A SIMULATOR FOR DEPTH ESTIMATION

A stereo-vision system that was developed for application in mobile robots turned out to lack depth resolution in the background of the pictures. A simulator was built to gain understanding of the parameters that influence depth estimation in stereo vision. In this article we will explain how these properties influence depth resolution and provide a link to the webtool that was made to interactively observe and evaluate the resulting depth resolution when the parameters are varied. This tool makes it possible to find the correct hardware that provides the resolution required, or to determine the resolution for specific hardware.

VICTOR SLUITER

Introduction

Stereo vision is widely used in robotics, in applications where robots need to orient themselves in 3D to pick & place items, but also in applications where mobile robots need to recognise and avoid obstacles. When stereo vision is needed in an application, the system designer can either buy an existing solution, such as the widely used Intel® RealSense™ cameras, or leverage open-source solutions to build a stereo-vision system that meets their needs.

FireBot

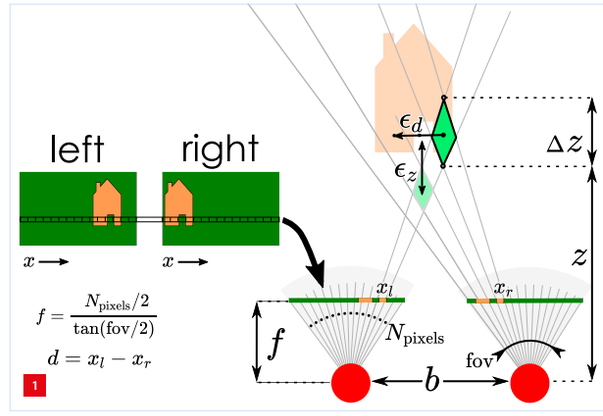
The application that piqued our interest in stereo vision is the *RAAK Publiek* FireBot project, funded by SIA, the Dutch Taskforce for Applied Research (*Nationaal Regieorgaan Praktijkgericht Onderzoek*). In this project, the Mechatronics research group at Saxion University of Applied Sciences (project coordinator) collaborates with the University of Twente, three companies and four Dutch fire brigades to improve the abilities of their unmanned reconnaissance robots. One of the research topics is to create a 3D map while driving in an environment that is filled with warm, dense smoke. Regular RGB vision (3D) systems are not suitable due to high scattering of the light by the smoke and water particles. Therefore, research was carried out using thermal cameras, detecting long-wavelength infrared.

While designing the system, we needed to select a thermal camera and choose the right resolution value and lens. One of the requirements was the depth resolution, so we had to determine what depth resolution could be achieved at a particular distance from the cameras and which parameters influence the resolution and the depth quantisation effects. The depth resolution is the smallest possible difference in depth that can be measured at a certain depth, and determines what level of detail can be observed. The quantisation is a result of having discrete pixels, and results in objects close to each other being assigned the same depth value.

Although the application with thermal cameras might be specific to our application, the methods and analysis described below are generic for all kinds of stereo-imaging systems including visible light.

Making depth estimations

Before any calculations are made on stereo resolution, the most common depth estimation algorithms make a few assumptions: the images do not have lens deformation and the cameras are perfectly coplanar. In most real-life situations a calibration is needed to correct for lens deformation and non-ideal camera placement. This is achieved by image undistortion and rectification based on a set of pictures made

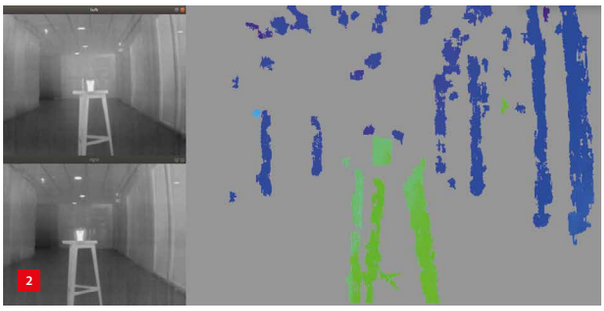


Schematic representation of depth estimation. The house is pictured by both cameras; a feature that is seen in both images leads to a depth by triangulation of the pixel coordinates (x_l and x_r) and the baseline distance b. The green diamond shows the area that is formed by the intersection of the two pixels.

AUTHOR'S NOTE

Victor Sluiter is a research engineer in the Mechatronics research group of Saxion University of Applied Sciences, located in Enschede (NL).

v.i.sluiter@saxion.nl
www.saxion.nl/onderzoek/smart-industry/mechatronica



A disparity map (right image) is generated from two thermal images (left) for a stool with a hot cup of coffee on top of it. Depth is rainbow-colour-coded from red (close) to purple (far away). Only green to purple are visible in this picture. Note how it is possible to observe that the middle leg is darker green and hence pointing backwards, something which cannot be deduced from either of the two thermal images.

of a known calibration object. More information can be found on Wikipedia [1] and the OpenCV (Open Source Computer Vision Library) website [2].

Figure 1 explains how depth is estimated. The two cameras (red circles) both see the house. A 'block matching' algorithm searches each pixel row for corresponding features in the left and right images. In this case the wall to the right of the door was found in both pictures. Since the baseline b and the focal length f (expressed in number of pixels; the relation with the field of view, fov, is shown in Figure 1) are fixed parameters in the set-up, the depth can be calculated: $z = f / (b \cdot d)$. Here, d is the number of pixels the feature has shifted between both images, also known as the disparity. Now the depth z is derived by triangulation.

In Figure 2 it can be seen how the visualisation algorithm from OpenCV shows the disparity as a colour map. A large disparity (features far away from each other when comparing the left and right image) means the object is relatively close to the camera, and when a feature is further away, it will be in almost the same location in both images. The resulting disparity image is rainbow-colour-coded and shows the nearest objects in this image as green/blue and objects in the background as purple. In the grey area no corresponding features could be found and no disparity could be calculated.

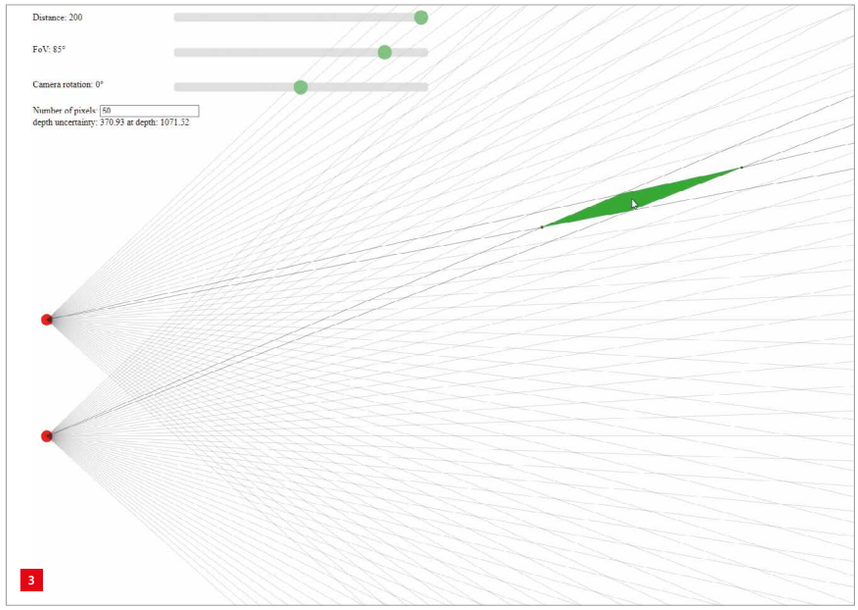
Note: Due to occlusion of part of the hind leg by one of the front legs, the disparity image can show only (the other) part of the hind leg. However, the region where the two legs 'overlap' is partly grey, which unjustifiedly suggests that the legs are separated (no occlusion). This is because here the disparity can not be determined due to the occlusion: one camera 'sees' the hind leg, but the other camera (the left one, in this case) does not.

One of the observations we made was that the depth resolution degrades progressively with increasing distance. From literature [3] we already know that there is a quadratic increase in depth error ϵ_z for a given distance z, caused by a disparity error ϵ_d:

$$\epsilon_z = (z^2 / f \cdot b)\, \epsilon_d$$

The disparity error ϵ_d (error in number of pixels) is caused by the block matching algorithm finding the wrong pixel as a feature match in the row (i.e. the wrong x_l or x_r, in this case because of erroneously selecting the wall to the left of the door as the relevant feature in one of the two images). This effect is illustrated in Figure 1, where a wrong match is shown with a dim green diamond. Although this explains depth error due to matching error, the artefacts we noticed in our disparity images were a very clear quantisation effect; objects close to the camera could be imaged with great detail in depth, but further from the camera all objects that had clear features would all get assigned the same depth. We saw the same depth, i.e. a 'binning' of depth values.

To better understand the theory behind stereo vision we developed a simulator. This simulator shows graphically what the implications of design choices are, for example what the nearest point is where the two images have overlap (from this point on, disparity and hence depth can be determined). What also appeared clearly when using the tool is that the depth quantisation (binning) error becomes larger for objects further away; the quantisation effect is also quadratic. From Figure 1 it can be deduced that the match of two features results in the green diamond-shaped area where the feature originated from. With fewer pixels, a smaller baseline or larger field of view, the diamonds become larger and the real object generating the image



Screenshot of the simulator. The intersection of camera pixels under the mouse cursor is highlighted and depth z and size of the bin Δz are given. Note the change in area of the intersections of pixels.

feature can be anywhere within that diamond. The size of the depth bin Δz can be approximated by:

$$\Delta z = 2\ (z^2 / f \cdot b)$$

Online tool

The simulation tool [4] is publicly accessible and can be used to perform some quick calculations and especially to obtain an intuitive understanding of how camera set-up parameters affect performance. This gives a better understanding of how vision set-up parameters influence the depth estimation in stereo-vision applications.

Parameters that can be entered are the baseline distance, number of pixels per row, horizontal field of view and camera rotation. The visualisation updates 'live' while changing parameters, showing the 'binning' of the possible depth values, but also the overlap of camera view. When moving the cursor across the screen, the depth value z and the size of the depth 'bin' Δz are shown for the area the mouse is hovering over. In this way, trade-offs can be made, for example between longer focal length for better resolution and shorter focal length for imaging closer by. Although we consider this a valuable tool, some disclaimers have to be made. First of all, the calibration discussed above will introduce artefacts in the image, so the angular field of view per pixel in the rectified and 'undistorted' image will probably be slightly different from the angular field of view per pixel of the 'raw' image. Secondly, we show the distance z and the size of the depth 'bin' Δz. It is disputable whether the depth should be represented as a depth in the centre of the diamond with a possible positive and negative deviation, because the larger the depth, the more distorted the diamond gets and hence the less meaning the 'centre' has). For clarity of presentation we decided to use the representation shown in Figure 3.

REFERENCES
[1] *en.wikipedia.org/wiki/Image_rectification*
[2] *docs.opencv.org/4.4.0/d9/db7/tutorial_py_table_of_contents_calib3d.html*
[3] D. Gallup et al., "Variable baseline/resolution stereo", *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, *ieeexplore.ieee.org/document/4587671*
[4] *tiny.cc/disparity*