
Real time depth computation using stereo imaging

Prashant Jagdishchandra Bagga

Mechanical Engineering Department, IDS, Nirma University, Ahmedabad, India

Email address:

prashant.bagga@nirmauni.ac.in

To cite this article:

Prashant Jagdishchandra Bagga. Real Time Depth Computation Using Stereo Imaging. *Journal Electrical and Electronic Engineering*. Vol. 1, No. 2, 2013, pp. 51-54. doi: 10.11648/j.jee.20130102.13

Abstract: Stereo imaging is one of the methods to get an estimate about depth distance of various objects in the scene from the camera. This paper presents the process involved in developing a stereo imaging module for use on a robot to plan paths and movements, and for avoiding obstacles. The module is implemented using the open source computer vision library OpenCV. This module works in real-time and can deal with dynamic scenes to a fair extent.

Keywords: Stereo Calibration, Re-Projection, Bouguet's Algorithm, Hartley's Algorithm

1. Introduction

A single picture taken from a normal camera does not give any idea about depth. There are two logical alternatives to find depth: 1. Structure from motion and 2. Stereo Imaging. Structure from motion method, uses many images of a static object taken by a single camera which is in motion, so that all angles of interest are covered. The images are then used to recreate not only the trajectory of the camera but also a 3D representation of the environment. Advantage of this method is that it requires only one camera and hence saves cost. The disadvantage of this method is that the pictures so taken are subjected to almost the same amount of processing as in stereo imaging, so this method does not save on processing power needed. Instead, complexity is added to plan trajectories if used on a robot. Also it requires a static environment, e.g. a building. Stereo imaging uses a special instrument which has two cameras mounted on it and is first calibrated and then used to capture pairs of images. The calibration data is utilized after that in computing depth. Advantages of this method are that once calibrated, one pair of images is enough to get depth and no planning of trajectory is needed. Hence, this approach is suited to dynamic environment that autonomous robots typically face. Disadvantage of this method is that it requires a special instrument which has two cameras immobile with respect to each other and which is synchronized, if possible at the hardware level.

Considering the above aspects and noting that the long term goal is to deploy this technology on an autonomous robot which is likely to encounter dynamic environments,

stereo imaging is selected to compute depth in real-time.

The Stereo imaging process: The four major steps in stereo imaging are:

1.1. Un-Distortion

This mathematically removes the radial and tangential lens distortion. The outputs of this step are undistorted images.

1.2. Rectification

The objective of this step and the next is to achieve a frontal parallel configuration of the cameras. This requires adjustment for the angles and distances between cameras. As mentioned earlier, the two cameras are mounted facing the same direction with all possible precision, but some amount of adjustment by software is always required. The outputs of this step are pairs of images that are row-aligned and rectified.

1.3. Correspondence

This finds the same features in the left and right camera views and aligns images such that same areas in both fall on same lines. The output of this step is a disparity map, where the disparities are the differences in x-coordinates on the image planes of the same feature viewed in the left and right cameras.

1.4 Re-Projection

If we know the geometric arrangement of the cameras, then we can turn the disparity map into physical distances

by triangulation. The output is a depth map.

Last step is described initially, so that the motivation for the rest is easily understood.

2. Re-Projection by Triangulation

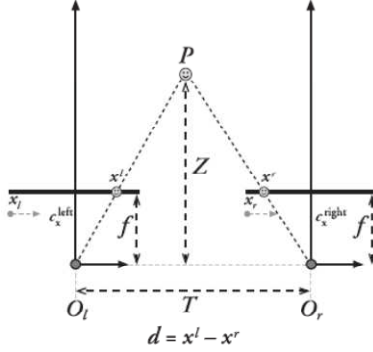


Figure 1. The Frontal Parallel Configuration for Re-Projection

It also shows the optical axes and centres of view of both the cameras. The focal length of each is f . T is the baseline. x_l and x_r are the camera positions of P in the left and right cameras respectively. From the geometry of the figures it is evident that depth Z can be calculated as –

$$\frac{T - (x_l - x_r)}{Z - f} = \frac{T}{Z} \rightarrow Z = \frac{fT}{x_l - x_r}$$

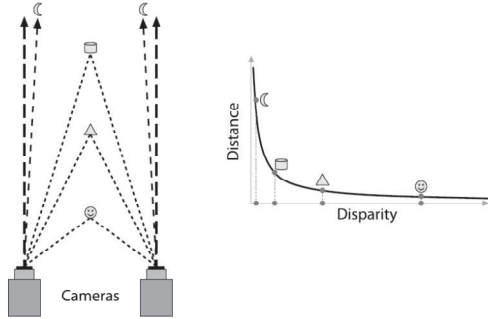


Figure 2. Relationship between Depth And Disparity

Since depth is inversely proportional to disparity, there is obviously a nonlinear relationship between these two terms. When disparity is near 0, small disparity differences make for large depth differences. When disparity is large, small disparity differences do not change the depth by much. The consequence is that stereo vision systems have high depth resolution only for objects relatively near the camera, as Figure 2 makes clear.

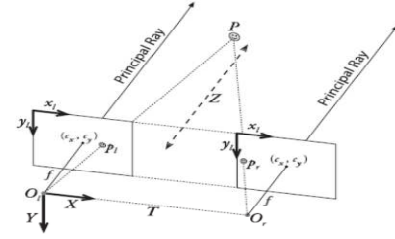


Figure 3. Relationship between Depth and Disparity

The centres of projection are at O_l and O_r with principal rays intersecting the image plane at the principal point (not the centre) (c_x, c_y) . After mathematical rectification, the cameras are row-aligned (coplanar and horizontally aligned), displaced from one another by T , and of the same focal length f . With this arrangement it is relatively easily to solve for distance. In the real world, cameras will almost never be exactly aligned in the frontal parallel configuration depicted in Figure 3. Instead, we will mathematically find image projections and distortion maps that will rectify the left and right images into a frontal parallel arrangement. When designing stereo rig, it is best to arrange the cameras approximately frontal parallel and as close to horizontally aligned as possible. This physical alignment will make the mathematical transformations more tractable.

If cameras are not aligned at least approximately, then the resulting mathematical alignment can produce extreme image distortions and so reduce or eliminate the stereo overlap area of the resulting images. For good results, synchronized cameras are needed. If they don't capture their images at the exact same time, then there is a problem if anything is moving in the scene (including the cameras themselves). If cameras are not synchronized, it will limit use of stereo with stationary cameras viewing static scenes.

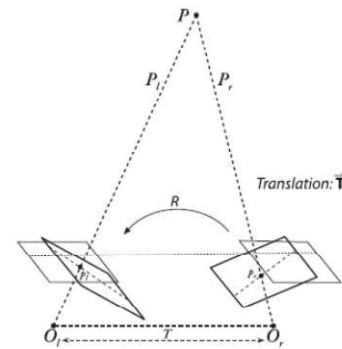


Figure 4. Alignment of two cameras

Figure 4. Depicts The Real Situation Between Two Cameras And The Mathematical Alignment We Want To Achieve. To Perform This Mathematical Alignment, It Is Required To Learn More About The Geometry Of Two Cameras Viewing A Scene.

3. Stereo Calibration

Calibration in stereo imaging means to compute various parameters associated with the stereo setup. This includes -

1. The rotation vector and translation vector between the two cameras.
2. Rotation vector for each camera to get a frontal parallel arrangement.
3. The intrinsic matrices or camera matrices of the two cameras.
4. The distortion matrices of the two cameras.

3.1. Calibrating a Single Camera

By calibrating a single camera, we can find out the intrinsic matrix, distortion coefficients and rotation and translation vectors of the camera with respect to the co-ordinate system of the object. For the last, we will need an object, certain points of which can be easily isolated by algorithms. A classical example is a chessboard. Figure 5 shows the image of a chessboard and its corners isolated by OpenCV's implementation of Zhang's algorithm [2].

3.2. Calibrating the Stereo Rig

For any given 3D point P in object coordinates, we can separately use single-camera calibration for the two cameras to put P in camera coordinates for the left and right cameras, respectively:

$$P_l = R_l P + T_l,$$

$$P_r = R_r P + T_r$$

The two views of P are related by:

$$P_l = R T (P_r - T).$$

Taking these three equations and solving for the rotation and translation separately yields –

$$R = R_r (R_l) T$$

$$T = T_r - R T_l$$

Now if many views of a chessboard are given, we can find out R_l , T_l , R_r and T_r (and hence R and T) for each view using the above equations. Because of image noise and rounding errors, each chessboard pair results in slightly different values for R and T . We take the median of these values as the value of R and T for the stereo rig. The intrinsic matrix and the distortion coefficients are computed in the same way.

4. Stereo Rectification

Our aim in stereo rectification is to re-project the image planes of the two cameras so that they reside in the same plane. Once we re-project the images to get them in the same plane, we have to align the two images in the same line. Then we want to crop the images to only show the overlapping area. Figure 5 shows the process of stereo rectification.

We want the image rows between the two cameras to be aligned after rectification so that stereo correspondence (finding the same point in the two different camera views)

will be more reliable and computationally tractable. Note that reliability and computational efficiency are both enhanced by having to search only one row for a match with a point in the other image. Bouguet's algorithm is used for stereo rectification.

Bouguet's algorithm [4] for stereo rectification simply attempts to minimize the amount of change reprojection produces for each of the two images while maximizing common viewing area.

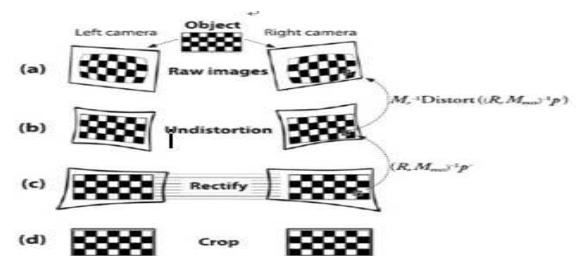


Figure 5. Stereo Rectification

4.1. Stereo Correspondence

Once images from the two cameras are in the same plane and aligned by line, we want to match points in the same line on both images and find disparity. The matching is done by stereo correspondence. OpenCV uses the Block Matching algorithm for this. This algorithm finds only strongly matching (high-texture) points between the two images. Thus, in a highly textured scene such as might occur outdoors in a forest, every pixel might have computed depth. In a very low-textured scene, such as an indoor hallway, very few points might register depth. There are three steps involved in the algorithm Pre-filtering Correspondence Post filtering

The OpenCV (open-source computer vision) library [5] was used to implement the real time stereo imaging system.

5. Conclusions

A platform independent module is developed for real time computation of depth. The module gives satisfactory results for a fairly dynamic environment. This module can be deployed on robots for detecting obstacles and hence planning paths. Its main advantage is that the Block Matching algorithm for computing correspondence is very fast and hence works well in real time. One specific application would be on a robot with an ability to navigate itself and an arm to grip objects

A major improvement would be to use a professional quality hardware-synchronized stereo camera, like the Bumblebee 2 from Point Grey Research. This would greatly increase the number of points for which the disparity is found.

The Graph Cut algorithm for stereo correspondence can also be used instead of the Block Matching algorithm. It gives highly accurate disparities for almost all the points in the view. However it is very heavy in terms of computation

required. So it should be used to take ‘snapshots’ when the robot reaches near to the object of interest.

References

- [1] “Learning OpenCV”, Gary Bradski and Adrian Kaehler, O’Reilly Media, 2008.
- [2] “A flexible new technique for camera calibration”, Zhang, Microsoft Research.
- [3] Wikipedia, “Epipolar Geometry”, Internet: http://en.wikipedia.org/wiki/Epipolar_geometry
- [4] “Camera calibration toolbox for Matlab: First example - corner extraction, calibration and additional tools”, Jean-Yves Bouguet
- [5] “OpenCV 2.0 C reference”, Internet: <http://opencv.willowgarage.com/documentation/>
- [6] Point Cloud Library, “PCL visualization”
- [7] Book “Digital Image processing”, Second edition, Rafael C Gonzalez /Paul Wintz, Addison-Wesley Publishing Company.
- [8] “Foveated gaze-contingent displays for peripheral LOD management, 3D visualization, and stereo imaging”, Andrew T. Duchowski, Arzu Çöltekin, Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) , Volume 3 Issue 4
- [9] “Stereo imaging with uncalibrated camera”, Xiaokun Li, Chiman Kwan, Baoxin Li, Nov 2006, ISVC’06: Proceedings of the Second international conference on Advances in Visual Computing – Volume Part I, Volume Part I
- [10] “Guidelines for 3D positioning techniques”, Robert J. Teather, Wolfgang Stuerzlinger, Nov 2007, Future Play ’07: Proceedings of the 2007 conference on Future Play, ACM.