# Thesis or Article

JeffGWood@mavs.uta.edu

June 20, 2016

# Contents


**1 Introduction** **2**

**2 Backround** **4**

**3 Point Interpolation** **6**

**4 Image segmentation** **8**

**5 Process** **9**

# Chapter 1

# Introduction

Through the development of applications such as augmented and virtual reality, object / scene reconstruction and visual effects, the process of generating images from an arbitrary vantage point can be found in a variety of applications. In this Thesis (or Article) I will discuss various methods for Image Creation from an arbitraryvantage point, which can be accomplished by two main methodologies of Geometric Construction and Image Synthesis. While both methods use stereo correspondance of multiple images, they differ in the way information is stored and used.

Geometric Construction (GC) contains information about the real-world spatial properties (Coordinates in space, Color), thus viewing results are non-constrained in vantage point. Image Synthesis (IS) relies on image properties (pixel displacement) and is thus viewing results are imited in the possible vantage points.

## Symbols and Notation

| Symbol | Description |
|---|---|
| $\mathbf{v}$ | *Vectors* in *lowercase* bold |
| $\mathbf{M}$ | *Matrices* in *uppercase* bold |
| $\mathbf{u}$ | Image coordinate |
| $\tilde{\mathbf{u}}$ | Image coordinate (expressed *homogeneously*) |
| $\mathbf{x}$ | Spatial coordinate |
| $\tilde{\mathbf{x}}$ | Spatial coordinate (expressed *homogeneously*) |
| ${}^{A}\mathbf{x}$ | Spatial coordinate in reference frame $A$ |
| ${}^{A}\tilde{\mathbf{x}}$ | Spatial coordinate (expressed *homogeneously*) in reference frame A |
| ${}^{C}_{B}\tilde{\mathbf{M}}$ | Change from of reference frame $B$ to reference frame $C$ |
| $s$ | Normalizing factor applied to *homogeneous* vector so last element becomes equal to 1 |
| ${}^{D}\mathbb{S}$ | Spatial reference frame $D$ |
| $[\mathbf{x}]_{\times}$ | Skew-symmetric matrix version of vector $\mathbf{x}$ used as *left*-operand in the *cross*-product such that $[\mathbf{x}]_{\times} \cdot \mathbf{y} = \mathbf{x} \times \mathbf{y}$ |
| $l$ | Epipolar line |

# Chapter 2

# Backround

Oridinarily, real-world data contains 3-dimensions. Because standard images only include 2-dimensional data, information regarding depth is lost (i.e. it is often difficult to judge distance from a single image without visual cues). *Stereovision* attempts to resolved this by finding the same point in both *stereoscopic* images (known as a *corresponding point*), and recovering the depth information. An elementry example of this occurs in stereoscopic images with relatively low distance between cameras (i.e they are righht next to each other). Objects that are *farther* away from the observer occur closer together in the stereo images, whereas objects *closer* to the camera appear appear farther appart in the stereo-images.

### Change of Reference

Each view from a pair of stereo-images encompasses its own *frame of reference* (i.e. the directions of *forward* or *backward* are unique to image and may differe considerably depending on camera displacement). This requires expressing points from different frames of reference (traditionally referred to *left* and *right*) in a single reference frame. As such it is necessary to be able to express coordinates in a given reference frame in any other reference frame.

Coordinates given in ${}^{A}\mathbf{x}$ can be expressed in ${}^{B}\mathbf{x}$ by the geometric transformation:

$${}^{B}\mathbf{x} = {}^{B}_{A}\mathbf{R} \cdot {}^{A}\mathbf{x} + {}^{B}_{A}\mathbf{t}$$

or

$${}^{B}\tilde{\mathbf{x}} = \left[\begin{array}{c|c} {}^{B}_{A}\mathbf{R} & {}^{B}_{A}\mathbf{t} \\ \hline 0 & 1 \end{array}\right] \cdot {}^{A}\tilde{\mathbf{x}}$$

$$= {}^{B}_{A}\mathbf{M} \cdot {}^{A}\tilde{\mathbf{x}}$$

where ${}^{B}_{A}\mathbf{M}$ is also the geometric transformation necessary to transform ${}^{B}\mathbb{S}$ into ${}^{A}\mathbb{S}$.

Withough calculating any new quantities, rearranging allows us to express coordinates in ${}^{B}\mathbf{x}$ in the ${}^{A}\mathbf{x}$ reference frame as:

$${}^{B}_{A}\mathbf{R}^{\intercal} \cdot ({}^{B}\mathbf{x} - {}^{B}_{A}\mathbf{t}) = {}^{A}\mathbf{x}$$

and similarly transforms ${}^{A}\mathbb{S}$ into ${}^{B}\mathbb{S}$.

### Epipolar constraint

Each point of of interest (also referred to as a *feature*) in a single image occurs in a 2-dimensional space at location $\tilde{\mathbf{x}} = [x, y, 1]^{\intercal}$. Unless the position in 3-dimensional space, or the corrsponding location of

$\tilde{\mathbf{x}}' = [x', y', 1]^\intercal$ in an image viewed from a different angle is known, depth information is lost. The most that can be determined from the 2-dimensional information is the *line of sight*, or *the region in 3-dimensions space the point can exist while still appearing as the same point in the original image.* From a mathematical context, this set of infinitley many points form a 1-dimensional subspace of the 3-dimensional space that makes up the physical world around us.

When viewed in the original image, this set of points overlaps and appear as a single point consistant with the original point at location $\tilde{\mathbf{x}} = [x, y, 1]^\intercal$. When viewed in an image from a differing angle, this set of points forms a line extending the boundaries of the image. Known as the *epipolar line*, the line has a row-vector form of $\mathbf{l}' = [A', B', C']$. The corresponding point of $\tilde{\mathbf{x}}' = [x', y', 1]^\intercal$ is limited in location to this epipolar line, and is thus constrained by substituting $\tilde{\mathbf{x}}'$ into the linear form of $\mathbf{l}'$. The resulting equation of $\mathbf{l}' \cdot \tilde{\mathbf{x}}' = A' \cdot x' + B' \cdot y' + C' \cdot z' = 0$, is commonly referred to as the *epipolar constraint.*

### Fundamental Matrix

### Intrinsic Calibration Matrix

### Essential Matrix

When coordinates from a reference frame are expressed as *normalized image coordinates* the range of possible NIC values in the corresponding image are given by the

# Chapter 3

# Point Interpolation

Pixels from image $a$ and image $b$ can be used to create a new images. This is done by interpolating the pixel positions ($\mathbf{p}^a_{uv}$ and $\mathbf{p}^b_{uv}$) of corresponding points between frames. Because not all pixels are established as corresponding points, pixel correspondances *between* corresponding points ($\mathbf{p}_{uv}$) are calculated through bi-linear interpolation of 4 established corresponding points:

$$\mathbf{P}_{uv} = \mathbf{P}_{00} \cdot (1-u) \cdot (1-v) + \mathbf{P}_{10} \cdot u \cdot (1-v) + \mathbf{P}_{01} \cdot (1-u) \cdot v + \mathbf{P}_{11} \cdot u \cdot v$$

This is done through the following series of linear equations

$$x_{uv} = \begin{bmatrix} u & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ 1 \end{bmatrix}$$

$$y_{uv} = \begin{bmatrix} u & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ 1 \end{bmatrix}$$

or as a single matrix equation of

$$\begin{bmatrix} x_{uv} & 0 \\ 0 & y_{uv} \end{bmatrix} = \begin{bmatrix} \mathbf{u} & \mathbf{0} \\ \mathbf{0} & \mathbf{u} \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}^T \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{v} & \mathbf{0} \\ \mathbf{0} & \mathbf{v} \end{bmatrix}$$

where

$$\mathbf{u} = \begin{bmatrix} u \\ 1 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} v \\ 1 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_{00} & y_{01} \\ y_{10} & y_{11} \end{bmatrix}, \text{and } \mathbf{M} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

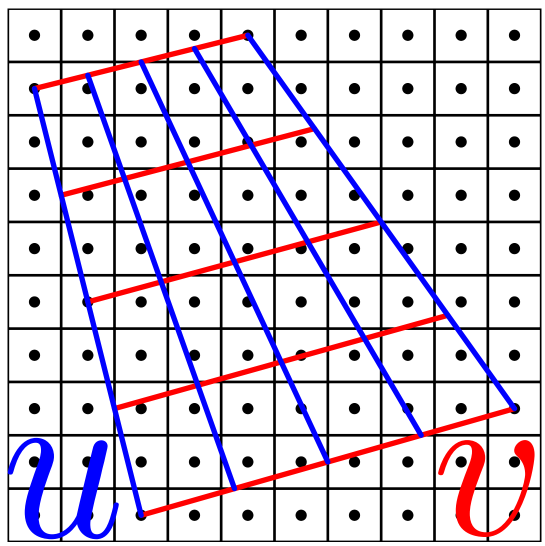


Figure 3.1: Bi-Linear Point Correspondance

# Chapter 4

# Image segmentation

Central to our need to localize corresponding points in stereo images is the ability to partition images by similar texture or planar attributes compared to the image at hole. Such techniques are referred to as *image segmentation*.

Image segmentation of regions of similar color or textures region is often approached from a graph-theory standpoint, in which individual pixels form the nodes of the graph. Edges are formed by a number of methods, the simplest of which is for each pixel to have 4 equally weighted edges connecting with the 4 immediate adjoining pixels in *North*, *East*, *South* and *West* vicinities (referred to as the **4-neighborhood region**). A common variation of this is to *also* include the next 4 closest adjoining pixels in the *Northeast*, *Southeast*, *Southwest* and *Northwest* vicinities (referred to as the **8-neighborhood region**). More sophisticated methods assign edge weightings proportional to the difference in color values (*scalar gray values* or *euclidean distance of color vectors*) between each pixel-pair.

Binary segmentation (partitioning into two regions) can be accomplished through min-cut / max-flow algorithms

# Chapter 5

# Process

The system in question contains 3 main components

1. Image Acquisition System
   - Webcam / Kinect set-up
   - If Webcam should also contain Image-Processing module for:
     - Feature Identification
     - Point-correspondance
     - Sub-Pixel interpolation
2. Point Cloud Processing
   - Should take inputs
   - Should produce point-clouds as one of the output
   - (Possible) Options for Surface Reconstruction include:
     - Calculation of surface Normal through PCA
     - Mesh construction through Delaunay trianglulation
     - Parametrization of Bezier surface through linear-least squares.