is fully compliant to the latest revision of GigE Vision® and GenICam™ standard and makes use of the latest building blocks such as large FPGA, high resolution analog front ends and supporting components. It full range of CCD and CMOS sensors, from VGA up to 5 megapixel resolution. Image quality is digitized with precision-analog front ends with up to 14 bit resolution (CCD sensors). It provides up to 1 Gbit/s GigE Vision interface and support for Power over Ethernet (PoE).

3.2.3 Camera Parameters

The quality of an image is determined by many things: Illumination, lens and camera parameters. This section speaks in brief about the different camera parameters that need to be changed during color calibration.

Gain: Gain determines the amplification of the CCD's output signal. The gain of a CCD camera is the conversion between the number of electrons (e-) recorded by the CCD and the number of digital units contained in the CCD image. The digitization happens in the analog to digital converter (ADC) and the units can also be called ADU (analog-to-digital unit).

It is useful to know this conversion for evaluating the performance of the CCD camera. A gain of 8, means that the camera digitizes the CCD signal such that each ADU (Analog-to-Digital unit) corresponds to 8 photoelectrons. The system gain of a camera is typically set so that the full well (each CCD is designed to hold only so many electrons within a pixel before they start to leak outwards to other pixel) of the CCD matches the full range of the digitizer (at 1x gain). The camera's gain needs to be selected to meet the needs of a given application. For example, the gain can be increased to 4x when the application is photon starved and a high-sensitivity mode is required. Alternatively, the gain can be reduced to 1/2x when the application is photon-shot-noise limited and a high SNR mode is required.

The amplification increases the contrast. A high gain, however, leads to noisy images.

Black Balance: The **black level** is the output of the camera when not illuminated. Dynamic range is limited by the darkest tone where texture can no longer be discerned; and this is the black level.

Gamma: Gamma is an implicit or explicit transfer function that maps input intensity to output intensity, usually in a nonlinear way. Gamma correction is the name of a nonlinear operation used to code and decode luminance or tristimulus values in video or still image systems. Gamma correction is, in the simplest cases, defined by the following power-law expression:

$$V_{out} = AV_{in}$$

where, *A* is a constant and the input and output values are non-negative real values.

Gamma correction compensates for the fact that video display devices (such as televisions) inherently convert image signals to light intensity in a nonlinear fashion. The non-linear behaviour of picture tubes is compensated by this factor. Gamma increases or decreases the middle gray levels. However, digital cameras record light using electronic sensors that usually respond linearly.

Exposure: The shutter determines the CCD‘s exposure time. Exposure is the total amount of light allowed to fall on the photographic medium. A photographic film (or sensor) has a physically limited useful exposure range, sometimes called its dynamic range. If, for any part of the photograph, the actual exposure is outside this range, the film cannot record it accurately. For example, out-of-range values would be recorded as "black" (underexposed) or "white" (overexposed).

This is a very sensitive parameter which determines the amount of light entering the camera and altering this would change the quality of the images drastically and hence is better to be set at a default value.

Balance Ratio: The color balancing is carried out by specifying this parameter. The color casts and other colored artefacts on the images can be corrected by altering the balance ratio. This was explained in detail in topic 3.1.1. This parameter varies the degree of red and blue in the image to achieve a lifelike color representation. Most cameras only provide one white balance parameter and so, increasing the degree of red leads to a decrease of blue and vice versa. High quality cameras offer two parameters and thus allow adjusting independently the degree of red and blue:

There are some physical parameters of the camera and some image processing parameters and the relation between these parameters will be interesting to observe. The parameters may not all mutually independent from one another. There might be some correlation between different parameters, i.e., altering one parameter can affect another parameter. Hence a combination of all the parameters needs to be tested in this experiment. These observations are validated in the result section of the chapter.

Parameters	Min	Max
Gain	190	1023
Black Level	10	1023
Gamma	1	4
White Balance	1	4
Balance Ratio	1	15
Exposure	1000	1000000

Table 1: Camera parameters

Table 1 shows all the parameters that the Basler Ace camera allows us to change. The range of values that each parameter can be varied within is also given.

3.2.4 Camera Interface

ActiveGige is an SDK and ActiveX control designed for rapid application development tools such as matlab. Provided is GigeViewer application allowing operation of multiple cameras. This interface supports all GIGE vision cameras.

3.2.5 Methodology

The main components of the experiment have been explained and this section details the experiment procedure in detail. Figure 37 is similar to the general procedures shown in Figure 33 but is in more detail.

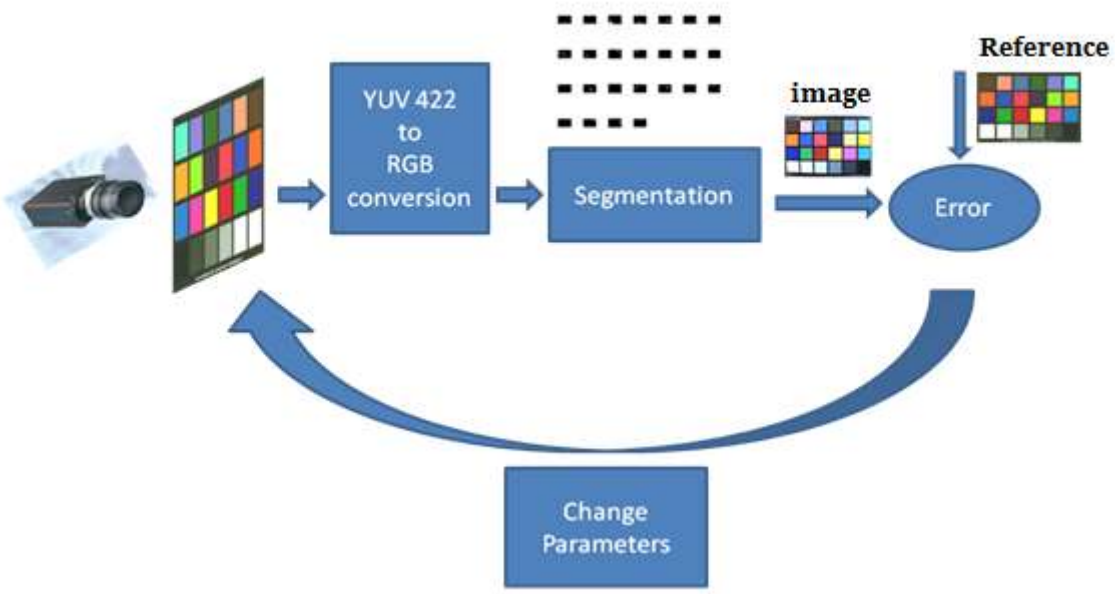


Figure 37: Detailed schematic of camera color calibration

1. **Acquiring image**

The images acquired by the camera need to be extracted onto the computer and processed. A Graphical User Interface displays the images acquired by the camera at each instant of time.

The type of image data received and the dimensions of the array are determined by the output format of the video. The camera interface gives the images in different formats: Mono8, YUV411, YUV422, YUV444, Bayer12, and Bayer16. The image obtained has to be converted to the RGB format to further process it. The YUV422 format is selected and the dimension of the data obtained is:

(0 to horizontal_width * 3 - 1, 0 to vertical_height – 1)

horizontal_width and vertical_height are the width and height of the image acquired. The image acquired here is in YUV422 interleaved format. The interleaved data is extracted and is then converted to the RGB (24 bit) format.

2. Segmentation

The Gretag Macbeth color chart has 24 colors and the RGB values of each color have been specified as shown in Figure 38. Figure 39 shows the image of the Gretag Macbeth color chart captured by the camera. The goal is to have these two images as similar to each other as possible, i.e. to establish correspondence between squares of the same color in camera color checker image and the standard color checker. As seen in the figures, the camera image is not the exact replica of the original. The color calibration can be done in any environment, under different lighting conditions depending on the application. The source lighting, the camera lens distortion (which causes the rectangular chart to appear curved) can cause uneven illumination within a square. Hence, no assumption is made on the shape of the squares (it is not assumed to be of the same area as the original color checker) while processing the camera image.

One way to compare them is to segment each color (each square in the chart) from the camera image and then average the pixel values across the square to obtain the RGB values of that particular color and then compare it with the standard value of the color specified by the chart. Another way is to just select a point within every square (as the entire square contains the same RGB value) and uses this value to compare, but this has a disadvantage. As mentioned before, the uneven illumination within a square can cause the pixel values to differ even for the same color. So the method of segmenting and averaging the value over the square is chosen.

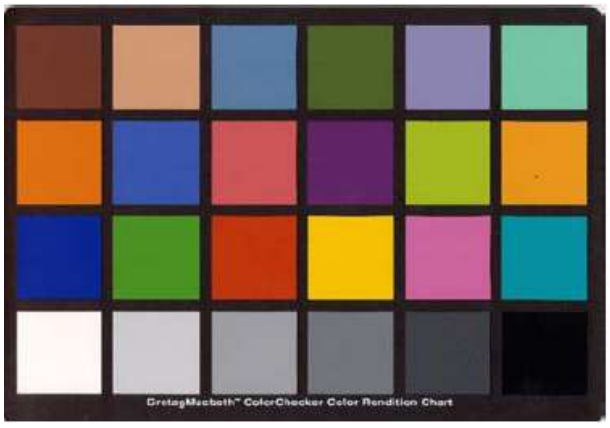


Figure 38: Original Gretag Macbeth color checker chart

Figure 39: the image of the Gretag Macbeth color checker chart

Several methods were tried to segment the color checker chart, namely, Region growing, K-NN algorithm and the K-Means algorithm. Both region growing and K-NN require some input by the user, certain pixels in each square to start the segmenting process while K-means is completely automatic. So K-Means algorithm is chosen to segment the image. The only input to the K means is the number of clusters needed, which is 24 in this case (24 squares to be segmented).

The result obtained is a crude segmentation of 24 different squares and also the black borders between and around the squares. The accuracy of the segmentation largely depends on the ambient lighting and aperture of the camera. If the colors are not well differentiated, then two squares can be considered of the same color and can be blended as one during segmenting. The solution is to get a good quality camera of the color checker chart just for segmenting at the beginning of the calibration. Once the segmented squares are obtained, morphological operations are performed to further enhance remove artefacts and unwanted segments. These are shown in Figure 40 and Figure 41 respectively.

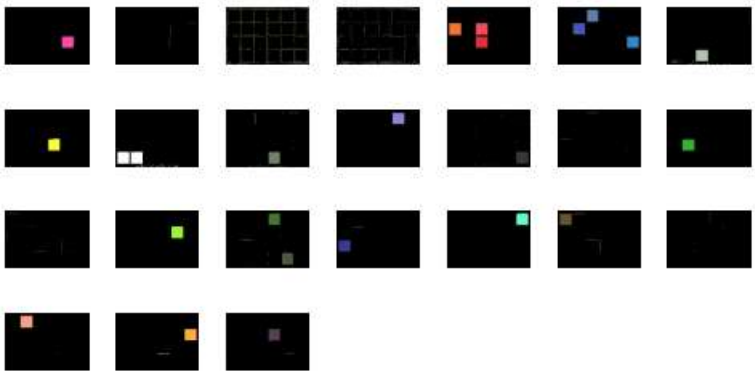


Figure 40: Segmentation with 24 clusters

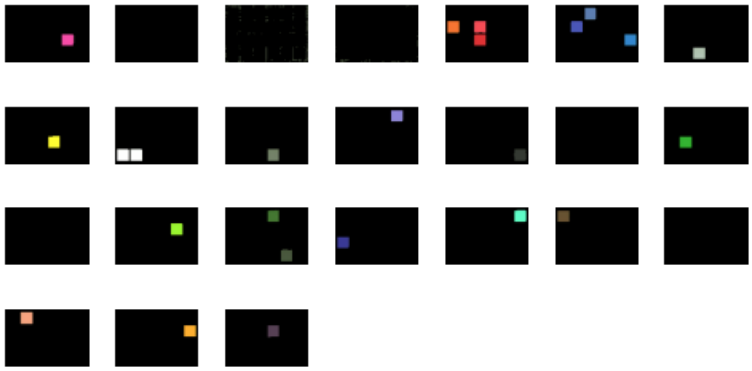


Figure 41: segmentation with 24 clusters after morphology

Figure 40 or Figure 41 shows the 24 clusters from K means algorithm. As can be seen from the figures, some clusters contain more than one square while the others don't have any (but there are 24 segmented squares in total). Each square in the cluster has to be separated and labelled appropriately. The segmented squares obtained are not in order, i.e., the K-means segmentation occurs iteratively and the resulting segmented blocks can be in any order and need not follow the 6x4 pattern of the color checker. So the next step is to label each of these squares based on the centroid information. After labelling the squares, the pixel values are summed up and averaged for the number of pixels available for each square in the camera image.

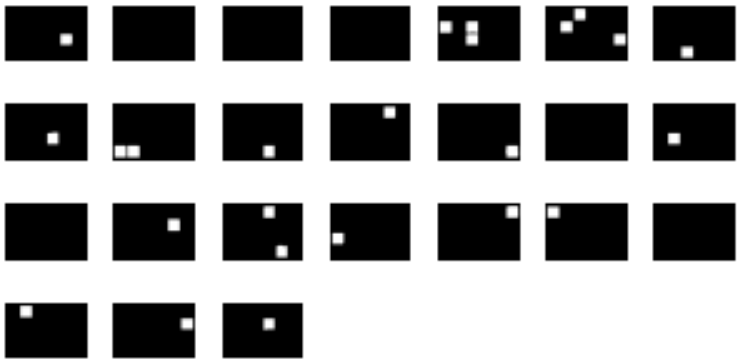


Figure 42: squares separation and labelling after obtaining centroid information from the segmented blocks

Note: This whole segmentation process occurs only once during the whole calibration. In the beginning, the segmentation is computed, and the information about the segmented squares (centroid,area..) is stored. Since the position of the chart is not moved once the calibration process starts, the segmentation step need not computed every time a new image is obtained. Instead the stored information is used to obtain the intensities of the segmented squares of the new received image. This saves ample amount of time.

3. Comparison

Once the pixel information (RGB values) is obtained for each color square of the image, it is compared with the RGB values of the original color chart. These values are already available by the manufacturer.

Figure 43: Gretag Macbeth color checker chart with RGB values

The RGB value for each square has been specified for the Macbeth color checker chart. Figure 43 shows the R, G, B value for each color (Ex: first square: Brown color has R=81, G=66 and B=52). The camera should reproduce the same colors, i.e. have the RGB values as close as possible to these.

The metric used is the difference between the images (for each color square). The camera parameters are changed each time and the difference between the two images is calculated.

$$\boldsymbol{metric} = \sum_{1}^{n} (I_{orig} - I_{cam})^2$$

I_{orig} = Intensity of a color square in the original color chart;
I_{cam} = Intensity of a color square in the camera image of the color chart;
n = number of squares (24)

The parameters that give the minimum metric are the set of parameters that the camera should be set to for proper color calibration. The minimization of the metric can be done in two different ways:

- **Brute Force Method**

This is the general approach where all the parameters are changed one by one and the difference is calculated. The parameters with the least error difference are the best parameters for that particular camera. But this is quite cumbersome; the user has to wait a while before getting the required parameter set.

The better way is to optimize and search for the parameters instead of trying all possible combinations at once.

- **Multi-resolution method**

To reduce the time taken by brute force method, down sampling method is used. The huge parameter set is down sampled and hence only few of the parameters are tested initially.

Parameters	Min	Max	Step size
Gain	190	1023	50
Black Level	10	1023	50
Gamma	1	4	1
Balance Ratio	1	15	1
Exposure (Limited Range)	1000 100000	1000000 700000	 50000

Table 2: camera parameters and step size chosen to vary them

For all the parameters (Table 2), step sizes are defined and each parameter is incremented to this amount every time. The number of combinations possible for these five parameters with the chosen step sizes: 74256 combinations!

Brute force approach runs for all these combinations to find the best possible combination of parameters (smallest error metric).

Down Sampling method reduces this parameter grid. If this huge parameter grid is down sampled by a factor of $2\hat{}5$, the parameter combinations to check reduce from 74256 to 2320. 2320 combinations are tried to obtain the best parameters (best point in the huge parameter grid of dimension 5). Now the search is made around this point, there are 32 points to check each in 5 dimensions. But again the brute force search is not used to search all the parameters around this point. Down sampling is utilized again around the point but with a lower scale ($2\hat{}2$), so $32/4 = 8$ points are checked in each dimension around the point to find the best parameter set

The results of the camera color calibration are discussed in the next section.

3.3 Results of Camera Color Calibration

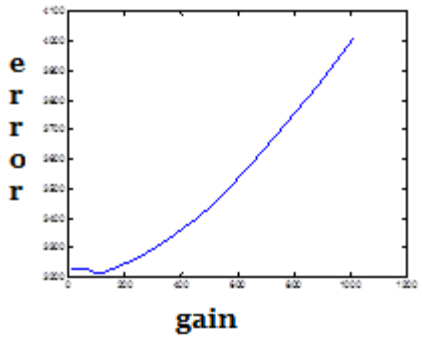
First the significant parameters that affect the quality of the image the most are found. Discarding the least significant parameters saves time when performing color calibration. Using these parameters, the camera color calibration is performed to get the best parameter set for this particular camera.

3.3.1 Significant parameters

First, the parameters are varied independently of each other and metric is calculated. The parameters that do not produce a significant change in the metric can be eliminated thus saving computational time.

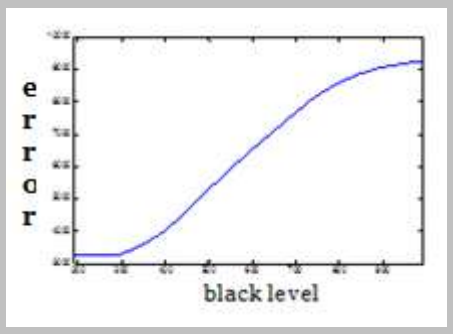
Gain

At lower values of gain, error is quite low. But the error increases at high values of gain. The image becomes too bright thus giving a large difference between the image and the original color chart.



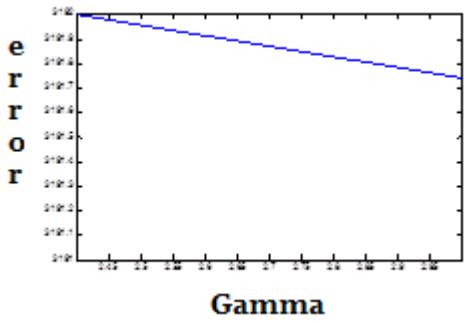
Black Level

Error increases with increase in black level parameter up to a certain point, after which it saturates.



Gamma

The error is almost constant for the whole range of gamma. The image does not change when this parameter is varied. So eliminating this parameter will not affect the end result in any way.



Balance Ratio

Error remains constant until a certain ratio and then increases.

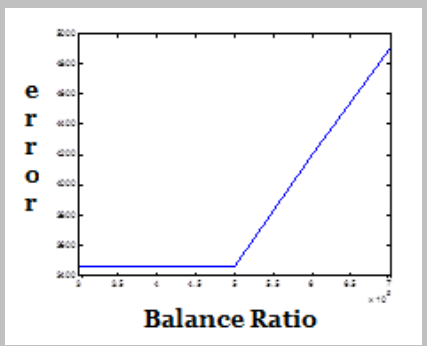
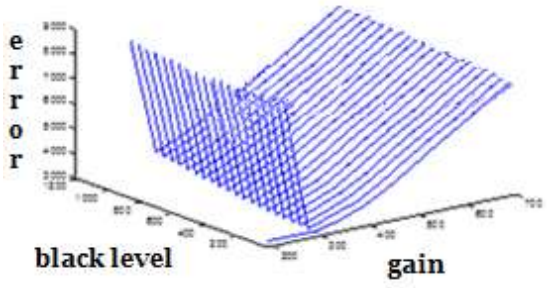


Table 3: metric vs. camera parameters

Some parameters are varied together and metric is calculated (Table 4).

Gain and Black Level

The graph to the right shows an instant correlation between gain and black level. Gain is a physical parameter while black level is an image processing parameter.
A high error is realised for all valued of black level when gain is high. A reasonable choice would be medium gain and a high black level.



Gain and Exposure

Gain and exposure are the parameters that affect the image quality drastically. As expected, when both gain and exposure are high, the error is high because the image gets noisier because of high gain and gets too bright because of over exposure.
An optimum balance between exposure time and gain should be found such that noise and motion blur are minimized in the resulting image.

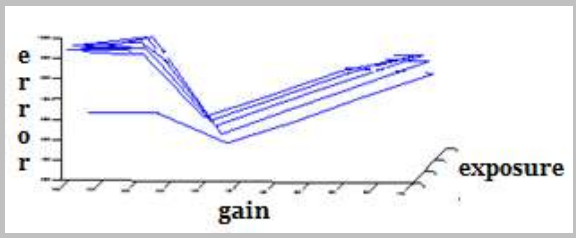


Table 4: correlation of camera parameters w.r.t metric

Exposure, as mentioned previously is a very sensitive parameter with a huge range. So, the exposure is not preferred to be changed during calibration. If it is, the range of valued should be limited. Gamma does not affect the result much either and need not be calibrated.

A combination of all these parameters is tested, and the set of parameters that gives the least error is the parameters the camera should be set to.

3.3.2 Color Calibration: indoor

The parameter set when the color calibration algorithm was applied to the camera indoors (room lighting) is shown in Table 5. A combination of all the five parameters were tried to find the best possible parameter set.

	Parameters	Min	Max	Step size	Best value
Significant Parameters	**Gain**	190	1023	50	490
	Black Level	10	1023	50	510
	Balance Ratio	1	15	1	4
Other parameters	Gamma	**1**	**4**	**1**	**1**
	Exposure	100000	700000	50000	400000

Table 5: Best parameter set for the camera

The whole calibration process depends on the number of combination of parameters and does not take more few seconds. But the segmentation that is carried in the beginning takes some time to iterate and segments all the blocks perfectly. If the lighting is good (to differentiate colors), the segmentation step occurs more fast. Also, some time is needed in converting the data received from the camera into the standard RGB format. Apart from these, the whole algorithm takes around 2 to 3 minutes.

Results from the camera:

The color calibration results highly depend on the application. The camera should be calibrated in the same place (same lighting) where it is supposed to be used later.

Figure 44: Image in a room before and after calibration

The images in Figure 44 show images taken by the camera when the camera was calibrated indoors. There is an improvement in the quality of the image. The black level and balance ratio have removes color casts and gain has improved the contrast.

Figure 45: Outdoor image taken from the camera which was calibrated indoors

To record outdoor events, the camera has to be calibrated outdoors in the sunlight. When a camera calibrated inside a room is used to take pictures of a scene outside, the result is not that good as can be seen in Figure 45. The process should be carried out with sunlight as the source to get the appropriate parameters.

Camera Color Calibration is a highly efficient way to achieve color uniformity in the images. It, being a pre processing process acts as an added advantage. The cameras can be calibrated before being used to record any event and hence does not add any delay in any real time applications. But the most important criterion is to calibrate the camera in the same place or with the same lighting as the event which the cameras will record later.

Chapter 4

Player reconstruction

The goal is to reconstruct the dynamic regions (players) in an intermediate view between any two cameras. Planes are used to model the players in the scene. Best plane, that effectively represents the player has to be selected. The best plane is the one that gives the best correspondence in the two views. As was mentioned in Chapter 2, the color un-uniformity hindered the process of finding accurate correspondences between the two images from the two cameras. This problem was solved by color calibrating all the cameras explained in Chapter3. So now that color uniformity has been established between different cameras, homography (explained in Chapter2) is revisited to obtain the best correspondence between the players which will help reconstruct the players in virtual viewpoints in between the cameras.

4.1 Player Homography

The idea of homography was introduced in Chapter 2. The player has to be modelled in 3D. 'Plane' is chosen as the modelling for the player even though the player is not flat because of the simplicity. Each player has a specific height and width, and the better option would be to place a cylinder or put multiple planes that would totally enclose the player which would provide more accurate representation of the player. But this advantage comes with greater computational complexity in defining the projective geometry. So, there is a clear trade off between ease of computation and accuracy. We choose to represent the player as a single plane because of the simplicity of its approach.

Figure 46: players in both views with ground coordinates

The data known is just the 3D coordinates of the players (on ground, i.e. z coordinate=0) (Damien Delannay 2009). Using the camera equations from 2.2.3 to project from 3D to 2D, the coordinates of the players in the 2D image planes of the two cameras are obtained as shown in Figure 46.

The next step is to define planes around these players on 3D separately, so that homography can be applied between the two images.

4.2 Player planes

3D planes are defined for the players. The only detail known is the position of player on the ground (or the coordinates of the player's feet). The procedure followed is to visualize a player as being inside a 3D box or a cylinder (Figure 47). The width and height of this box is roughly assumed as the height and width of a normal human form. The height is assumed to be 2m or 200 cm and width to be around 0.6m or 60 cm approximately. A plane cutting this 3D box defines the player plane.

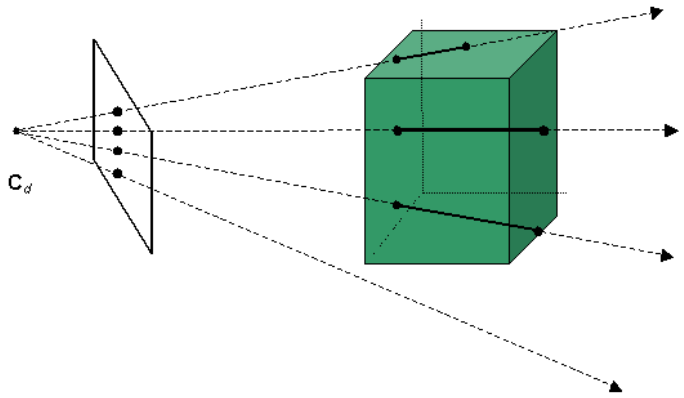


Figure 47: The player visualised inside a cylinder

Any plane can be defined by

$$ax + by + cz + d = 0$$

A plane equation can be deduced by knowing 3 points on the plane.

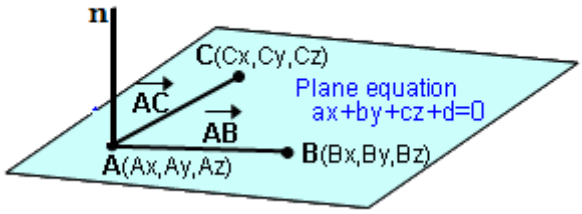
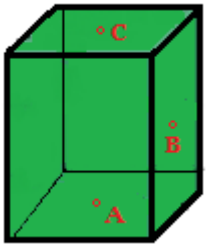


Figure 48: Plane equation from 3 points on a plane

The three points are chosen as follows:



A: the ground coordinates: feet of the player: (x,y,z=0)
C: the same coordinates as 1 but at the other end: head of the player: (x,y,z=2m)
B: a random point on either sides of the box.

Figure 49: Three points on the player box to form a plane

Knowing these 3 points, the normal n is estimated,

$$n = \overrightarrow{AB} \times \overrightarrow{AC} = (a, b, c)$$

Knowing the normal and any point on the plane (A), $d = -ax - by - cz$ is estimated and a plane is constructed. The 3D planes when projected onto the 2D image planes appear as shown in Figure 50.

Figure 50: The extreme points of the 3D player planes projected onto the image plane

4.2.1 Challenge: depth

To determine planes for each player, the depth information is most necessary. The depth information is the most important because it determines if the ray passing from the player plane to the first camera falls short in distance thus giving wrong projections in the image plane of the second camera.

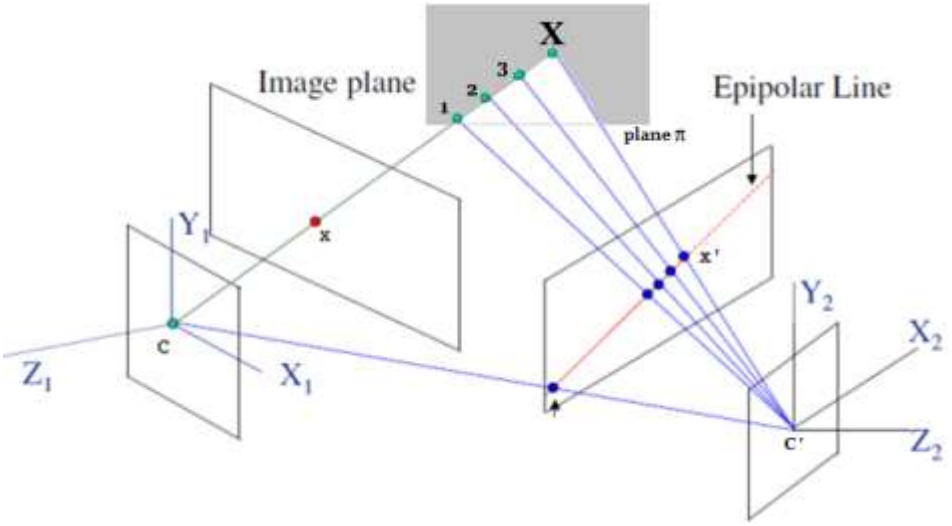


Figure 51: Correspondence between two views by plane induced homography

Epipolar geometry revisited: Consider Figure 51 where C and C' are the two cameras. X is a point in 3D space and x and x' are the image of this point in the two image planes respectively. To know the importance of depth, consider the points $1, 2, 3$. All these points lie on the ray joining the camera to the plane but do not originate from the plane. If X is the player in the 3D space and the plane π is the player plane selected, all the three points $(1, 2, 3)$ fall short in depth to meet the player plane. The point X on the player plane is the point of the perfect depth which gives the correct images x and x' in both the image planes and hence the correct correspondence between the two images.

The plane π is given by $\pi^T X = 0$;

$$\pi = [v\ 1];$$
$$v = [a\ b\ c\ ];$$

The line joining the camera center and any point in 3D space can be given by

$$C + (P_1^{-1}x - C)d$$

C: Camera center (first camera)
P_1: Projection matrix of first camera
d: Depth

At a proper depth, this ray originates from the point X. But this point X lies on the plane π and hence will satisfy the equation $\pi^T X = 0$. Solving this will give the depth information, using which the point X can be re projected back on to the second image plane.

$$\pi^T(C + (P_1^{-1}x - C)d) = 0$$

$$d = \frac{-\pi^T C}{\pi^T\ (P_1^{-1}x - C)}$$

The point x' in the second image plane can be given by,

$$x' = P_2 X$$

$$x' = P_2 \,.\, (C + (P_1^{-1} x - C) d)$$

P2: projection matrix of camera2

This point x' corresponds to the point x in the first image for the appropriate depth, which is calculated from the previous formula.

Consider player number1. The player is visualized to be inside a 3D box and a plane is inserted onto him through the box. The extreme points of the plane projected on to the 2D image plane of the first camera gives Figure 52.

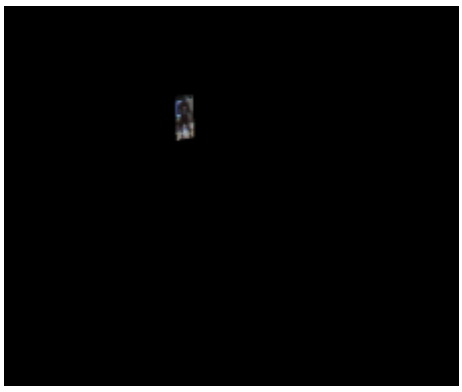


Figure 52: The plane of the player (in 3D) projected on to the camera's image plane

Homography is applied for this plane and the corresponding points of this player in the second view are determined. This can be seen in Figure 53, where the point on player1 selected in the first image projects on to a point on the player1 (almost at the same place) in the second image, thus validating the plane homography.

Figure 53 : Example of homography for player plane between two images

4.3 Plane Selection

The plane selection is an important task because the selection of the plane determines the accuracy of the model or representation of the player. Players are of different sizes and orientations on the field, but we approximate the player on to just a plane. So, according to the plane, the homography varies and the correspondences very. If the plane encloses the player perfectly, good correspondence will be observed in both views. But if the plane falls short, the points on the player that lie outisde the plane will be projected to wrong points (somewhere different from the player).

Let us define planes on the player and check the projections on different viewpoints. The viewpoint can be anywhere in between the two cameras. 9 different virtual viewpoints are considered in between two cameras.

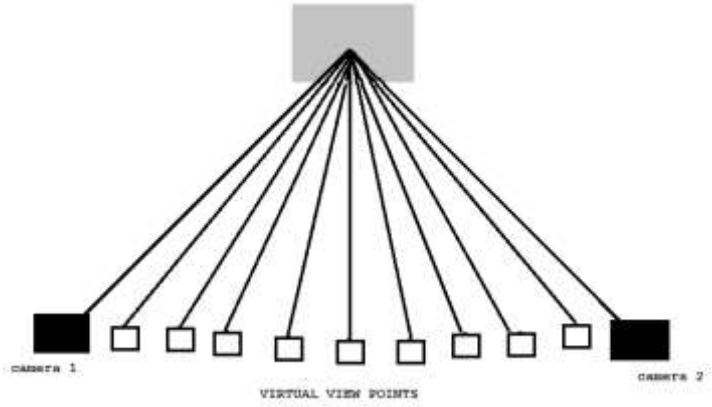


Figure 54: Different viewpoints in between two cameras

Consider a plane fixed at 0 degree angle for all the players and for all the viewpoints and the planar homography is applied for this plane to get a reconstruction at each viewpoint. The plane angles follow the reference shown in Figure 55.

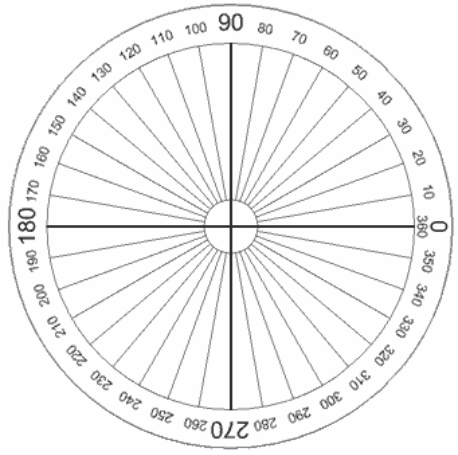


Figure 55: Angles of plane rotation

The plane, fixed at an angle 0 degrees proved lucky for the player selected, since the player orientation is almost straight, this plane works well for all the views.

Figure 56 shows the player projected from camera1 in all the views in between two cameras. But this is not always the case.

Figure 56: projection of player from camera1 in all the viewpoints with a 0 degree plane

If the plane was fixed at an angle 90 degrees, the projection in certain views (view no. 6 or 7 among the 9 viewpoints) is not perfectly visible (

Figure 57). So the plane also depends on the chosen viewpoint.

Figure 57: projection of player from camera1 in all the viewpoints with a 90 degree plane

So, selecting a good plane for each player is of paramount importance. The following two segments describe the two method of selecting the best possible plane.

- Approach 1: selecting the plane based on player orientation.
- Approach2: selecting planes depending on the virtual viewpoint orientation.

4.3.1 Best plane: approach 1

The plane is selected based on the player orientation. The plane is rotated around the 3D box (which encloses the player) to get planes at different angles (coping to the orientation of the player). The plane is varied from 0 to 360 degrees in step of 10 degrees and the best angle is chosen for the

chosen viewpoint in between the two cameras. An example of rotating the planes in the box enclosing the player can be visualised in Figure 58.

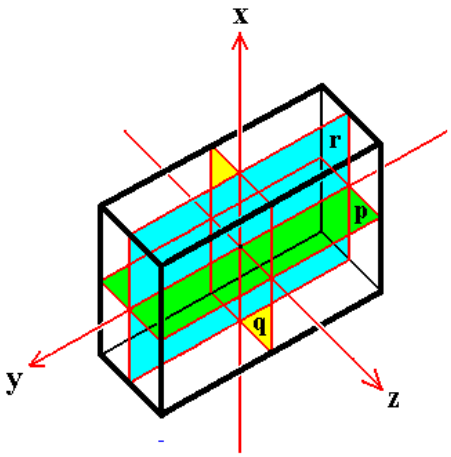


Figure 58: plane rotation inside a box

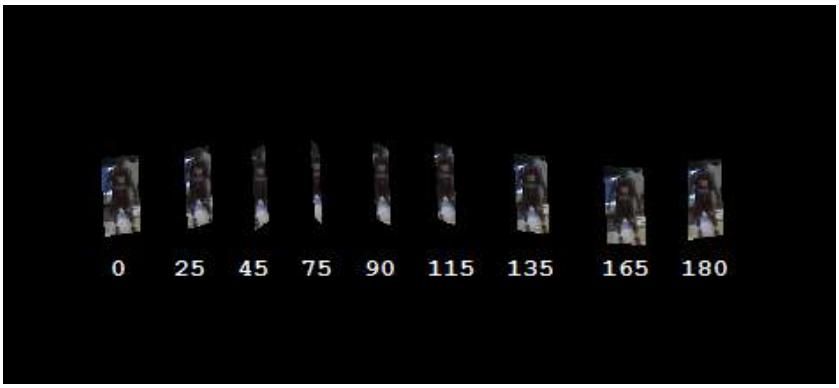


Figure 59: different planes for player

Figure 59 shows some angles of the player planes. For example, for this particular player, choosing a plane at 90 degrees is very bad. The player orientation is almost flat and hence a simple plane of 0 degrees could do the job.

Figure 60: different planes for player

While for the player in Figure 60, an angle of 45 degrees or 135 degrees is better for the orientation of the player.

How to choose the best plane?

So, the main question is how to determine which of these planes, is the best. The end result of the plane homography is to obtain best correspondences between the views. So the metric to select the plane is the difference between the correspondences of the two views.

Metric: For every plane (all the angles) homography is estimated and the corresponding points in the second view is obtained. These points are projected on to the first view. As the player in the first view and the projection of the same player from the second view should be the same, the difference between the two is used as a metric to select the best projection. The smallest difference indicates the best correspondence and hence that plane will be chosen as the best plane.

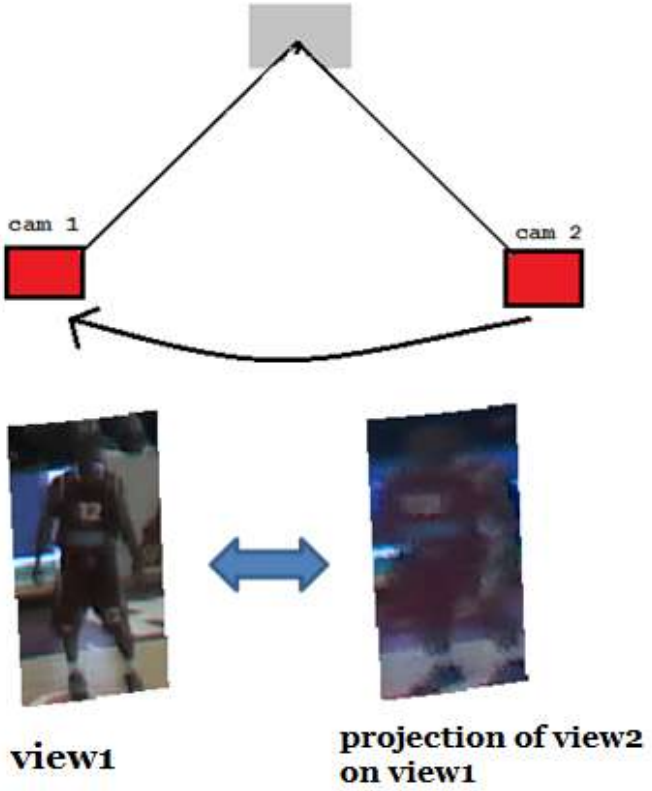


Figure 61: comparison between player in view1 and projection of view2 on view 1

$$\text{Metric} = (I1 - I2)^2$$

$I1$: Player in view1

$I2$: Corresponding player in view2 found by the plane induced homography

Note: Both $I1$ and $I2$ are normalized to the number of pixels such that the difference can be obtained.

In Figure 61, the two camera views are considered. The player in view1 is compared with the projection of player of view2 on view1. The difference between these images is calculated and the plane which gives the smallest difference is chosen as the best plane (best angle).

Player	1
Best plane angle(in degrees)	135

Table 6: the best plane (or angle) for the player for view1 (camera1)

These planes are best for view1, i.e. the camera1 view. Let us verify if the same angle/plane holds good for all the other viewpoints.

The midpoint between the two cameras is the chosen viewpoint (Figure 62). For a given plane, the player in view1 is projected on to this view. Similarly the corresponding player in view2 is projected on to the same view. These two projections are compared to obtain the best plane for this viewpoint

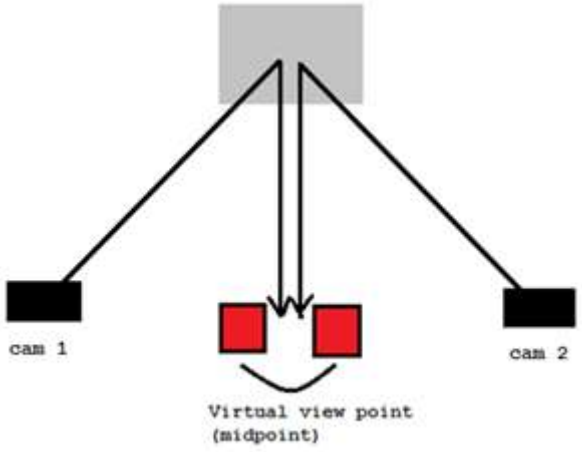


Figure 62: comparison of projections at a virtual viewpoint (midpoint between the cameras)

Player	1
Best plane angle(in degrees)	170

Table 7: Table 5: the best plane (or angle) for a player in virtual viewpoint

The best plane angle obtained for player1 for this viewpoint is different from the plane angle obtained previously where the chosen viewpoint was the camera1 view. (The difference is not that high because the player's orientation is almost straight, so both angles are near to the 0 degree orientation). The player planes may be also dependent on the viewpoint from where the player is chosen to be viewed than on the player orientation.

The next topic speaks about selecting a plane based on viewpoint orientation.

4.3.2 Best plane: approach 2

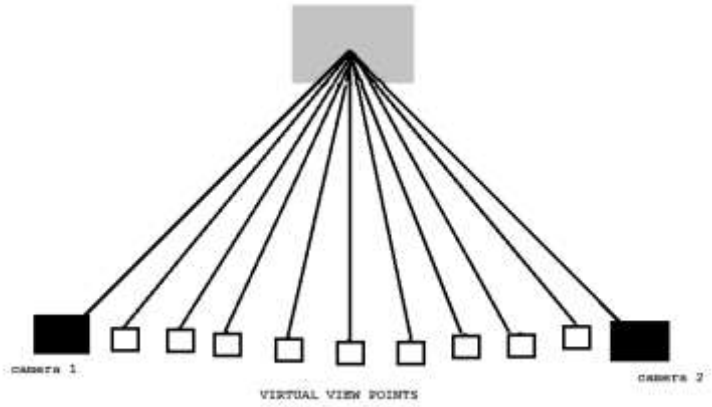


Figure 63: different viewpoints between the cameras

As observed previously, the viewpoint also plays a very large role in selecting the player plane. So, let us try to construct a plane based on the virtual viewpoint instead of the orientation of the player. Before, three points were selected on the player (according to a particular angle) and the plane was estimated. Here, a new method to select the player plane is explained.

New normal vector: Before the normal to the plane was calculated by choosing three points on the player. Now, in this method (approach 2), the normal to the plane is the ray that is passing through the camera centre.

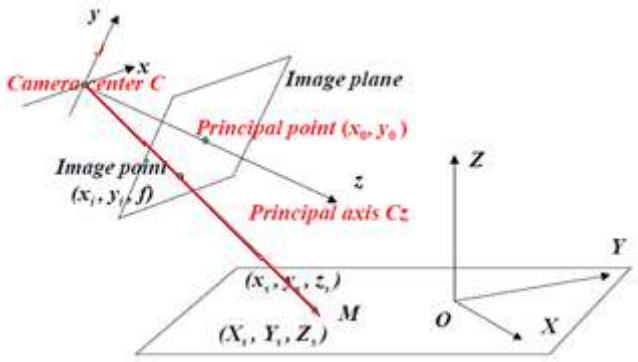


Figure 64: Camera coordinates and scene coordinates

The normal is the line from the centre of the camera (the red line in Figure 64). Two points on this line are the camera centre C and the point (x_s, y_s, z_s) in the 3D space.

- Camera centre : $C = [C_x, C_y, C_z]$
- from segment 2.2.2 in chapter 2, the second point is

$$C' = \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ O_3^T & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix}$$

As the normal follows the Z axis,

$$C' = \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ O_3^T & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

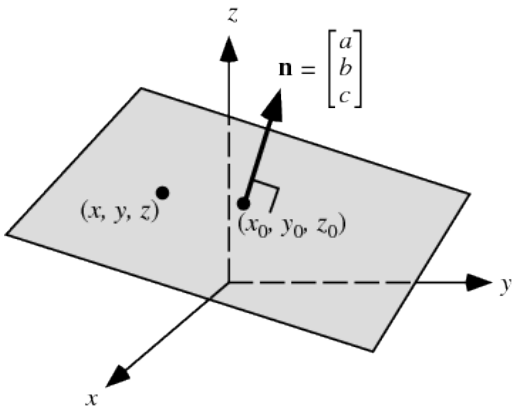
$$C' = \begin{bmatrix} [R]_{3x3} & C \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{n(normal)} = \boldsymbol{C} - \boldsymbol{C'}$$

The plane equation is built in the following way. The standard equation of a plane in 3D space is

$$ax + by + cz + d = 0$$

The normal to the plane is the vector $n = (a, b, c)$



The equation of a plane with nonzero normal vector $n = [a\ b\ c]^T$ and passing through the point r_0=$(x_0, y_{0,} z_0)$ is

$$n(r - r_0) = 0$$

where $r = (x, y, z)$

$$n = C - C' = [a; b; c];$$

Also, $r_0 = (x_0, y_0, z_0)$ is the coordinates of the player on ground (given).
Plugging in gives the general equation of a plane,

$$ax_0 + by_0 + cz_0 + d = 0$$

where, $d = -ax_0 - by_0 - cz_0$

Knowing the plane equation, the homography can be estimated as explained in 4.2.1.

Figure 65 and Figure 66 show the players projected from camera1 and camera2 to all the other viewpoints using this new way of plane selection.

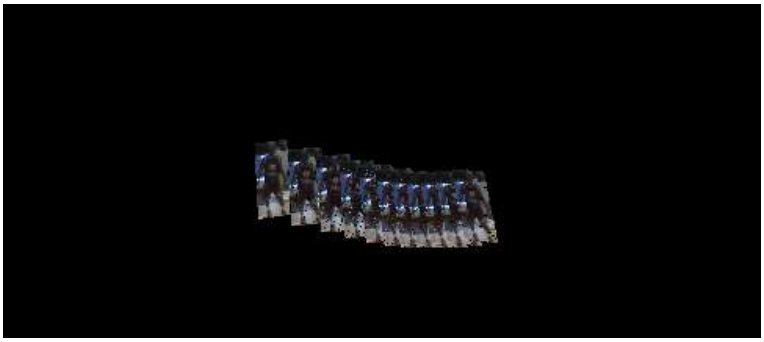


Figure 65: player1 projection from view1 (camera1) in all viewpoints

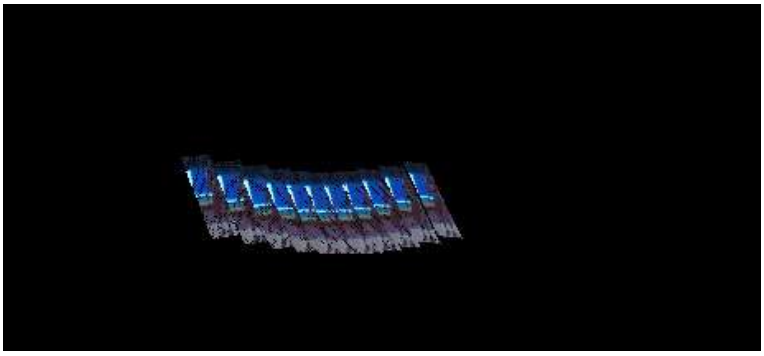


Figure 66: player1 projection from view2 (camera2) in all viewpoints

(The projections appear overlapped because all the nice projections are tried to obtain on a single figure).

So, two different approaches have been tried to obtain the best possible model for 3D modelling of the plane. Now we try to investigate the reconstruction results based on this best model. So let us verify the model in reconstructing the player. This is discussed in the next topic by introducing the probability maps.

4.4 Probability maps

The best planes have been selected for each player and this is assumed to be the best model to represent the player. So the correspondences found for a player with the best plane, must give a clear silhouette of the player (i.e. clear distinction between the player and the background). To obtain this silhouette (which can be visualized as the segmented version of a player), the correspondences between the views have to be compared. Depending on the plane selected on the player, the points projected could either lie on the player or in the background (behind the player). The decision has to be made by forming a certain probabilities within both views.

The 3D modelling we have implemented is a weak model with just a plane to represent the whole player. This model is verified by the probability map.

To reconstruct the player in the virtual viewpoint from the two projections (of the players) this probability map/decision map is very important. Consider a point projected onto the virtual viewpoint, this point can belong to the player/background, and hence relies on the probability map.

Camera1	Camera2	Probability(p)
Player	Player	1
Background	Background	0
Player	Background	0<p<1

Table 8: probability map (1)

PROBABILITY MAP

Probability	corresponding point
1	player
0	Background
0<probability< 1	either player/background (***Ambiguity***!)

Table 9: probability map (2)

Reconstruction is made in accordance with the model. Color is the most important information used to compare the correspondences and create a silhouette of the player. This was the reason why achieving color uniformity between images (color camera calibration) was of high priority.

The probability maps are created for the player in the next section.

4.4.1 Probability map estimation

A virtual viewpoint is chosen in between the two cameras and the best plane for this particular viewpoint is selected for each player. Homography is estimated for each plane and the players from both the camera images are projected on to the virtual viewpoint. These two projections are used to build the probability maps for each player.

Consider a single player as an example to obtain the probability map. (Figure 67)

Figure 67 : The best plane (projected on to 2D) for the player

A plane is selected for this player (approach1: plane based on player orientation). The projection of this player from view2 (camera2) is obtained. The player in view1 (camera1) and the projection from view 2 are seen in Figure 68. A probability map has to be devised based on these two images.

Figure 68 : player in view1 (camera1) and the projection of player from view2 (camera2) on view1

The comparison between these two images should almost provide the player and background mismatch between the views. The difference between them could be normalized to provide a gray scale map which acts as the probability map. Select a position; check the pixel at this point in both the views. If it lies on the player on both views, the difference is zero. The color uniformity was achieved using color calibration, so it is safe to assume, that the difference will be zero. So, the black regions (0) denote the player. But this color uniformity also proves a problem for the uniform background regions (background space is large), and some background pixels can be mistakenly assumed as foreground. For the rest of the regions, the difference is non zero because the point may lie on player in the first view and on the background in the second or vice versa. They can belong to either player or background. The gray level probability map is shown in Figure 69.

Figure 69: Probability map (1)

To see it more clearly, colormap is shifted (Figure 70).

Figure 70: colormap shifted probability map (1)

The difference values in the map can itself behave as probabilities. Pixels with lowest difference can be termed as players while the one with highest value can be of background but the decisions might be erroneous.

There are three ways of making wrong decisions,

1. The point on player can be chosen as background : because of our model selection
2. The point on player can be chosen as background : because of color mismatch
3. The point on background (somewhere around the player) is chosen as player: again because of color (uniform background problem)

The first error is inherent because of the selection of our model. The second error will not materialise if the camera is color calibrated properly.To eliminate the third error, some player contour or edge information is combined with color information from before method to obtain a better probability map. The background around the player is eliminated by some simple thresholding techniques.

Figure 71 shows the background of the two images from the two cameras. This is used to extract the players from the images. Appropriate color based thresholds are used to eliminate background as much as possible and to obtain the player silhouette.

Figure 71: Background from two views

But the background known is not perfect (it is obtained by averaging many frames) and hence the player extraction is not perfect. Figure 72 shows the regions of the player after eliminating the background around.

Figure 72: Segmentation mask applied on the corresponding region of player in both views

Comparing these, a mask can be generated, which gives the better representation of a player for reconstruction. The pixels belonging to the player in both images according to the plane selected are termed chosen as the player (red regions). Around the edges of the player, there is some ambiguity with in color and these can be either on player or background (green regions) (Figure 73). Some pixels inside the player are termed as background, and this is because of bad segmentation only.

Red : Player

Green : Player or background (Ambiguous)

Figure 73: probability map or mask (2)

The third kind of error mentioned previously, about considering background pixels as the player has been eliminated here. The segmentation of the player based on color is also not perfect and hence again around the edges of the player there is some ambiguity about the decision.

So, we have investigated the reconstruction based on our selected model and reconstruction based on color. The concern with color mismatch has been solved by our camera color calibration. Even though we try to find the best plane to model the player, there are still some concerns in the accurate reconstruction of the player. As a next step, to reconstruct the player in a chosen viewpoint, these probability map or mask can be applied to the player region in both the views and then project them on to the viewpoint. These two regions could be averaged to visualise the reconstructed player in a chosen viewpoint.

CONCLUSION

This work is an attempt towards virtual viewpoint synthesis in a multi view camera network. We try to generate an intermediate viewpoint image from two cameras, which capture a basketball match. A combination of a weak model based approach and the transfer based approach is used for view synthesis. A very simple model is used to generate the 3 dimensional objects in the scene (the players), namely a 'plane'. The players are modelled as planes and the color information is used to synthesize the texture of these 3D models. Transfer of correspondences between views is made, which used projective geometry between neighbouring views. The epipolar geometry is used to find dense correspondence between the views.

The best plane (3D model) for a player is the one that encloses the whole player in the best possible way. Homography, estimated to find correspondences between both camera views, depends on the plane selected. We performed a simple ground based homography to reconstruct the ground regions, where we observed the color dissimilarity between the images of two cameras. This was a serious challenge because there is high reliance on color while selecting the best model for the player. The best plane is chosen as the one which gives the best correspondence of the player between the views and the metric to decide is based on color.

So, before proceeding with player reconstruction, we did three color correction techniques. The first two are implemented on the images as a post processing step and hence adds some delay if the application is real time. The third method is the color calibrating the camera and is a pre processing step, because all the cameras are color calibrated before using them to take images. All the images obtained from these color calibrated cameras have color uniformity. The results are effective and it is also computationally very easy and efficient.

After color calibration, best plane for the player is selected first based on the orientation of the player and next based on the orientation of the viewpoint in between the cameras. Depending on this best 3D model chosen, the correspondences between the two views are compared to obtain a probability map which should give clear distinctions between the player present on the plane and the background, which will help reconstruct the player. The probability maps generated do give an approximate representation of the player but there are still some errors and hence does not provide us with a perfect silhouette of the player. These errors are because the assumption that our model is the best to represent the player accurately is not true. The errors can also be from color mismatch, but we have solved it by color calibration.

The virtual viewpoint synthesis does suffer through some challenges and this work paves way for some improvements. The model selected to represent the player is of the simplest type: a plane. Many criterions like the player orientation, size has to be enclosed within this simple model. Better models like a cylinder or multiple planes could be researched with. Visual hulls are an interesting technique in generating geometric entities and can prove as a better model for a scene. (Also, lack of good segmentation of the players from the image is an inherent disadvantage. If the perfect segmentation could be achieved, the reconstruction of the player in a virtual view could be calculated with a relative ease, without determining the probability maps.)

Researching with these new methodologies will help in synthesizing images in a more accurate manner in any viewpoint in between the cameras. This may lead to the creation of completely new and enjoyable ways to present or view entertainment and sporting events in the near future.

Bibliography

[1] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, T. Wiegand. “In-termediate view interpolation based on multiview video plus depth for advanced 3D video systems.” *15th IEEE International Conference on Image Processing (ICIP).* 2008.

[2] Bjorn Stabell, Ulf Stabell. *Duplicity Theory of Vision.* Cambridge University Press, 2009.

[3] C. L. Zitnick, S. Bing Kang, M. Uyttendaele, S. Winder, R. Szeliski. “High-quality video view interpolation using a layered representation.” *SIGGRAPH.* 2004.

[4] Damien Delannay, Nicolas Danhier and Christophe De Vleeschouwer. “Detection and recognition of sports(wo)men from multiple views.” *Third ACM International Conference on Distributed Smart Cameras (ICDSC).* 2009. 1-7.

[5] Dubrofsky, Elan. *Homography Estimation.* Carleton University, 2007.

[6] Dyer, S. M. Seitz and C. R. “View morphing.” *Proc. SIGGRAPH.* 1996.

[7] Fairchild, Mark D. “Color Appearance Models:CIECAM02 and Beyond.” *IS&T/SID 12th Color Imaging Conference.* Chichester, UK, 2005.

[8] Faugeras, Olivier. “Three-Dimensional Computer Vision.” MIT Press, 1993.

[9] Hanrahan, M. Levoy and P. “Right field rendering.” *Proc. SIGGRAPH.* 1996.

[10] Ilie, Adrian. “Ensuring Color Consistency across Multiple Cameras.” University of North Carolina at Chapel Hill, 2004.

[11] Saito, N. Inamoto and H. “Virtual Viewpoint Replay for a Soccer Match by View Interpolation From Multiple Cameras.” *IEEE TRANSACTIONS ON MULTIMEDIA.* 2007.

[12] Saito, S. Yaguchi and H. “Arbitrary viewpoint video synthesis from multiple uncalibrated cameras.” *IEEE Trans. on Systems, Man and Cybernetics.* 2004.

[13] Shashua, S. Avidan and A. “Novel view synthesis by cascading trilinear tensors.” *IEEE Trans. Visualiz. Comput. Graph.* 1998.

[14] Simone Bianco, Francesca Gasparini and Raimondo Schettini. “Combining Strategies for White Balance.” Universit`a degli Studi di Milano-Bicocca, 2007.

[15] T. Kanade, P. W. Rander, and P. J. Narayanan. “Virtualized reality:Constructing virtual worlds from real scenes.” *IEEE Multimedia.* 1997.

[16] Tsai, R. Y. “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-Shelf TV cameras and.” *IEEE J. Robot. Autom.* 1987.

[17] Ulrich Fecker, Marcus Barkowsky, and Andre Kaup. “Improving the Prediction Efficiency for Multi-View Video Coding Using Histogram Matching.” *Chair of Multimedia Communications and Signal Processing.* University of Erlangen-Nuremberg, 2006.

[18] Williams, S.E. Chen and L. “View interpolation for image synthesis.” *Proceedings of SIGGRAPH ’93.* 1993.

[19] ZHANG, ZHENGYOU. “Determining the Epipolar Geometry and its Uncertainty:.” *International Journal of Computer Vision*, 1996.

[20] Zisserman, Richard Hartley and Andrew. *Multiview Geometry in Computer Vision.* Cambridge University Press, 2000.