Abstract.- This paper presents a comprehensive study of all the challenges associated with the design of a platform for the reconstruction of 3-D objects from stereo images. The overview covers: Design and Calibration of a physical setup, Rectification and disparity computation for stereo pairs, and construction and rendering of meshes in 3-D for visualization. We outline some challenges and research in each of these areas, and present some results using some basic algorithm and a system build with off-the-shelf components.

Index Terms.- Epipolar geometry, Rectification, Disparity calculation, Triangulation, 3-D reconstruction.

1 Introduction

To get the impression of depth in a painting some artists of the Renaissance such as Filippo Brunelleschi, Piero della Francesca, Leonardo Da Vinci [1] and Albert Durer researched the way a human eye sees the world. They bridged the space between the artist and the object. In Figure 1 you can see the perspective machine which was used to determine the projection of 3-D points to a plane known as the image plane. The projections were found by drawing lines between the 3-D points and the eyes of the artist (vanishing points).

![Perspectiva machine](image)

Figure 1: Perspectiva machine [2]

It is now well known that the same mathematical principles used to understand how humans see the world can be used to build 3-D models from two images. There are many applications for 3-D reconstruction such as: Photo Tourism [3], visualization in tellemersive environments [4] [5] [6], video games (e.g. Microsoft Kinect), facial animation [7], etc.

The stereo computation paradigm [8] is broadly defined as the recovery of 3-D characteristics of a scene from multiples images taken from different points of view. The paradigm involves the following steps: image acquisition, camera modeling, feature extraction, image matching, depth determination and interpolation. There is a large body of literature in the area of computer vision which deals with this problem [9] [10] [11].

In this paper we will describe and show the common steps to build a 3-D visual reconstruction from two images, and we will demonstrate some results using a custom built stereo testbed.

The rest of the paper is organized as follows: section 2 describes the single view geometry; section 3 describes the epipolar geometry for two views; in section 4 the correspondence problem is presented; in section 5 the physical and algorithmic architecture used for our experiment are described; in section 6 some results using our platform are shown; finally, section 7 presents our conclusions.

2 Single View Geometry

Cameras are usually modeled using a pinhole imaging model [10]. As illustrated in Figure 2 images are formed by projecting the observations from 3-D objects onto the image plane.

The image of a point $P = [X, Y, Z]^\top$ in a physical object is the point $p = [x, y, 1]^\top$ corresponding to the intersect between the image plane and a ray from the camera focal point $C$ to $P$. Given that the world coordinate system and the coordinate system associated with a camera coincide (as shown in the figure), we have

$$x = \frac{X}{Z} \quad \text{and} \quad y = \frac{Y}{Z},$$

(1)

where $f$ is the focal length of the camera. However, in the case that the world coordinate system does not agree with the camera frame of reference and the point $P$ is expressed in world

![Pinhole camera model](image)

Figure 2: Pinhole camera model [10].
coordinate system as \( P_0 = [X_0, Y_0, Z_0]^\top \), then we have that
\[
P = RP_0 + t, \tag{2}
\]
where \( R \) and \( t \) are the rotation and translation defining the rigid body transformation between the camera and the world coordinate frames. These are called the **extrinsic parameters** for a camera. Given this model, the coordinates of a point in the image domain satisfy:

\[
\begin{bmatrix}
\lambda \\
x \\
y \\
1
\end{bmatrix} = \Pi_0 \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_0 \\
Y_0 \\
Z_0 \\
1 \end{bmatrix}
\]

where \( \lambda := Z \) and
\[
\Pi_0 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}.
\]

The previous equation is obtained under the assumption of an ideal camera model. In general, the coordinates of the image plane are not perfectly aligned and may not have the same scale (i.e., uneven spacing between sensors in a camera sensor array). This leads to the more general equations:

\[
\begin{bmatrix}
\lambda \\
x \\
y \\
1
\end{bmatrix} = \Pi \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\
Y \\
Z \\
1 \end{bmatrix}
\]

where
\[
\Pi = \begin{bmatrix}
fs_x & fs_y & o_x \\
0 & fs_y & o_y \\
0 & 0 & 1
\end{bmatrix},
\]

\([o_x, o_y]^\top\) is called the principal point, \( s_x \) and \( s_y \) specify the spacing between sensors in the \( x \) and \( y \) directions in the camera sensor array, and \( s_\theta \) is an skewing factor accounting for misalignment between sensors. The entries of \( K \) are known as the **intrinsic parameters** of a camera. All of these parameters can be found by measurements of image points from 3-D points with known coordinates.

In addition to the linear distortions that are modeled by \( K \), cameras with wide field of view can have radial distortions which are modeled using polynomial terms in the model \([10]\). Cameras can be calibrated to obtain undistorted images which we will assume it is the case from now on.

### 3 Epipolar geometry

Epipolar geometry \([9]\) studies the geometric relationship between the points in 3-D and their projections in the image planes of stereo cameras for a vision system.

In Figure 3, one can observe that the 3-D point \( P \) has perspective projection \( p_1 \) and \( p_2 \) \([12]\) in the left image plane \( I_1 \) and in the right image plane \( I_2 \), respectively. The points \( p_1 \) and \( p_2 \), known as **corresponding image points**, can be found by intercepting the image planes with the lines from the 3-D point \( P \) to the optic centers \( C_1 \) and \( C_2 \). The so-called **epipolar plane** is defined by the points \( P \) and the optic centers \( C_1 \) and \( C_2 \).

Due to the geometry of the setup it is possible to verify that the image points \( p_1 \) and \( p_2 \) must satisfy the so called **epipolar equation**:
\[
p_2^T F p_1 = 0 \tag{5}
\]

where \( F \) is the **fundamental matrix** \([9, 13]\) and is defined as
\[
F = K_2^{-T} E K_1^{-1} \tag{6}
\]

The matrices \( K_1 \) and \( K_2 \) are the matrices of intrinsic parameters and \( E \) is the so-called **essential matrix**. The matrix \( E \) is given by
\[
E = [T_{21}, R_{21}] \tag{7}
\]
where \( R_{21} \) and \( T_{21} \) are the rotation and translation matrices between camera 1 and 2, and \( [T]_4 \) for \( T = [T_1, T_2, T_3] \) is defined as
\[
[T]_4 = \begin{bmatrix}
0 & -T_3 & T_2 \\
T_3 & 0 & -T_1 \\
-T_2 & T_1 & 0
\end{bmatrix} \tag{8}
\]

Equation 5 can be used to solve for the fundamental matrix \( F \) given that enough corresponding image points are provided. One such algorithm is known as the **eight point algorithm** \([14]\). Hence, since the matrices \( K_1 \) and \( K_2 \) can be learned from individual calibration of the cameras, then the essential matrix \( E \) can be found. One can solve for the rotation and translation (up to scale) between cameras by using \( E \).

Figure 3 also illustrates one more fact that can be used to find corresponding points between images once the intrinsic and extrinsic parameters are known. Essentially, given a point \( p_1 \), we know that the 3-D point \( P \) has to lay in the plane defined by \( T_{21} \) and the line from \( C_1 \) to \( p_1 \) (i.e., the epipolar plane). This plane intercepts the image plane \( I_1 \) in the line \( l_1 \) and the image plane \( I_2 \) in the line \( l_2 \). That is, we know that
any point in the line \( l_1 \) must correspond to some point in the line \( l_2 \). This turns the identification of corresponding points into a 1-D search problem. This process is known as disparity computation \[15\] \[11\]. Finally, it is common to warp the original image planes such that the lines \( l_1 \) and \( l_2 \) are both horizontal and aligned. This process, known as rectification \[16\], is done to simplify computations during the search.

Finally, given that the intrinsic parameter, extrinsic parameters, and the corresponding points \( p_1 \) and \( p_2 \) are known as well, it is possible to recover the 3-D position of the point \( P \) via a process called triangulation \[17\]. Essentially, \( P \) is found to be the intercept between the line passing through \( p_1 \) and \( C_1 \), and the line passing through \( p_2 \) and \( C_2 \).

4 The Correspondence Problem

In order to estimate the essential matrix \( E \) and to perform the disparity computations, we require the identification of corresponding points (i.e., points in the image planes that are the projection of the same physical point in 3-D). This is a very challenging and ambiguous problem \[11\] due to possibly large changes in perspective between cameras, the occurrence of repeated patterns in images, and the existence of occlusions.

4.1 Generic Correspondence Problem

In order to find corresponding image points for the estimation of the essential matrix \( E \), it is possible to use techniques that identify unique points. The correspondence between points can then be determined by a metric between local descriptors. The SIFT and GLOH \[18\] descriptors are two commonly used techniques.

For the case of calibration \[19\] \[20\] in controlled environments, it is commonly assumed that the observed object has a regular pattern that is easy to identify (e.g., a checkerboard pattern). In these cases, it is possible to use more specialized region detection algorithms such as the corner Harris detector \[11\].

4.2 Disparity Computations

If the intrinsic and extrinsic calibration parameters between two cameras are known, it is possible to rectify the images such that corresponding points can be found along horizontal lines in the images. As mentioned before, finding correspondences turns into a 1-D search problem known as disparity computation. In order to perform this search, it is possible to specify a metric that characterizes the difference between two regions (e.g., normalized cross correlation). Then, we look for the location along the horizontal line that minimizes the difference between these regions \[11\]. This is a very simple methodology that will be used in this paper in order to perform disparity computations. Of course, there are many more sophisticated algorithms that give more accurate results \[15\].

5 Testing Platform

In this section, we introduce the design of a simple stereo platform used to building 3-D models using the framework presented so far.

5.1 Physical Architecture

Figure 4(a) shows a frontal view of two digital camera configuration used for our experiments. Figure 4(b) shows the back view. The cameras were mounted using screws to prevent any motion during the calibration and image acquisition.

The brands of the two digital cameras that were used are Sony DSC-S750 and Canon SD1200 IS. Both cameras were configured to have the same settings.

5.2 Algorithmic Architecture

The Camera Calibration toolbox \[21\] was used to calibrate the cameras. Figure 5 illustrates the steps of our implementation of 3-D visual reconstruction. Image acquisition was done using the camera setup described before. We pre-processing the two images with gaussian filters to reduce noise and Adobe Photoshop CS3 is used to remove the background. The metric used to calculate the disparity map is the normalized cross correlation \[10\]. As a post-processing step, we use a median filter to correct the problem of outliers. After that, we generate a Delaunay triangulation of the disparity image, which is then used to obtain a 3-D mesh by projecting key points from the image plane to 3-D as described in section 3. Finally, the color texture is mapped from the right image to the 3-D mesh.

![Figure 5: Outline of the reconstruction Algorithm](image-url)
Figure 6: (a) Detection of corners of the checkerboard pattern, and (b) detection of internal points of the checkerboard pattern.

Figure 7: (a - b) Original images of the cube, (c - d) rectified images, and (e - f) images without background.

6 Results

In this section we present some 3-D reconstruction results using the methodology described in this paper. We also discuss limitations and drawbacks.

6.1 3-D Reconstruction Results

We used a checkerboard pattern of $10 \times 7$ squares for the calibration. Figure 6 shows two images of this process in which corner points are automatically selected using the Camera Calibration toolbox [21].

Most of the implementation is done in MATLAB. However, Paraview Viewer [23] is also used for the visualization of the point clouds, which are processed using the decimate and smooth filters from the application.

Figure 8 illustrates the 3-D reconstruction obtained from the cube images. Sub-figure (a) is the processed disparity map obtained from the pre-processed rectified images. Sub-figures (b) and (c) illustrate a cloud of points and the resulting 3-D mesh, respectively. Sub-figures (d) and (e) show the surface of the cube without and with smoothing. Finally, sub-figure (f) corresponds to the 3-D mesh with the color texture from the rectified right image.

Figures 9 and 10 show similar results using a pair of images of a teddy bear instead of a cube.

6.2 Limitations and finding problems

One limitation was the lack of lighting control during the image acquisition process, which causes artifacts such as shadows and color variations between images as observed in Figure 11 (a) and (b). These artifacts cause errors in the disparity calculation as observed in Figure 11 (c), which translate into inconsistent reconstruction of the object as observed in Figure 11 (d).

Figure 8: (a) Disparity map for the cube images, (b) matching points, (c) 3-D mesh, (d) reconstructed surface, (e) smoothing of the surface, and (f) surface with texture.
As described earlier, we used a normalized cross correlation approach to compute the disparity map. This process requires the specification of a neighborhood, which is a challenging problem on its own [10]. The disparity process also requires some knowledge of the maximum offset between corresponding points [11], which may not be known in advance. Figure 12 illustrates a comparison between a failed and a good reconstruction using the approach described in this paper.

Tests were performed on a machine Sony VAIO VGN-FE880E with a processor Intel (R) Core (TM) 2 CPU 1.66 GHz and memory RAM 2GB with an operating system Windows 7 Ultimate 64-bit. Time for the processing of our algorithm is twenty-five minutes.

**7 Conclusion**

In this paper, we outline all the steps needed to generate a 3-D reconstruction of an object by using a stereo image pair. These 3-D models can be used in a variety of applications ranging from photo tourism [3] to virtual conferencing [6]. The method used for disparity computation uses a simple normalized cross correlation to figure out correspondences, which is not a very robust approach but it is simple enough for demonstration purposes. There are more robust methods that use techniques such as dynamic programming and genetic algorithms [15].

One of the main challenges observed during the reconstruction process was the variability of lighting conditions, which can lead to artifacts in the reconstruction. There is also a large dependency between steps in the reconstruction. For example,
small errors in calibration affect the rectification process, causing problem in the disparity computation, and leading to erroneous 3-D reconstructions. For this reason, it is necessary to put special care to the design of a stable physical architecture for image acquisition.

Acknowledgments

The authors of this paper template would like to thanks to: Alex Cuadros, Juan Carlos Gutierrez and Nestor Calvo.

References
