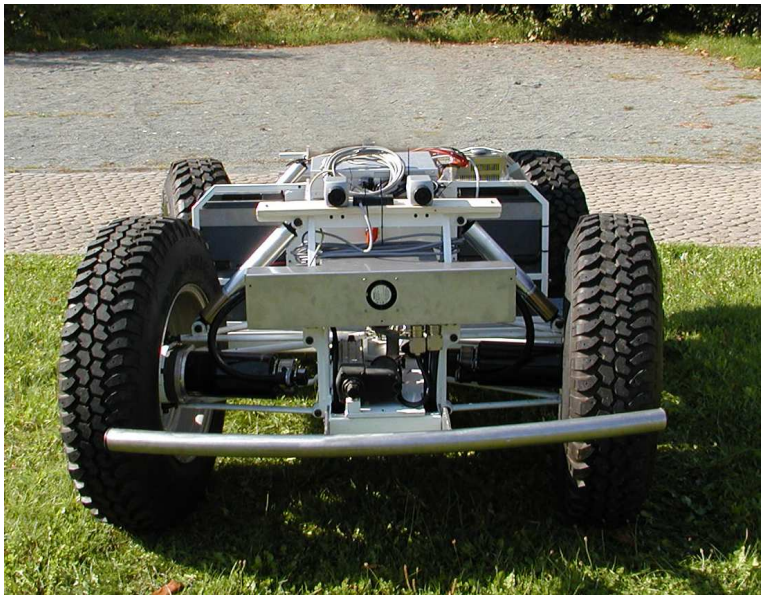
ROBOTICS LABORATORY
DEPARTMENT OF COMPUTER AND
INFORMATION SCIENCE
KAISERSLAUTERN INSTITUTE OF
TECHNOLOGY

Technical Report



Stereo Vision for Mobile Robotics

Bernd Helge Schäfer

September 15, 2004

Contents

1. Introduction

1.1 Motivation

Following recent development in unmanned space travel [Iagnemma 03, Schenker 03], agricultural automation [Thuilot 01, Lenain 03, Wellington 04], archaeological exploration [Gantenbrink 99] and evolution of military devices [Hong 02, Debenest 03] it becomes apparent that the need for unmanned ground vehicles is strongly increasing.

In order to implement self-dependent mobile platforms a variety of sensor systems need to be made available such that the robot can perceive and intelligently interact with its environment. The sensory equipment mostly comprises range and visual sensors like laser range finders and video cameras. Equipped with pairs of cameras, the system will be capable of extracting range data from the video images. Although this method is much less accurate than state-of-the-art laser scanners, so called stereo camera systems are often used for rough range estimates. Additional colour information from the cameras can be incorporated into the range image as well.

As laser range finders are relatively heavy and expensive compared to video cameras, the latter are often preferred. Furthermore many military application forbid the usage of activly scanning sensors such that video cameras appear to be the means of choice. In application fields where precise range as well as colour information is crucial, hybride sensor-fusion approaches are often called into service.

1.2 Objectives

In the scope of the project at hand a stereo vision system has been set up. This comprises the integration of basic camera control facilities into the MCA[1] framework, the necessary calibrations and the integration of components crucial for the extraction of range data from stereo images.

As a basis for accessing the cameras, an open source fire wire digital camera library has been used. The computation of the disparity map itself has as well been deligated to a third party component. During the realisation, interchangability of the

[1] MCA → Modular Controller Architecture, `http://mca2.sourceforge.net/`

disparity map generating component has been a central matter. In future projects the infrastructure generated here shall be used for profound research on 3D-vision. This will as well comprise projects on disparity map generation.

2. Range Extraction from Stereo Images

2.1 Crash Course in Stereo Vision

In this chapter the basic principle of stereo vision and range data extraction from pairs of images is going to be outlined briefly. For more detailed and technical descriptions the reader may have look at [Forsyth 03].

The fundamental idea behind stereo vision is to exploit the fact that a feature in two pictures taken with cameras which are a few inches apart appear in slightly different locations. If the images are overlapped, this phenomenon can easily be perceived. The x-distance between two corresponding points or locations is called disparity and can be calculated with assistance of matching algorithms.

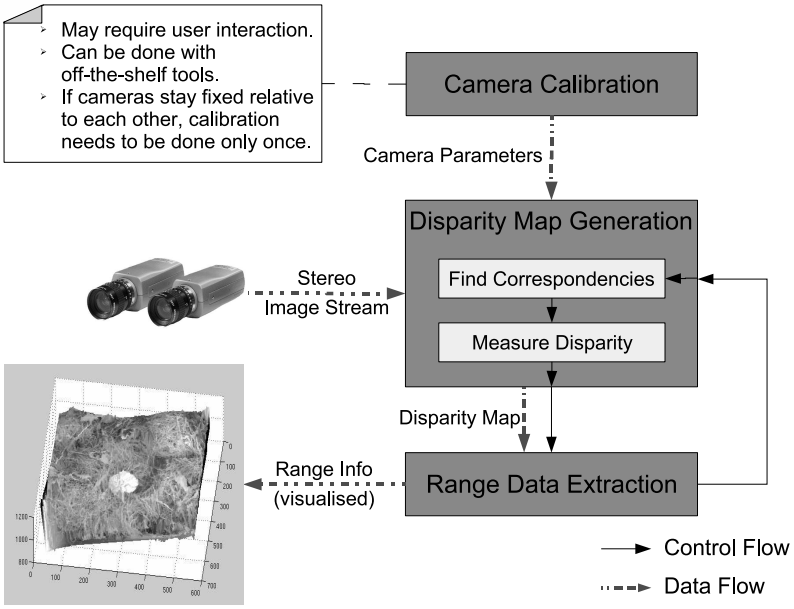


Figure 2.1: Outline of a typical Stereo Vision System.

Such a step yields a *disparity map* which represents what corresponding spots are more distant from each other and which are less. The information is usually coded into eight bit values per pixel such that the disparity map can easily be visualised as a greyscale image. The darker the pixel in the map, the smaller is the x-distance between the corresponding spots in the pair of images. Section 2.1.2 gives some more details on how such a map is computed.

Based on the disparity map, 3D coordinates can straight forwardly be extracted via triangulation. An intuitive summary of the transformations necessary will be provided in Section 2.1.3 before evolving the mathematical formulae for the 3D reconstruction in detail.

Figure 2.1 recapitulates the fundamental steps necessary for successful depth perception using stereo vision. Here the 3D information obtained has been accumulated with colour information from the camera images. The visualisation of the 3D reconstruction has kindly been provided by SIC[1] at RMA[2].

Additional to the steps mentioned above, colour and geometrical calibrations will become necessary in practice. A short introduction to both topics will be given in the next section before going into detail with disparity map generation and interpretation.

[1]SIC → Signal & Image Centre, `http://www.sic.rma.ac.be`
[2]RMA → Royal Military Academy (Belgium), `http://www.rma.ac.be`

2.1.1 Camera Calibration

Play of Colours

In theory stereo vision incorporating the steps of disparity map generation and range extraction via triangulation appears to be a sound and straight forward concept. As very often in robotis, noise and sensory limitations in accuracy, i.e. the interaction with the real world, thwarts simple implementations.

Since two cameras are needed for the recording of the image data, a fundamental problem arising in stereo vision is the fact that these two cameras will never voluntarily deliver the same light intesity or colours. Figure 2.2 shows two pictures grabbed at the same time with two cameras of the same type. Although the parameters of both cameras have been set to identical values the images look very different in colour and intensity. Due to such discrepancies the matching algorithm being deployed for the generation of the disparity map will probably fail to identify corresponding spots correctly. This can result in complete failure of the range data extraction.

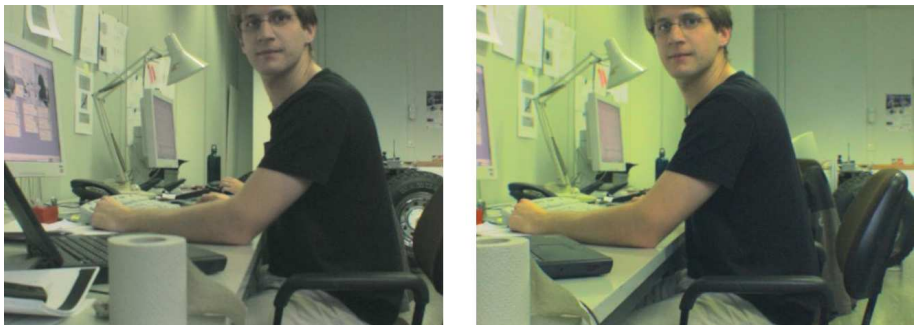


Figure 2.2: Discrepance in Light Intensity and Colour.

Luckily the colours do not need to be realistic, only the difference between one colour perceived by the two different cameras is supposed to be similar. Furthermore many matchers work on greyscale images and thus purely rely on brightness values. This makes things already easier but in the end such calibrations are not trivial.

In the scope of this project histogram adjustment techniques have been used for calibration purposes. The fundamental approach briefly sketched comprises the histograms comparison of greyscale images taken by both cameras. The parameters are manually adjusted until the histograms resemble each other. Note that it is crucial that the cameras capture more or less identical images from the scenery chosen for calibration.

The setup designed for the greyscale histogram adjustment is depicted in Figure 2.3. At the bottom, the histograms for the left (a) and the right (b) greyscale image above are shown[3]. The LCD box *Histogram Similarity* (c) shows a normalised measure for the resemblance of the two histograms. The measure is in the interval [0,1] where 1 means, that the histograms are identical and 0 that they have nothing in common.

[3]In Figure 2.3 the greyscale images have already been rectified and undistorted. For details on rectification an undistortion see paragraph *Camera Geometry, Rectification and (Un)Distortion* of this section.

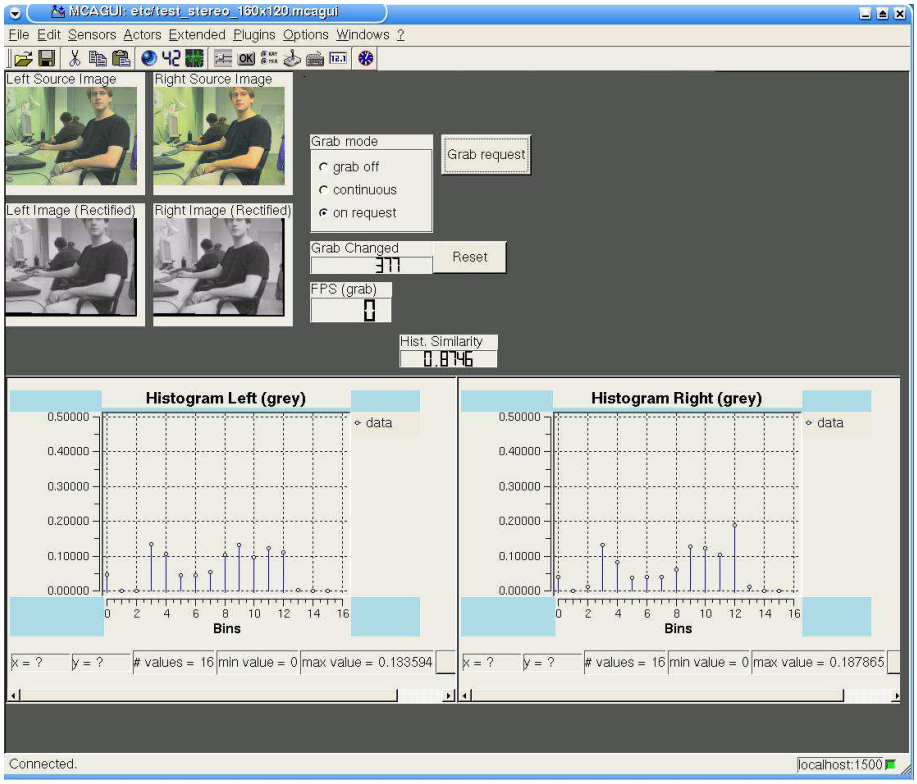


Figure 2.3: Histogram Adjustment Platform

Note that the scene chosen in Figure 2.3 it not optimal. As the cameras ideally should capture the same image it is best to take a view where all objects are far away from the cameras. That way occlusion effects which result from the object's three dimensional nature are minimal.

For details on this and other more sophisticated colour calibration techniques, the reader should have a look at technical literature in computer vision (p.ex.: [Porikli 03]).

Camera Geometry, Rectification and (Un)Distortion

Range data extraction from a pair of images relies on the matching of geometrical features. The core task of matching spots represents a search problem which can efficiently be tackled on the basis of epipolar line assumption. I.e.: One feature in pictures taken with parallely mounted cameras should reside on the same horizontal line in both images (See Figure 2.4). That way the search space can tremendously be constrained. Similar structures need only be compared if they are found at similar horizontal levels of the image.

In order to exploit the fact described above it is necessary to mount the cameras such that the optic axis are parallel to each other and both perpendicular to the baseline[4]. Later it will become apparent that this setup is also handy for a straight forward 3D reconstruction. Actions performed to yield such a setup is called camera rectification. First of all the cameras should be adjusted manually such that the epipolar assumption holds more or less. In the literature software-based approaches

[4]baseline → line connecting the lens centres of the cameras

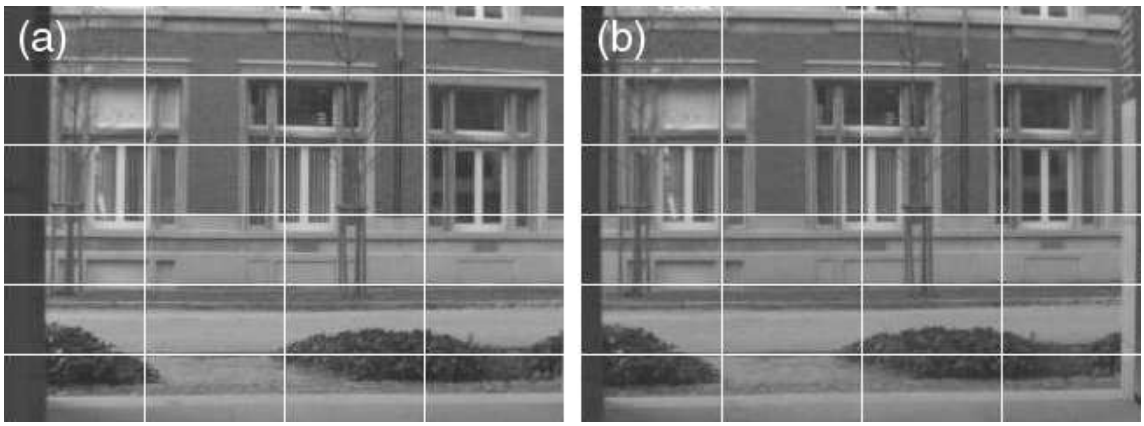


Figure 2.4: Grid-Segmented Images from the left (a) and the right (b) camera.

for enhancing this manual setup by applying static rectification transformations can be found (P.ex.: [Lotti 97]). In the scope of this project the aspect of rectification has widely been neglected as the base construct did not provide many manual rectification options.

A second effect having a severe impact on the assumption made above is distortion introduced by the lenses of the cameras. The general fact that matching algorithms in the stereo vision domain mostly rely on geometrical characteristics of features (See [Horn 01]; Page 299ff) implies that distortion effects in the input images need to be as minimal as possible. Depending on the camera lens quality and the reception unit, these effects can have severe impact on the range data extraction. Therefore further geometrical calibrations are usually necessary in practice. In the context of this project a flexible method requiring little additional equipment has been used.

The procedure is carried out fairly easy. Several images of a chessboard pattern with different orientations in space are captured. After that the corners and fields of the chessboard have to be outlined to the calibration algorithm. With the knowledge of the dimensions of one chessboard field it is possible to compute the plane in which the chessboard has been captured relative to the cameras. The distance and the orientation difference between the two cameras, often referred to as extrinsic camera parameters, can that way be determined. If sufficiently many samples are taken, the deviation between expected and actual position of the chessboard field corners can be determined, which results in the intrinsic camera parameters focal length[5], principal point[6] and distortion coefficients.

These parameters can then be used in order to undistort the images captured by the camera. This process is usually done in software. In Figure 2.5 (a) an image is shown as it comes from the camera. The strong distortion effects seen there can almost be compensated through software transformations which yield the image depicted in Figure 2.5 (b).

The calibration technique introduced by Zhengyou Zhang[7] has been applied in the *Camera Calibration Toolbox for Matlab*. The package is freely available on the inter-

[5]Distance from the lens centre to the image plane.
[6]Point in the image, which resides on the optical axis.
[7]Microsoft Research

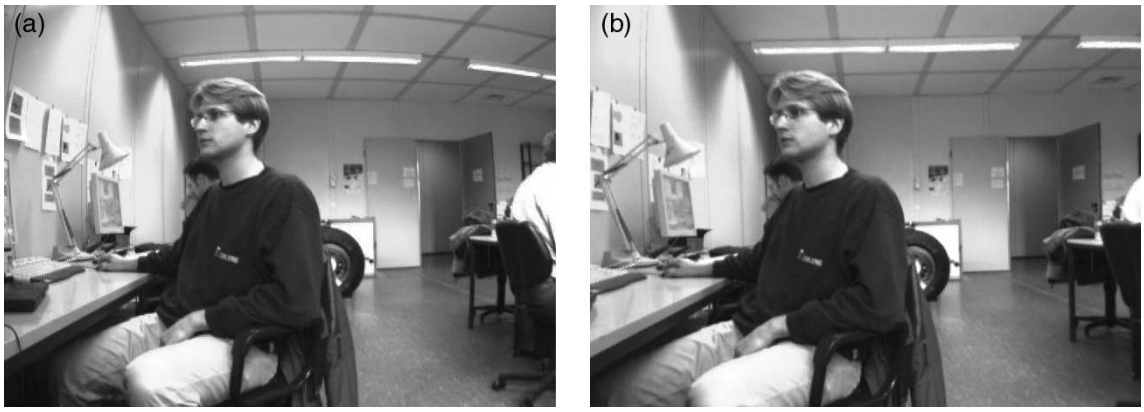


Figure 2.5: Source Image (a) versus Undistorted Image (b).

net[8] and has been used during the work at hand. A detailed tutorial comes with the distribution. For mathematical details on the method the reader shall be referred to [Zhang 99]. In the following it is assumed that the camera stereo head is calibrated sufficiently accurate.

2.1.2 Obtaining a Disparity Map

As already aluded above the first step to range information extraction from stereo image pairs is the computation of the disparity between an object displayed in the two images. In Figure 2.6 the concept of pixel disparity in the context of a stereo camera system is shown.

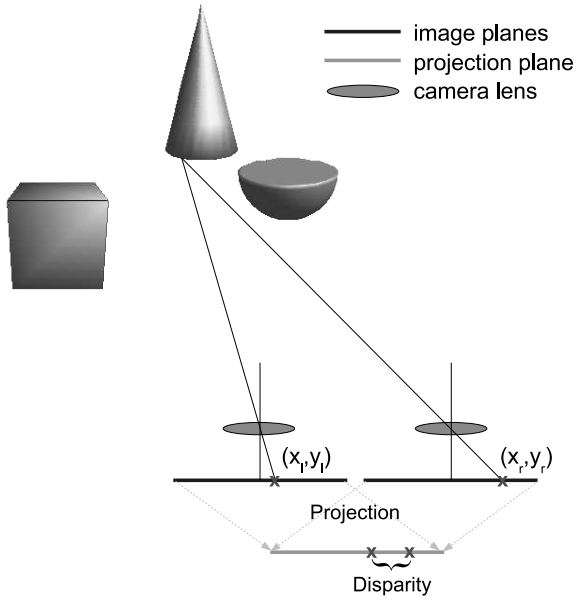


Figure 2.6: Concept of Pixel Disparity in the Context of Stereo Vision.

An object which is captured by both cameras is projected to the 2D image plane of each camera. One camera sees the object slightly more from the right and the other

[8] http://www.vision.caltech.edu/bouguetj/calib_doc/

more from the left side. That way the pixels representing the same feature of this object, reside in different locations in the two image planes (Figure 2.6 (x_l, y_l) and (x_r, y_r)). If the images are then projected into one common plane, the x-distance between pixels representing the same feature, can be measured. The resulting value is called *pixel disparity*. Mathematically speaking the pixel disparity is $x_l - x_r$. In order to compute pixel disparities, first of all corresponding spots have to be discovered. This is the complicated part of any disparity map generating algorithm. Given a set of corresponding features in the two images, the x-distance between them can be determined in a straight forward fashion and coded into scalar values.

Usually these values are stored in eight bits and are thus situated in the integer interval [0...255]. They describe the disparity between two matching locations in pixels. This is beneficial since disparity maps can that way be displayed as greyscale images with eight bits per pixel (See Figure 2.7). The lighter areas are closer to the camera, since the disparity of the correnspponding spots is greater and thus closer to 255. Consequently the darker spots represent objects, which are farther away from the cameras, as the disparity between matched points is smaller. In the disparity map depicted in Figure 2.7 (c) the two persons at the desks can clearly be distinguished concerning the distance from the cameras. Furthermore part of the computer display and a notion of the desk lamp can be seen.

Figure 2.7: Left (a) and Right (b) Image and the computed Disparity Map (c).

In the context of this work finding corresponding pixels in the images, computing the disparity and creating a map has been deligated to a third-party library implemented according to [Birchfield 99]. If interested in that kind of computations the reader should have a look at this very detailed work. In the following section the transformation of a disparity map into 3D coordinates will be explained.

2.1.3 How to Interpret a Disparity Map

Up to now relative rage information has been extracted from the stereo images in terms of the pixel disparity between matching locations given as a disparity map. The next step is the extraction of 3D coordinates from the 2D disparity map. The fundamental idea of how to do this will be presented in this Section.

Figure 2.8 recycles the schematic drawing of a stereo camera system, introduced in the last section. The cameras are assumed to be mounted such that the optic axis are parallel to each other and perpendicular to the baseline[9]. The point residing half

[9]Line going through the centres of the camera lenses.

way between the centres of the camera lenses on the baseline is the *origin* of the coordinate system in which the resulting 3D coordinates are to be interpreted. As representative of the coordinate system the origin has been marked in the drawing.

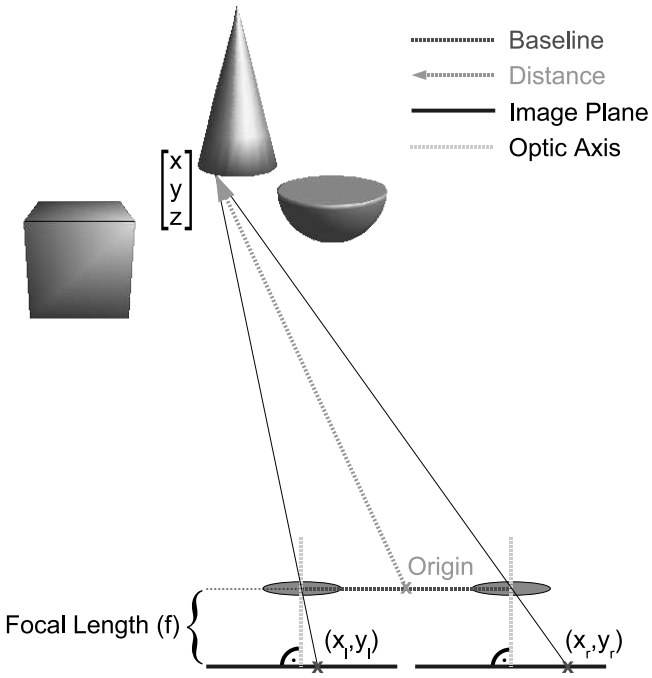


Figure 2.8: Camera Geometry Schema.

Object features are represented as pixels in the image planes. As already mentioned above the image planes can be projected into a common projection plane, where the x-distance between two corresponding spots, the disparity, can be measured. With assistance of triangulation methods the disparity information may be transformed into 3D coordinates.

2.1.4 3D Reconstruction

This section deals with the computation of 3D coordinates from a disparity map as described above. These cartesian coordinates can be casted into range information according to Pythagoras' theorem. Both representations carry identical information. Depending on the application at hand one or the other appears suitable. As the reconstruction from one given the other is evident only the mathematics behind the transformation of a disparity map into cartesian coordinates will be evolved step by step at this point.

In figure 2.9 the left camera of the stereo head is viewed from upside behind the image plane. Note that the point where the optic axis pierces the image plane is meant to be the origin of the image coordinate system.

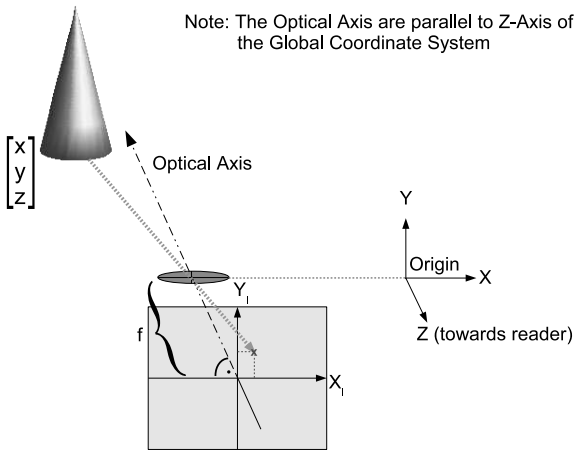


Figure 2.9: Left Camera - View from upside behind the Image Plane.

The global coordinate system into which the disparity data are to be transformed shall now be defined. Its X-Y-plane is chosen in a straight forward fassion to be parallel to the X-Y-planes spanned by the image coordinate systems of the cameras. The Z-axis of the global coordinate system will be pointing towards the image planes. As this axis is perpendicular to all the X-Y-planes defined so far, it is parallel to both optic axis (See Figure 2.9).

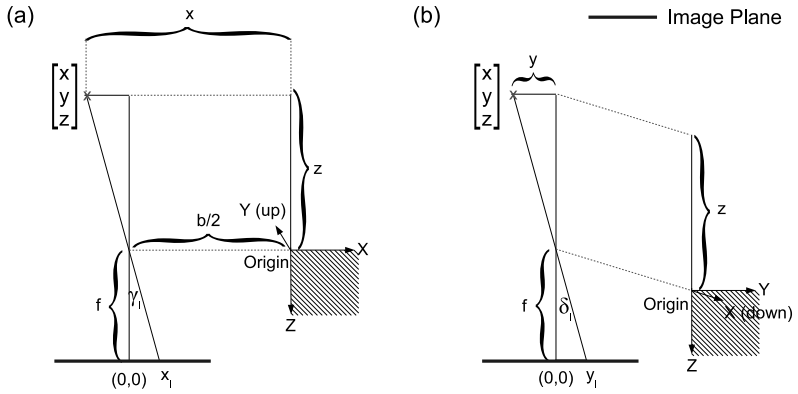


Figure 2.10: Left Camera - Pojection into X-Z- (a) and Y-Z-Plane (b).

With the help of two projections the assumptions and definitions made are shown to be sound in terms of disclosing a direct solution to the problem of 3D reconstruction. For the left camera these are exemplified in Figure 2.10. For the right camera the relations are the same besides that the origin of the global coordinate system is on the other side. Knowing that the *tangent* of an angle in a rectangular triangle equals the relation of *opposite leg* to *adjacent leg* and that the oposite angles formed by crossed lines are pair-wise identic, yields the following equations.

$$tan(\gamma_l) = \frac{x_l}{f} = \frac{x + \frac{b}{2}}{z} \tag{2.1}$$

$$tan(\gamma_r) = \frac{x_r}{f} = \frac{x - \frac{b}{2}}{z} \tag{2.2}$$

$$tan(\delta_l) = \frac{y_l}{f} = \frac{y}{z} = \frac{y_r}{f} = \tan(\delta_r) \tag{2.3}$$

Where f is the common focal length of the cameras and b the length of the baseline. Note that the left camera coordinate system resides in the negative and the right in the positive x-half of global coordinate system resepectively. This consideration explains the diference between Equations 2.1 and 2.2. Furthermore the reader should recall that the corresponding pixels are always on the same horizontal level. That way δ_l is equal to δ_r, which clarifies the emergence of Equation 2.3.

Derivation of Formula for x

Equations 2.1 and 2.2 can be solved for z and equated with one another.

$$\frac{(x + \frac{b}{2}) \cdot f}{x_l} = \frac{(x - \frac{b}{2}) \cdot f}{x_r} \tag{2.4}$$

$$(x + \frac{b}{2}) \cdot x_r = (x - \frac{b}{2}) \cdot x_l \tag{2.5}$$

$$x = \frac{\frac{b}{2} \cdot (x_r + x_l)}{x_l - x_r} \tag{2.6}$$

Derivation of Formula for y

Equations 2.1 and 2.2 can be solved for x and again be equated.

$$\frac{x_l \cdot z}{f} - \frac{b}{2} = x = \frac{x_r \cdot z}{f} + \frac{b}{2} \tag{2.7}$$

Furthermore Equation 2.3 can be solved for z using either the part including y_l or y_r. At this point y_l is chosen as representative to yield:

$$z = \frac{y \cdot f}{y_l} \tag{2.8}$$

Combining these two cancels out f and provides and evident solution for y as presented below.

$$y = \frac{b \cdot y_l}{x_l - x_r} \tag{2.9}$$

In the literature the fact that y_l equals y_r is often used in order to blow up this equation as to have a result which stronly resembles the formula for computing x. As this can lead to confusion it has been decided to show this second versions at this point.

$$y = \frac{\frac{b}{2} \cdot (y_l + y_r)}{x_l - x_r} \tag{2.10}$$

Derivation of Formula for z

For deducing the third formula Equation 2.7 is simply solved for z which unfolds as:

$$z = \frac{b \cdot f}{x_l - x_r} \tag{2.11}$$

With assistance of Equations 2.6, 2.9 (or 2.10) and 2.11 the disparity map can be transformed into a 3D-scatterplot. Note that the focal length f is needed to be provided in *pixel* rather than in mm. This value can usually be obtained from a geometric camera calibration procedure as described in Section 2.1.1. The values for x_l and x_r can be deduced from the disparity map. As the reconstruction is performed pixel-wise the x-coordinate of the disparity value in the map is one of these values and adding the disparity to this yields the second. Which is which depends on the sign of the disparity value. As the disparity is defined here to be $x_l - x_r$ it should usually be negative. Many disparity map generating algorithms see this the other way around. This has to be taken into account carefully.

3. Implementation and Results

In the scope of this project the focus has been on quickly getting operational a stereo vision head for outdoor purposes. The target platform is the mobile robot RAVON[1] (See Apendix A) developed in the Robotics Laboratory[2] at the Kaiserslautern Institute of Technology[3].

At this point many problems like the impact of daylight to camera images[4], vibration of the vehicle, etc. have not been regarded. These topics in themselves bare many pitfalls and need intense attention. It has been assumed that it will be possible to regulate shutter and aperture in a transparent and adaptive matter, such that image quality in a suitable range can be assured. Supplementary projects in these fields will be necessary if the system is really supposed to be in service outdoors. The project at hand is intended to provide the infrastructure for work in the stereo vision domain in general.

3.1 Base Components and Software used

This section is dedicated to the introduction of the hardware and the third party components utilised here.

As image sources a pair of Sony DFW-V500 (See Appendix B.2) and a pair of pointgrey Research dragonfly (See Appendix B.3) cameras have been available. First tests have been made using the dragonfly cameras but it soon became apparent that these cameras are not suitable for outdoor purposes as they lack important objective regulation options. The Sony cameras provide standard C-mount objective sockets which makes the use of a variety of different objectives possible.

The framegrabber written to access the cameras has been implemented on the basis of the firewire camera library libdc1394[5]. The disparity map generating function

[1] RAVON → Robust Autonomous Vehicle for Offroad Navigation
[2] Robotics Lab at (KIT): `http://agrosy.informatik.uni-kl.de/`
[3] Kaiserslautern Institute of Technology: `http://www.uni-kl.de`
[4] Light conditions change quickly due to clouds temporarily occluding the sun, etc.
[5] `http://libdc1394.sourceforge.net/`

used has been taken from the *Open Computer Vision Library*[6] just like some other image manipulation routines.

The implementation is based on the Modular Controller Architecture[7] which is generally used in the robotics laboratory at Kaiserslautern Institute of Technology.

3.2 System Structure

At this point a rough survey over the structure of the implemented system shall be given. Figure 3.1 shows the MCA diagram of the core components as well as the control and data flow.

The system has been devided into two MCA groups which run in different MCA parts. Additionally a part for the communication with MCAGUI (`mMCAGUIBridge`) has been implemented, as only the last part started can communicate with the group. This part has thus merely been implemented for technical reasons and just forwards the signals from MCAGUI to the appropriate parts and vice versa. In the following only the two principal groups will be subject to further exemplification.

The group `gStereoVision` encapsulates the framegrabber which is responsible for retrieving the images from the cameras. Additionally an image recording module has been implemented in order to be able to store images to the harddisk. That way it is possible to replace the `gStereoVision` component by an image loading facility for test purposes if the cameras are unavailable for some reason. Together with a database driven storage engine currently under development it will be possible to record and playback complete robot runs in terms of the sensor data received. The fundamental idea underlying these efforts is the automated testing of algorithms processing sensor data on a scientific basis. One requirement for this is that realistic scenarios have to be reproducable which is exactly the aim of the storage solution mentioned. The module `mImageRecorder` is one component which will later be using this framework.

The second group of the core system is named `gImageProcessing` and contains the components taking care of the range extraction from the images obtained from the framegrabber. In the following the main loop will be explained in more detail.

The Main Loop

Whenever the framegrabber has retrieved new images, they are stored in the blackboard *frame_bb* and the higher components are notified that new data is available via the MCA edge GRAB_CHANGED. The MCA group `gStereoFrameGrabber-Group` can be told to grab images continously in every sense cycle by passing the mode GRAB_CONTINOUSLY over the edge GRAB_MODE. Furthermore the user has the possibility to let the framegrabber retrieve grabs on request. This mode is set by passing a GRAB_ON_REQUEST signal via the edge GRAB_MODE. If in this mode, the framegrabber will fetch new images every time a request is placed over the MCA edge GRAB_REQUEST.

[6] Open Computer Vision Library → OpenCV:
Project Home Page: `http://www.intel.com/research/mrl/research/opencv/`
Obtain Source Distribution: `http://sourceforge.net/projects/opencvlibrary/`
[7] MCA → `http://mca2.sourceforge.net/`

On notification from the framegrabber, the part containing the group `gImageProcessing` retrieves the images from the blackboard, rectifies them using the intrinsic camera parameters previously determined by calibration, discharges the colour information and computes the disparity map using the MCA module `mDisparityMapGenerator`. For that purpose the environment may provide the module with the parameter *maximal disparity*. The undistorted images are stored into the blackboard *undistorted_images_bb* for diplay purposes. That way the quality of the calibration can roughly be checked visually. The disparity map is stored in the blackboard *disparity_map_bb* and the MCA module `mDispMapTo3DCoordinates` is called into service via the signal DISPARITY_MAP_CHANGED.

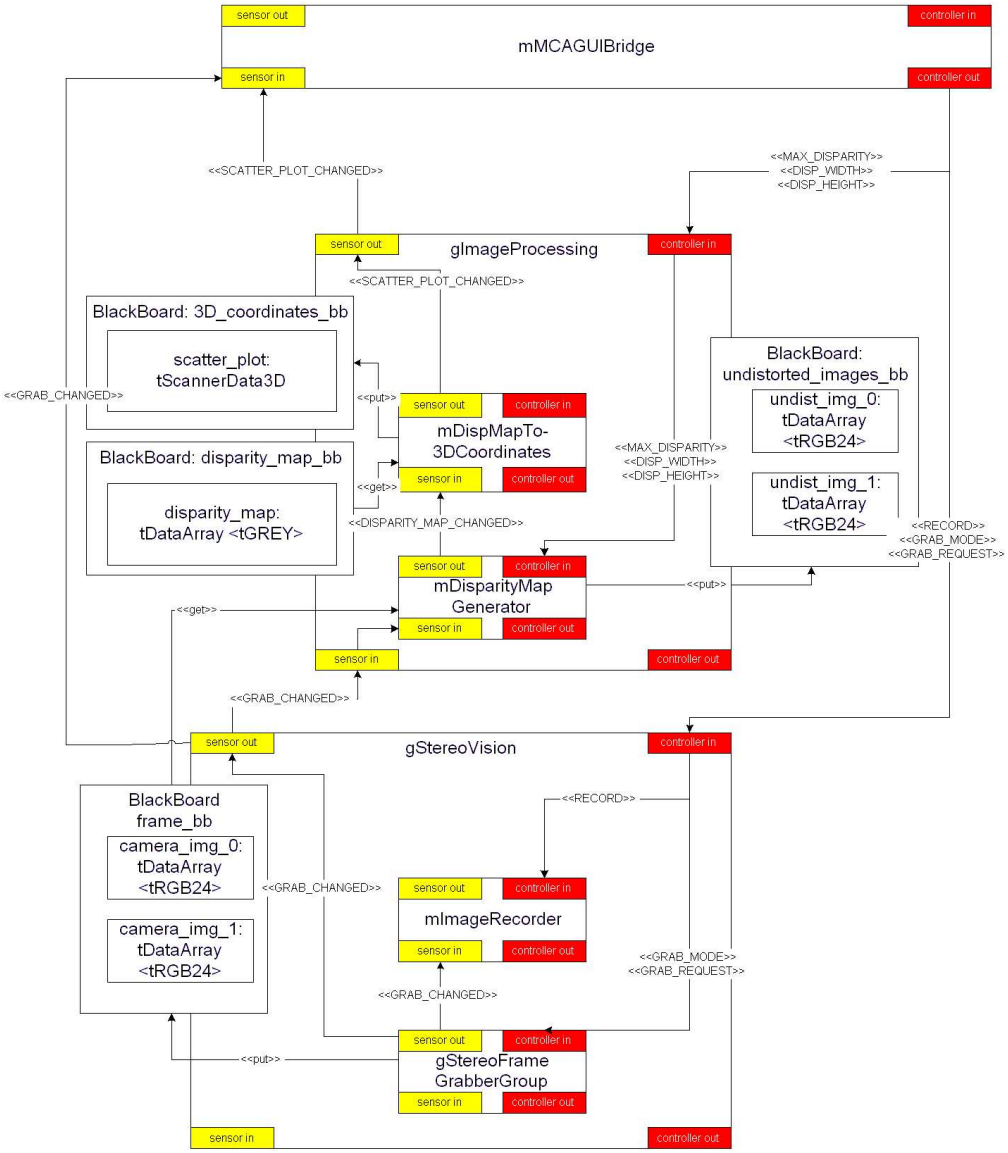


Figure 3.1: MCA Diagram of the implemented System.

At the end a 3D scatter plot is stored in the blackboard *3D_coordinates_bb* and the environment is notified that a new scatter plot is available via the MCA edge SCATTER_PLOT_CHANGED.

In order to speed up the disparity map generation it is possible to provide module mDisparityMapGenerator with a resolution lower than that of the source images. After the undistortion the images are resized and the disparity map is computed on the basis of these lower resolution images. After that the disparity map is scaled to the original size of the source images in order to have the correct relations for the 3D reconstruction.

This functionality has been implemented as compromise between performance and quality. In Section 3.4 it will become apparent that disparity map generation is not real-time competent with high resolutions. On the other hand geometric calibration can not be performed with very low resolutions because the corner extraction will become impossible.

3.3 Disparity Map Quality In- and Outdoors

This section is dedicated to the system testruns which have been performed in order to get an intuitive impression of the algorithm's work. Figure 3.2 recycles the disparity map already shown in Section 2.1.2.

Figure 3.2: Left (a) and Right (b) Image and the computed Disparity Map (c).

As already mentioned basic structures can be identified in the disparity map. Nevertheless strong horizontal artefacts cloud the good overall impression.

Furthermore thorough tests outdoors are still pending. At the moment it appears unrealistic to get satifactory results outdoors without additional work in the domain of colour constancy and image quality assurance in general. Figure 3.3 shows a typical outdoor scenario from the view of the developed stereo vision system with the generated disparity map.

Figure 3.3: Left (a) and Right (b) Image and the Disparity Map (c) outdoors.

These images have been taken late in the afternoon when the sun had constantly been hidden behind clouds. That way it has been possible to manually adjust the

aperture of the camera objectives. Figure 3.4 shows reflextion (a) and outshine (b) effects immanently at hand in outdoor scenarios. Due to rapidly changing light conditions outdoors it will be necessary to automatically regulate the aperture. This will be subject to further research.

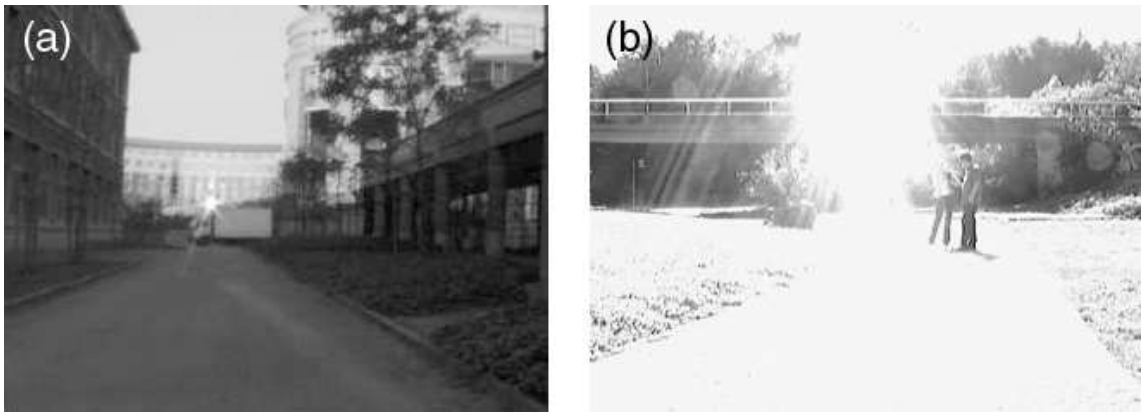


Figure 3.4: Reflexion (a) and Outshine (b) Effects Outdoors.

The question is, how much amelioration can be obtained with better calibration and what problems in principle result from the extraction algorithm chosen. Other approaches and matching algorithms should be tested in order to get an impression of the particular strengths in varying scenarios.

3.4 Runtime Tests

As the stereo vision system is supposed to be used in online navigation approaches, the update cycle of the scatter plot has to be in a tolerable range. Usually a component is said to be realtime capable if it can be computed in $30ms$ or less.

In order to get a feeling for what performance is realistic on the computers at hand it has been decided to make some test runs. Performance tests have shown, that most of the computational power is absorbed by the disparity map generating facility. Therefore detailed runtime tests have been performed only measuring the time needed for this part of the algorithm.

The implementation employed here has been taken from the *Open Computer Vision Library*[8] powered by Intel. The function classically accepts two greyscale images from which a greyscale disparity map is computed. Additionally the parameter *maximal disparity*, which represents the search width the algorithm will cover, has a severe impact on the runtime. All other parameters have been set to the standard values suggested in the documentation.

In the following the outcome of runtime tests will be shown, which vary in source image resolutions and the parameter *maximal disparity*. As already mentioned above the core problem of any disparity map generating algorithm is a matching problem. The algorithm at hand employs the *Birchfield measure* (See [Birchfield 98])

[8]Open Computer Vision Library → OpenCV:
Project Home Page: `http://www.intel.com/research/mrl/research/opencv/`
Obtain Source Distribution: `http://sourceforge.net/projects/opencvlibrary/`

for determining the pixel dissimilarities. The parameter maximal disparity states, up to what distance (in pixels) the algorithm shall search for corresponding spots in the source images. The maximal disparity has to be in the interval [0...min{255, source_image.width}]. The upper bound of 255 is a practical one. Since the target data structure for a disparity value is an eight bit integer, larger disparities can not be represented. The other boundary is reasonable because two corresponding spots can never be farther apart than the width of the source images. Otherwise the object would be beyond the border of at least one of the images. Table 3.1 shows the result of runtime tests carried out on one of our laboratory computers. For details on the configuration see Appendix B.1.

Max. Disp	Resolutions (horizontal x vertical)			
(in Pixels)	640 x 480	320 x 240	160 x 120	80 x 60
50	601,9 ms	156,0 ms	39,4 ms	9,0 ms
100	997,4 ms	269,3 ms	55,6 ms	9,8 ms
150	1.533,2 ms	355,0 ms	67,2 ms	
200	2.018,4 ms	385,9 ms		
255	2.269,1 ms	444,8 ms		

Table 3.1: Runtime Test Results I.

As can be seen, resolutions higher than 160 x 120 are unrealistic for realtime applications on today's mobile robot hardware. For a framerate of about 10 fps the mean calculation time needs to be lower than 100 ms, since the rest of the main loop will take some time as well. Therefore it has been decided to focus on the resolutions 160 x 120 and 80 x 60 in future projects.

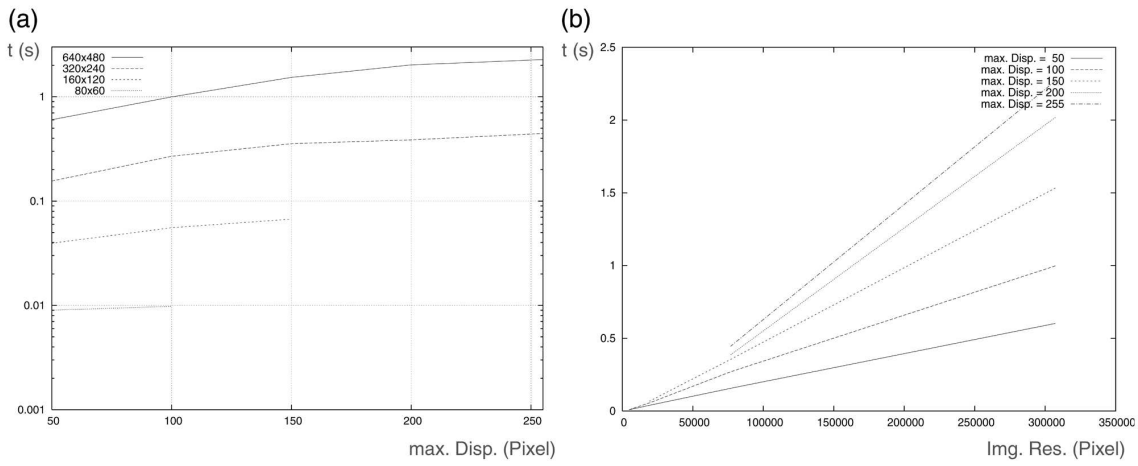


Figure 3.5: Runtime Test Results II.

Figure 3.5 shows the same result as a plot. In Figure 3.5(a) the maximum disparity is mapped to the x-axis and the computation time to the logarithmic y-axis. In this plot the almost exponential relationship between the maximal dispatity and the runtime of the algorith can be seen. Furthermore an almost linear relationship

between number of pixels per image and the runtime can be determined (See Figure 3.5 (b)[9]).

At the end of this work it has been possible to achieve a framerate of 8 fps on the computer mentioned in Appendix B.1, under the condition that the graphical user interface (MCAGUI) has been deactivated. The video widget currently incorporated in MCAGUI does not support overlay and is thus computationally very expensive. A CPU load of about 30 % has been determined using the unix command line tool *top*.

[9]Resolutions: 640 x 480 → 307200; 320 x 240 → 76800; 160 x 120 → 19200; 80 x 60 → 4800

4. Summary and Discussion

In the scope of this project the groundwork for 3D-vision-based operations has been set up. The technical problems of isochronically accessing two cameras[1] have been tackled and components for image interpretation have been glued together. The basic concepts of depth perception with cameras has been explained and hints as well as literature references have been given to the reader. Runtime conditions for one core algorithm have been presented in order to give a feeling for the complexity of such computations.

All in all the results are noteworthy taking into account how many difficulties are immanent in the problem of range data extraction from stereo images.

When employing the system outdoors additional problems may arise from vibrations. A few tests have been performed experimenting with shutter settings which have yielded promising results. The images appear to be astonishingly sharp. Outshine effects and other difficultly to tackle problems will have to be regarded intensily in further projects.

[1] i.e. read out both cameras at exactly the same time

A. Mobile Robot RAVON

The robot, for which the stereo camera system subject to this project has been developed, is based on the RobuCar TT platform (See Figure A.1) built by Robosoft[1].

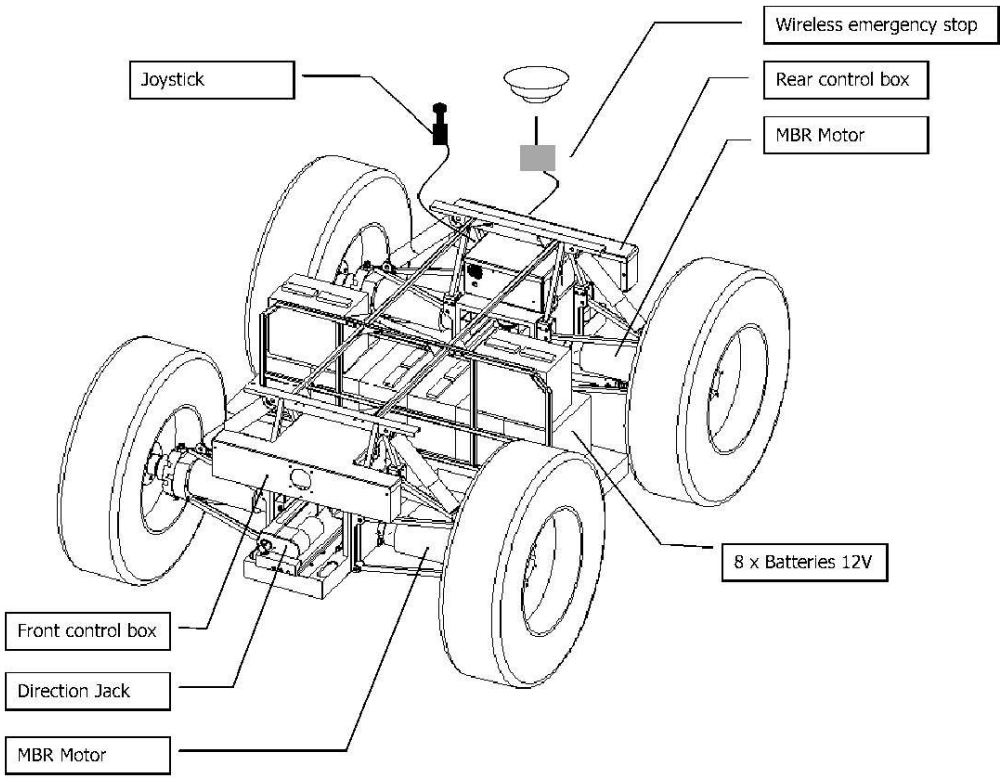


Figure A.1: Outdoor Platform RobuCar TT.

[1]http://www.robosoft.fr/

In the following the basic features of RobuCar TT are listed:

Length	2400 mm
Width	1400 mm
Height	800 mm
Weight	350 kg
Power Supply	8 sealed lead batteries (12 V, 60 Ah each)
Runtime	about 4 hours
Drive	four wheel drive with four independent electric motors
Steering	front and rear wheel steering via linear motors
Velocity	7 km/h max.
Controller	2 Motorola MPC 555
Computer	On-board PC
Floor Clearance	300 mm
Max. Slope	30 %

The extension developed at the Robotics Laboratory of the Kaiserslautern Institute of Technology is called RAVON[2].

The construction provides outdoor-tyres for maximal grip as well as a high-performance suspension system. The single wheel electric motors and all wheel steering capabilities allow advanced driving manoeuvres.

[2]RAVON → Robust Autonomous Vehicle for Offroad Navigation

B. Accessory Hardware

B.1 Runtime Test Computer Configuration

CPU:	Intel® Pentium® IV 2.8GHz
Cache:	512*kb*
Memory:	512*mb*
OS:	Gentoo Linux
Kernel:	2.4.24-rthal5
libdc1394:	Version 1.0.0 (`http://libdc1394.sourceforge.net/`)
OpenCV:	Version 0.9.6 (`http://www.intel.com/research/mrl/research/opencv/`)

B.2 SONY DFW V500

Interface:	IEEE 1394
Max. Speed:	400 Mbps
CCD:	Wfine CCD™ (Primary Color Filter, Progressive Scan, Square Pixels)
Type	CCD interline transfer, progressive scan
Size	1/3”
Format:	VGA (640 x 480), Non-Compressed YUV
Resolutions:	160 x 120 YUV(4:4:4)
	320 x 240 YUV(4:2:2)
	640 x 480 YUV(4:2:2)
Max. Framerate:	30 fps
Shutter (rmt)	1/20,000 to 2 s
Gain (rmt)	0 to 18 dB

The manual with complete specifications can be downloaded from the internet at:
`http://bssc.sel.sony.com/Professional/docs/brochures/is-1094.pdf`

B.3 pointgrey Research Dragonfly™

Interface:	IEEE 1394
Max. Speed:	400 Mbps
CCD:	Sony ICX424
Type	CCD interline transfer, progressive scan
Size	1/3”
Format:	VGA (640 x 480), Non-Compressed YUV Format 7
Max. Framerate:	30 fps

Detailed technical information can be obtained at `http://www.ptgrey.com/products/dragonfly/`

List of Figures

Bibliography

[Birchfield 98] Stan Birchfield, Carlo Tomasi. A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling. In *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 1998.

[Birchfield 99] S. Birchfield, C. Tomasi. Depth Discontinuities by Pixel-to-Pixel Stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.

[Debenest 03] Paulo Debenest, Edwardo F. Fukushima, Shigeo Hirose. Proposal for Automation of Humanitarian Demining with Buggy Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[Forsyth 03] David A. Forsyth, Jean Ponce. *Computer Vision: A modern Approach.* Pearson, 2003.

[Gantenbrink 99] Rudolf Gantenbrink, Stephen Belless. The UPUAUT Project - A Robot for Pyramid Exploration. http://www.cheops.org, 01 1999.

[Hong 02] Tsai-Hong Hong, Christopher Rasmussen, Tommy Chang, Michael Shneier. Fusing Ladar and Color Image Information for Mobile Robot Feature Detection and Tracking. In *7th International Conference on Intelligent Autonomous Systems*, 2002.

[Horn 01] Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, 2001.

[Iagnemma 03] Karl Iagnemma, Shinwoo Kang, Christopher Brooks, Steven Dubowsky. Multi-Sensor Terrain Estimation for Planetary Rovers. In *7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2003.

[Lenain 03] R. Lenain, B. Thuilot, C. Cariou, P. Martinet. Adaptive control for car like vehicles guidance relying in RTK GPS: rejection of sliding effects in agricultural applications. In *IEEE International Conference on Robotics and Automation*, 2003.

[Lotti 97] J.-L. Lotti, Y. Yvinec. Projet Stéréovision. Technischer Bericht, Ecole Royale Militaire, 1997.

[Porikli 03] Fatih Porikli. INTER-CAMERA COLOR CALIBRATION USING CROSS-CORRELATION MODEL FUNCTION. *CISS*, 2003.

[Schenker 03] P. Schenker, T. Huntsberger, P. Pirjanian, S. Dubowsky, K. Iagnemma, V. Sujan. Rovers for Intelligent, Agile Traverse of Challenging Terrain. In *International Conference on Advanced Robotics*, 2003.

[Thuilot 01] B. Thuilot, C. Cariou, L. Cordesses, P. Martinet. Automatic guidance of a farm tractor along curved paths, using a unique CP-DGPS. In *IEEE/RSJ International Conference on Robots and Systems*, 2001.

[Wellington 04] Carl Wellington, Anthony Stentz. Online Adaptive Rough-Terrain Navigation in Vegetation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

[Zhang 99] Zhengyou Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. *ICCV*, 1999.