© 2017 IJSRSET | Volume 3 | Issue 5| Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Science and Technology

Stereoscopy - A Visual System

Dilipkumar N. Padhiar
Computer Science Department , M.B.Patel Science College, Anand, Gujarat, India

ABSTRACT

Stereopsis is a process in visual perception leading to the perception of the depth or distance of objects [1]. With the technological advance, two-dimensional images of the three-dimensional world became no longer sufficient. The visual system and its remarkable power to give an immediate perception of depth by reconstructing a three dimensional world from a two dimensional workion of that world, is what the science is trying to develop. The eye is the center of the visual system, it is a complex biological device, which takes visible light and converts it into a stream of information (color, shape), that can be transmitted via nerves to the brain [1]. The information captured by the eye is a 2D reconstruction from what is in its sight. But with only one eye, important information is missing; the presence of two slightly different positions of the eyes in the visual system is to acquire the needed information to make it possible for the brain to recover the depth from the information stream captured by each eye.

Keywords: Two-Dimensional and Three-Dimensional World, CCD, JMF, ImageJ, Java Advanced Imaging, DICOM, FITS, TIFF, GIF, JPEG, BMP, Correlation Based Matching, Stereo Matching, Motion Detection

I. INTRODUCTION

The goal of this work is to implement a software program capable of reconstructing a 3D scene from 2D pictures taken from slightly different positions, which would determine the depth of any point in the picture.

The reconstruction process undergoes many different tasks, such as camera calibration, which is extracting the camera parameters, and moving forward to try to reduce the computational effort using rectification.

The next important task will be to match the two points in the images that correspond to the same 3D point, and finally the reconstruction process by extracting the position in 3D coordinates of the points in the images

The first step is the capture of a scene from the 3D world to a 2D image. This was invented during the 16th century, but in 1969 Willard Boyle and George Smith invented the first CCD (charge-coupled device) camera which contains a grid of CCD image sensors [2]. The sensors captures and converts the light intensity into electrons, reads the value of each cell in the image, then turns each pixel into a digital value by measuring the amount of charge at each photosite and converts that measurement into binary form. That is what we call today digital photography.

Since photography is converting a 3D scene into a 2D image, the way this transformation is done, and its parameters are of a great interest. Since those parameters change from a device to another, a part of this student thesis is to extract those parameters from each camera.

II. METHODS AND MATERIAL

For a task like stereovision, the programming language must be powerful and user friendly at the same time. The programming language has to have a strong support for various image manipulations. Java was designed to make it much easier to write bug free code, thanks to its automatic memory allocation and

deallocation. Java has also a huge library which expands continuously to make it easier for the programmer to change the library code depending on the needs. Java has also the advantage of being platform independent.

2.1 Java Media Framework (JMF):

In 1997, Sun Microsystems, in co-operation with Intel and silicone Graphics, launched the version 1.0 of the Java Media Framework (JMF). The objective thereby was to create an API for the integration of time-based media in Java applet and applications that allow us the selection, processing and presentation of Multimedia. At the beginning, none of the three offered implementations could however become generally accepted, because of the strongly limited function range. In 1998, the alliance was broken with the resignation of Intel. In 1999, Sun, in a partnership with IBM, published the 2.0 version of the JMF and for the first time video capturing was made possible [3].

JMF has the same features that can be found in a regular Multimedia application: picture and tone are captured with a video camera and stored on a video Tape. The Stored data can be played again by inserting a video tape in the VCR which sends appropriate signals to television and speakers.

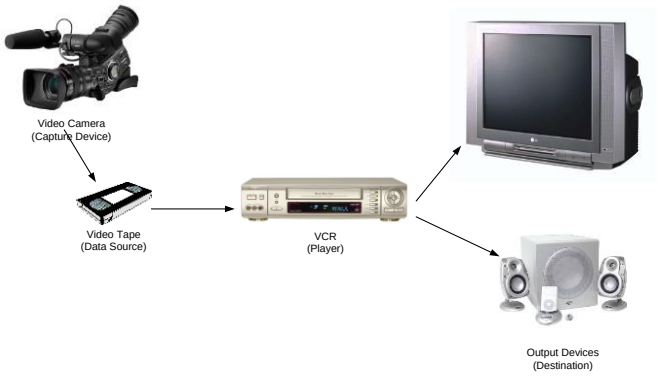


Figure Error! No text of specified style in document.**-1: The JMF Model**

JMF uses this same basic model [4]: Capturing audio and video with JMF requires the appropriate input devices such as microphones and cameras.

A *data source* encapsulates the media stream much like a video tape and a *player* provides processing and control mechanisms similar to a VCR, Playing the multimedia file requires then, the appropriate output devices such as speakers, and monitors. [4]

2.2 Java Advanced Imaging:

The Java Advanced Imaging (JAI) allows high performance image processing functionality to be incorporated into Java applications. JAI is simple to use, which reduces the time spent to develop an application, it has also the advantage of being built on the network-centric Java Platform, so that users can easily build applications for high-end image processing and visualization over the network. In addition, JAI supports most of the image formats, which makes it less complicated for the programmer [5].

2.2 ImageJ

ImageJ is a public domain Java image processing program inspired by National Institut of Health (NIH) Image for the Macintosh [1]. ImageJ is a powerful tool which can display, edit, analyze, process, save and print 8-bit, 16-bit and 32-bit images. It can read many image formats including TIFF, GIF, JPEG, BMP, DICOM, FITS and "raw". It can also calculate area and pixel value statistics of user-defined selections. It can measure distances and angles, create density histograms and line profile plots. It further supports standard image processing functions such as contrast manipulation, sharpening, smoothing, edge detection and median filtering. It does geometric transformations such as scaling, rotation and flips whereby Image can be zoomed up to 32:1 and down to 1:32. All analysis and processing functions are available at any magnification factor. The program supports any number of windows (images) simultaneously, limited only by available memory.
Spatial calibration is also available to provide real world dimensional measurements in units such as

millimeters. Density or gray scale calibration is also available. ImageJ was designed with an open architecture that provides extensibility via Java plug-ins. Custom acquisition, analysis and processing plug-ins can be developed using ImageJ's built in editor and Java compiler.

ImageJ is being developed on Mac OS X using its built in editor and Java compiler, in addition the BBEdit editor and the Ant build tool. The source code is freely available. The author, Wayne Rasband (wayne@codon.nih.gov), is at the Research Services Branch, National Institute of Mental Health, Bethesda, Maryland, USA.

2.2 Jama

In a CCD camera, the physical image plane is the CCD array, an N x M rectangular grid of photo sensors, each of which sensitive to light intensity. Each photo sensor generates a continuous electric signal. The video signal is sent to an electronic device called frame grabber, where it is digitized into a 2D rectangular array of N x M integer values and stored in a Memory buffer. At this stage, the image can be represented by an N x M matrix, whose entries are called pixels. The aim of image processing is actually the manipulation of those matrix elements, therefore the use of JAMA, which is one of the mathematical tools designed for java.

JAMA provides user-level classes for constructing and manipulating real, dense matrices. It is meant to provide sufficient functionality for routine problems. It is intended to serve as the standard matrix class for Java. JAMA was developed by Mathworks and NIST (National Institute of Standards and Technology and Technology)[6]

III. PINHOLE CAMERA GEOMETRY

The most common geometric model of an intensity camera is the perspective or pinhole model.

The pinhole camera is the simplest and ideal, model of camera function. It is a camera without a lens, the light producing the image passes through a small hole

It consists of the image plane I and a 3D point O the origin of the 3D camera reference frame, the distance between O and I is called the *focal length* and the line through O and perpendicular to I is the optical axis.

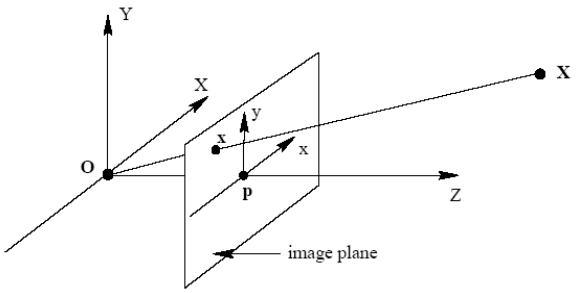


Figure Error! No text of specified style in document.**-2:** The perspective camera model

The principle of a camera is then to work a 3D point M from the camera reference frame into the 2D image I, we consider that the image of M $(X_{camera}, Y_{camera}, Z_{camera})$ in I is $m(x, y)$.

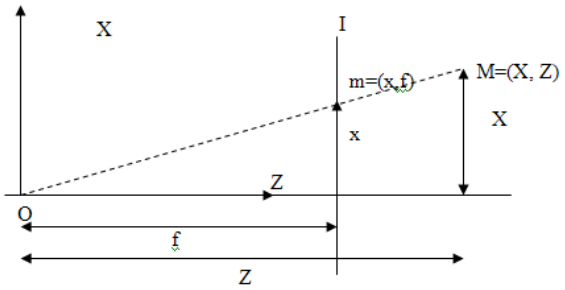


Figure Error! No text of specified style in document.**-3:** The perspective camera model restricted to the X, Z axis

The figure above shows the geometric relation between any 3D point in the camera reference frame and its 2D image in the image plane, we have:

Equation Error! No text of specified style in document.**-1**

$\frac{x}{f} = \frac{X}{Z}$ in the (X, Z) plane

$\frac{y}{f} = \frac{Y}{Z}$ in the (Y, Z) plane

So that any point M with the world coordinates $(X_{camera}, Y_{camera}, Z_{camera})^T$ is mapped to the 2D point m $\left(f \frac{X_{camera}}{Z_{camera}}, f \frac{Y_{camera}}{Z_{camera}}, f \right)^T$ on the image plane.

Mathematically the point in the space or its image in the image plane is described by homogeneous vectors, and the workion can be expressed in terms of a matrix multiplication.

Equation Error! No text of specified style in document.**-2**

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

From homogenous to non homogenous coordinates, we divide the first two coordinates by the 3rd one.

Equation Error! No text of specified style in document.**-3**

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \rightarrow \frac{1}{z} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

We must now link the camera reference frame with the coordinates of an image point in pixel units [7]:

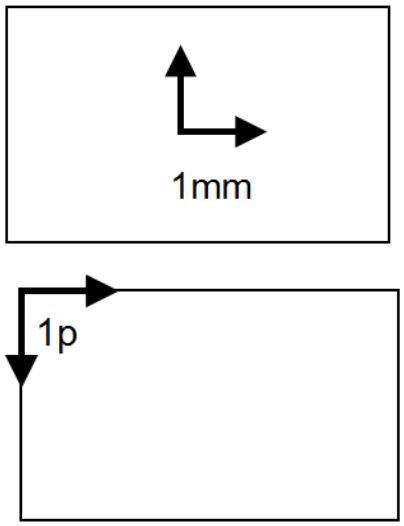


Figure Error! No text of specified style in document.**-4:** Image Coordinates and pixel Coordinates

Equation Error! No text of specified style in document.**-4**

$$x = -(x_{pix} - o_x)S_x = f\frac{X}{Z} \quad \text{and}$$

$$y = -(y_{pix} - o_y)S_y = f\frac{Y}{Z}$$

with (o_x, o_y) the coordinates in pixel of the image center, and (S_x, S_y) the effective size of the pixel in millimeters in the horizontal and vertical direction respectively.

From Equation 3-4 we have:

Equation Error! No text of specified style in document.**-5**

$$x_{pix} = \left(-\frac{f}{S_x} + o_x \right)\frac{X}{Z} \quad \text{and}$$

$$y_{pix} = \left(-\frac{f}{S_y} + o_y \right)\frac{Y}{Z}$$

Combining Equation 3-4 and 3-2 we obtain:

Equation Error! No text of specified style in document.**-6**

$$\begin{pmatrix} x_{pix} \\ y_{pix} \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{f}{S_x} & 0 & o_x \\ 0 & -\frac{f}{S_x} & o_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

But in practice, we usually express a 3D point in a fixed 3D coordinates system.

That means for a point M, its coordinates in the camera reference frame $(X_{camera}, Y_{camera}, Z_{camera})$ and coordinates in the world coordinates $(X_{world}, Y_{world}, Z_{world})$ can be described with the geometry of the Figure below:

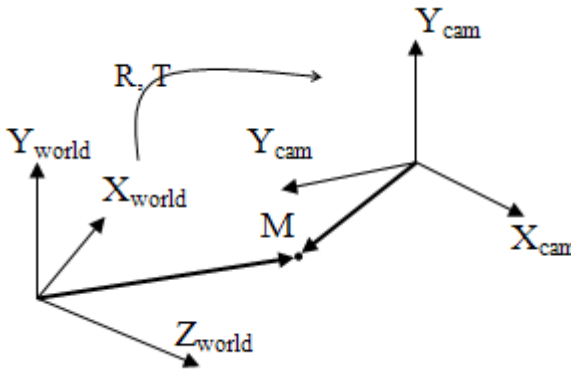


Figure Error! No text of specified style in document.**-5:** The relation between camera and world coordinate frames

Mathematically this geometrical transformation can be expressed using a transformation Matrix D so that:

$$M_{camera} = D \cdot M_{world}$$

$$\begin{pmatrix} X_{camera} \\ Y_{camera} \\ Z_{camera} \\ 1 \end{pmatrix} = \begin{pmatrix} R_{3x3} & t_{3x1} \\ 0\,0\,0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{pmatrix} \quad (3.5)$$

The image m in I of M in the world coordinates system, is then described with m = PM where m and M are represented with a vector and the used perspective transformation with the matrix P, where P is:

$$P = \begin{pmatrix} -\frac{f}{S_x} & 0 & o_x & 0 \\ 0 & -\frac{f}{S_x} & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

But in practical applications, the image coordinate system in which we address the pixels in an image is decided by the camera sensing array and is usually not the same as the ideal image coordinate system.

- The origin o of the actual image plane generally does not coincide with the camera reference frame origin O, because of possible misalignment of the sensing array.
- Determined by the sampling rate of the image acquisition devices, the scale factors of the image coordinates axes are not necessarily equal.
- Additionally the two axes image may not form a right angle as a result of the lens distortion.

The following transformation is used to handle these effects instead of (3.3).

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -f_x & r & x_0 & 0 \\ 0 & -f_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad (3.7)$$

The meaning of the used parameters is as follow

- f_x and f_y are the horizontal and vertical scale factors $f_x = \frac{f}{S_x}$ and $f_y = \frac{f}{S_y}$
- x_0 and y_0 are the coordinates of the center of the workion in the image
- r is the factor of distortion of the image, if the image is a rectangle, r = 0

IV. RESULTS AND DISCUSSION

Stereo Matching

Two eyes or cameras receive slightly different views of the three-dimensional world; the difference of the position in both vertical and horizontal directions of

the correspondent points in the stereo pictures represents the *Disparity map*.

There are two approaches of the correspondence problem:
Correlation Based Methods and Feature Based.
In this work the Correlation based method is chosen for its easier implementation and its dense disparity map, but this method is inadequate for matching image pairs taken from very different viewpoints.

Correlation Based Matching

In correlation methods the elements to match are image windows of fixed size, and the similarity criterion is a measure of the correlation between windows in the two images.

The corresponding element is given by the window which maximizes the similarity criterion within a search region.

Correlation Based Matching algorithm [Trucco, Verri, Fusiello]:

The input is a stereo pair of images, I_l and I_r.
Let p_l and p_r be pixels in the left and right images, 2W+1 the width in pixels of the window, $R(p_l)$ is the search region in the right image associated with p_l, and $\psi(u,v)$ a function of two pixel values, u, v.

For each pixel $p_l = [i, j]^T$ of the left image:

1. for each displacement $d = [d_1, d_2]^T \in R(p_l)$ compute

$$c(d) = \sum_{k=-W}^{W} \sum_{l=-W}^{W} \psi\left(I_l(i+k, j+l), I_r(i+k-d_1, j+l-\right.$$

2. the disparity of p_l is the vector $\bar{d} = \left[\bar{d}_1, \bar{d}_2\right]^T$ that maximizes c(d) over $R(p_l)$: $\bar{d} = \arg\max_{d \in \Re} \{c(d)\}$

The ouput is an array of disparities, one per each pixel of I_l;

with $\psi(u,v) = -(u-v)^2$ which performs the so called SSD (sum of squared differences) or block matching.

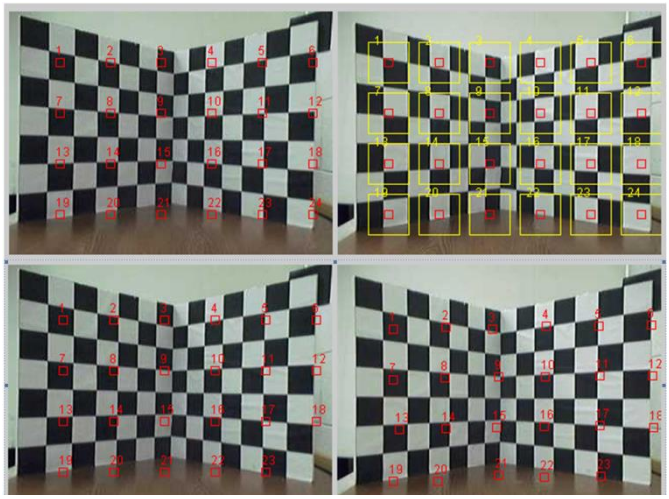


Figure Error! No text of specified style in document.**-6:** Search Area and Block Matching

Motion detection:

The same algorithm can be used to detect motion in a video sequence.
Two successive images in a video are very similar so that the position of an object in a picture p_1 is very close to the position of the same object in the following picture p_2.

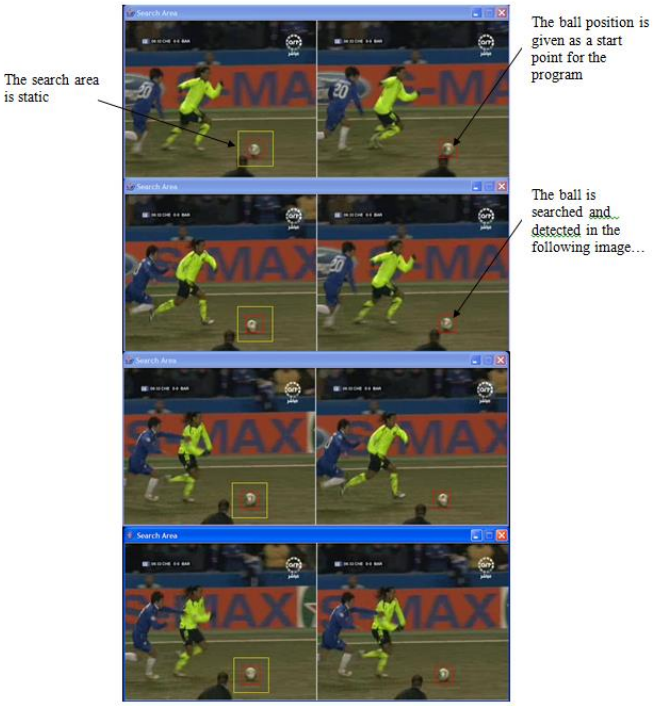
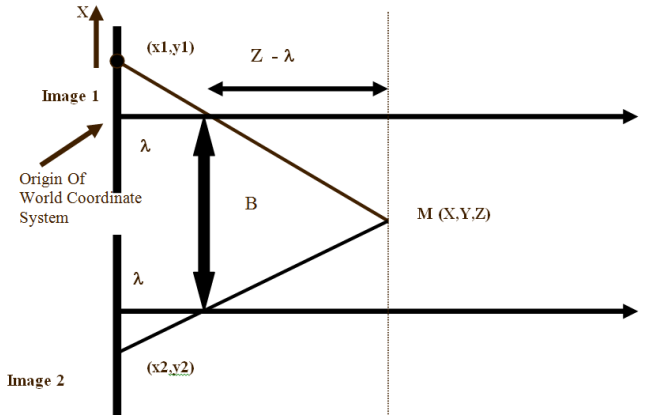


Figure Error! No text of specified style in document.**-7:** Motion Detection

3D Reconstruction

After solving the matching issue, we will discuss the last step of the 3D Reconstruction.
The reconstruction strategies depend on the amount of a priori knowledge available on parameters of the stereo system.
One strategy is based on the camera calibration, which usually assumes that the camera geometry is known. The other is called uncalibrated stereo analysis, which reconstructs the perspective 3D scene without knowing the camera positions.



The diagram above shows the top view of a stereo system composed of two pinhole cameras.

The left and right image planes are coplanar.
The way in which stereo determines the position in space of M is triangulation, that is, by intersecting the rays defined by the centers of projection and the images of M.

Triangulation depends crucially on the correspondence problem. Triangulation is the simplest case, but only applicable if the extrinsic and intrinsic parameters of the stereo system are known.

V. CONCLUSION

The purpose of this work was to implement a Software program, capable to reconstruct a 3D scene using two low resolution webcams, "Logitech Pro 4000". Java was the ideal tool, because of the huge library which expands continuously, the different libraries used in this work, were very efficient, and helped reducing the complexity of the programming part.

The Java programming software used in this work, was "Netbeans", this program is free, and easy to use. The results are not 100% accurate, because of the different approximations, but the goal of this work was successefully reached.

VI. REFERENCES

[1]. http://www.wikipedia.org
[2]. Optical Detectors for Astronomy II: Paola Amico, James W Beletic "The Invention and Early History of the CCD". ISBN 0792365364 – Springer 2000
[3]. Java Media APIs: Dr. Alejandro Terrazas, Dr. John Ostuni, Dr.Michael Barlow ISBN 0672320940 - Sams Publishing 2002
[4]. Java Media Framework API Guide: 1998-99 Sun Microsystems http://java.sun.com/products/java-media/jmf/2.1.1/guide/
[5]. Java Avanced Imaging http://java.sun.com/products/java-media/jai/whatis.html
[6]. JAMA: A Java Matrix Package http://math.nist.gov/javanumerics/jama/
[7]. Introductory Techniques for 3-D Computer Vision: E. Trucco, A. Verri ISBN 0132611082 1998 Prentice-Hall
[8]. Digitale Bildverarbeitung - Eine Einfuehrung mit Java und ImageJ : W. Burger, M. J. Burge Springer-Verlag 2005
[9]. 3D Video Communication: Oliver Schreer, Peter Kauff, Thomas Sikora ISBN 047002271 Wiley 2005
[10]. Handbook of Image and Video Processing: Al Bovik University of Texas at Austin ISBN 0121197905 Academic Press 2000