# Vertical world feature detection and mapping using stereo vision and accelerometers

**Jorge Lobo** ⋆

Institute of Systems and Robotics
Electrical Engineering Department
University of Coimbra
3030 Coimbra
Portugal

**Carlos Queiroz** ⋆

Informatics Engineering Department
Polytechnic Institute of Tomar
Quinta do Contador
2300 Tomar
Portugal

**Jorge Dias** ⋆

Institute of Systems and Robotics
Electrical Engineering Department
University of Coimbra
3030 Coimbra
Portugal

**Abstract.** This paper explores the integration of inertial sensor data with vision. A method is proposed for vertical world feature detection and map building. Visual and inertial sensing are two sensory modalities that can be explored to give robust solutions on image segmentation and recovering of 3D structure from images. This enlarges the application potential of vision systems. From the inertial sensors and a few stereo vision system parameters we can recover the horizon, vertical and ground plane. The collienation of image ground points can be found. By detecting the vertical line segments in each image, and using the collineation of ground points for the *foot* of each segment, the lines can be matched and recovered in 3D. The mobile robot using this vision system can than map the detected vertical features in a world map as it moves.

## 1 Introduction

The most common representations of the space in mobile robotics can be grouped in two classes: *metric maps*, and *topological maps*. A *metric map* represents the environment according to the absolute geometric position of the objects. A *topological map* is an abstract representation that describes relationships among features of the environment, without any absolute reference system. In this paper we explore the metric approach since it is more suitable for the development of map-building techniques using different sensor modalities.

Since map building is one fundamental problem in mobile robotics, different approaches have been purposed. The Occupancy Grid map, first introduced by Moravec and Elfes [1], has been widely used [2], [3]. In our previous work [4] we used occupancy grids to represent the environment as a two dimensional array of cells, each of which indicates the probability of being occupied. This type of maps is especially suitable for imprecise range sensors like sonar rangefinders, but not for other sensors like laser range finders or depth from stereo vision. Geometric feature for map building has been explored by many researchers [5][6].

Inertial sensors explore intrinsic properties of body motion, and can be useful for map building. From the principle of generalised relativity of Einstein we known that only the specific force on one point and the angular instantaneous velocity, but no other quantity concerning motion and orientation with respect to the rest of the universe, can be measured from physical experiments inside an isolated closed system. Therefore from inertial measurements one can only determine an estimate for linear accelerations and angular velocities. Linear velocity and position, and angular position, can be obtained by integration. Inertial navigation systems implement this process of obtaining velocity and position information from inertial sensor measurements. In the paper, we show how a metric map can be built from geometric information extracted from data obtained by the combination of inertial sensing and image data. Using these two sensor's modalities we extract physical properties, such as verticality of lines. The use of inertial and visual cues for autonomous robot navigation has already been proposed by Viéville and Faugeras [8][9].

The rest of this paper is arranged as follows. Section 2 describes the system's geometric model. The geometric features extracted from the inertial sensor data, the vertical, the horizon and the collineation of ground plane image points, are discussed in section 3. Section 4 discusses a method for detection of vertical image lines. In section 5, we describe how the correspondence problem between image features is solved. The experimental setup used and some results are presented in Section 6.

⋆ {jlobo,carloqz,jorge}@isr.uc.pt

## 2 System geometry and model

The proposed system has an inertial unit at the middle of the stereo camera baseline, as seen in figure 1. The cameras' pan is controlled so as to have a symmetric vergence. The system's coordinate frame referential, $\{\mathcal{C}\}$ named *Cyclop*, is defined as having the origin at the center of the baseline of the stereo cameras, as seen in figure 2.

Fig. 1. System on Mobile Robot

Each camera has its own referential, $\{\mathcal{R}\}$ and $\{\mathcal{L}\}$. Besides being translated from $\{\mathcal{C}\}$ along the baseline ($\boldsymbol{T_r}$ and $\boldsymbol{T_l}$), and rotated $\theta_R$ and $\theta_L$ along the $\{\mathcal{C}\}$ Z-axis ($\boldsymbol{R_r}$ and $\boldsymbol{R_l}$), their axis are also swapped to comply with the typical referential convention used for camera images ($\boldsymbol{S_r}$ and $\boldsymbol{S_l}$) - see figure 2. Notice that in our case we have symmetric vergence, i.e. $\theta = \theta_R = -\theta_L$.

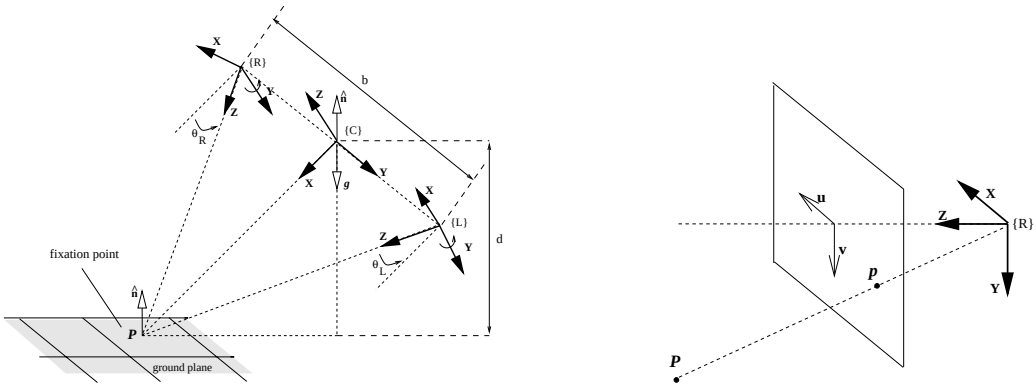



Fig. 2. a) System Geometry; b) The camera referential and image coordinates.

To express a world point $\boldsymbol{P}$, given in the camera referential, on the Cyclop referential $\{\mathcal{C}\}$ we have

$$ {}^{\mathcal{C}}\boldsymbol{P} = \boldsymbol{T_r}.\boldsymbol{R_r}.\boldsymbol{S_r}.{}^{\mathcal{R}}\boldsymbol{P} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{R}}.{}^{\mathcal{R}}\boldsymbol{P} \qquad {}^{\mathcal{C}}\boldsymbol{P} = \boldsymbol{T_l}.\boldsymbol{R_l}.\boldsymbol{S_l}.{}^{\mathcal{L}}\boldsymbol{P} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{L}}.{}^{\mathcal{L}}\boldsymbol{P} \tag{1} $$

where

$$ \boldsymbol{S_r} = \boldsymbol{S_L} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{T_r} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\frac{b}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{T_l} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{b}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{R_r} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{R_l} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

and $b$ is the baseline distance.

The pinhole camera model is used for the image formation process. Assuming that image acquisition maintains square pixel ratio and no skew we have for an image point the following projective mapping

$$s\boldsymbol{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \boldsymbol{C}\begin{bmatrix}\boldsymbol{I}\ 0\end{bmatrix}\boldsymbol{P} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

where $u$ and $v$ are the pixel coordinates with origin at the image centre, $f$ is the camera focal distance and scale factor and $\boldsymbol{P} = (X, Y, Z)^t$ is in the camera referential, and the image point is defined as a projective point, up to a scale factor $s$.

If the 3D point $\boldsymbol{P} = (X, Y, Z)^{\mathsf{T}}$ was not given in the camera's frame of reference, $\begin{bmatrix}\boldsymbol{I}\ 0\end{bmatrix}$ in the above equation would become $\begin{bmatrix}\boldsymbol{R}\ \boldsymbol{t}\end{bmatrix}$ to take into account the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ to the cameras frame of reference, i.e. the camera's extrinsic parameters. Matrix $\boldsymbol{C}$ represents a simplified model of the camera's intrinsic parameters.

The scale factor is arbitrary, and reflects the fact that only the projective ray for each image point is know, and the image plane can be scaled by any non zero $s$. Since only the orientation of the projective ray is known, any representation of that orientation is valid.

## 3 Gravity vector gives image horizon, vertical and ground plane

An inertial system prototype built at our lab was used to provide the inertial data. The sensors used in the prototype system include a three-axial accelerometer, three gyroscopes and a dual-axis inclinometer. The inertial system prototype was mounted onto the active vision system as seen in figure 1.

### 3.1 Gravity vector

The measurements $\boldsymbol{a}$ taken by the inertial unit's accelerometers include the sensed gravity vector $\boldsymbol{g}$ summed with the body's acceleration $\boldsymbol{a}_b$:

$$\boldsymbol{a} = \boldsymbol{g} + \boldsymbol{a}_b \tag{4}$$

Assuming the system is motionless, then $\boldsymbol{a}_b = 0$ and the measured acceleration $\boldsymbol{a} = \boldsymbol{g}$ gives the gravity vector in the system's referential. So, with $a_x$,$a_y$ and $a_z$ being the accelerometer filtered measurements along each axis, the vertical unit vector will be given by

$$\hat{\boldsymbol{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = -\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} = \frac{1}{\sqrt{a_x^2+a_y^2+a_z^2}}\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \qquad \text{and} \qquad {}^{\mathcal{C}}\hat{\boldsymbol{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ 1 \end{bmatrix} \tag{5}$$

where ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ is the homogeneous normal vector given in the Cyclop frame of reference.

Notice that if our assumption of the system being motionless or subject to constant speed is correct, than in the above equation

$$\sqrt{a_x^2 + a_y^2 + a_z^2} = 9.8ms^2 \tag{6}$$

and this condition can be tested and monitored by the system.

### 3.2 Vertical

In equation (5) the vertical unit vector is given in the Cyclop referential. The vertical for each of the cameras is given by

$${}^{\mathcal{R}}\hat{\boldsymbol{n}} = \boldsymbol{S}_{\boldsymbol{r}}^{-1}.\boldsymbol{R}_{\boldsymbol{r}}^{-1}.{}^{\mathcal{C}}\hat{\boldsymbol{n}} \qquad \text{and} \qquad {}^{\mathcal{L}}\hat{\boldsymbol{n}} = \boldsymbol{S}_{\boldsymbol{l}}^{-1}.\boldsymbol{R}_{\boldsymbol{l}}^{-1}.{}^{\mathcal{C}}\hat{\boldsymbol{n}} \tag{7}$$

### 3.3 Horizon

In an image the horizon can be found by having two distinct vanishing points [10]. With a suitable calibration target (e.g. a levelled square with well defined edges) the horizon can be determined.

Knowing the vertical in the cameras referential and the focal distance, an artificial horizon can be traced. A planar surface with a unit normal vector $\hat{\boldsymbol{n}}$, not parallel to the image plane has, when projected, a *vanishing line* given by

$$n_x u + n_y v + n_z f = 0 \tag{8}$$

where $f$ is the focal distance, $u$ and $v$ image coordinates and $\hat{\boldsymbol{n}} = (n_x, n_y, n_z)^t$

Since the vanishing line is determined alone by the orientation of the planar surface, then the projections of planar surfaces parallel in the scene define a common vanishing line. The horizon is the vanishing line of all levelled planes, parallel to the ground plane.

With one vanishing point, obtained from two parallel lines belonging to some levelled plane, and from equation (8) the unknown scaling factor $f$ in equation (3) can be estimated [7][11]. The quality of the estimation of $f$ depends on the noise level in the accelerometer data. Nevertheless it provides a reasonable estimate for a completely uncalibrated camera.

## 4 Ground plane

Consider a world point ${}^{\mathcal{C}}\boldsymbol{P}$ that belongs to the ground plane. The plane equation is given by

$${}^{\mathcal{C}}\hat{\boldsymbol{n}}.{}^{\mathcal{C}}\boldsymbol{P} + d = 0 \tag{9}$$

where $d$ is the distance from the origin to the ground plane, i.e. the system height. In some applications it can be know or imposed by the physical mount, but it can be determined by the stereo system through a process of visual fixation [12] [13].

If the vision system fixates in a point ${}^{\mathcal{C}}\boldsymbol{P}_f$ that belongs to the ground plane, and assuming symmetric vergence (i.e. $\theta = \theta_R = -\theta_L$), the distance $d$ is given by the projection of ${}^{\mathcal{C}}\boldsymbol{P}_f$ on the gravity vector direction

$$d = -{}^{\mathcal{C}}\hat{\boldsymbol{n}}.{}^{\mathcal{C}}\boldsymbol{P}_f = -{}^{\mathcal{C}}\hat{\boldsymbol{n}}.\begin{bmatrix} \frac{b}{2}\cot\theta \\ 0 \\ 0 \\ 1 \end{bmatrix} = -n_x\frac{b}{2}\cot\theta \tag{10}$$

as can easily be seen in figure 2. In this figure there is no lateral inclination, but 10 is valid for any angle, since the attitude is given by ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$. This value of $d$ will be used to determine if other points in the image belong or not to the level plane containing the image fixation point.

## 5 Robot navigation frame of reference

With the system mounted on the mobile robot, all detected features can be charted onto the robot's world map, constructed as the robot moves in its environment. It is not convenient to construct this map in the Cyclop $\{\mathcal{C}\}$ referential, and a better choice is to convert the points to a robot navigation frame of reference $\{\mathcal{N}\}$. The vertical unit vector $\hat{\boldsymbol{n}}$ and system heigh $d$ can be used to define $\{\mathcal{N}\}$, by choosing ${}^{\mathcal{N}}\hat{\boldsymbol{x}}$ to be coplanar with ${}^{\mathcal{C}}\hat{\boldsymbol{x}}$ and ${}^{\mathcal{C}}\hat{\boldsymbol{n}}$ in order to keep the same heading, we have

$${}^{\mathcal{N}}\boldsymbol{P} = {}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}\boldsymbol{P} \tag{11}$$

where

$${}^{\mathcal{N}}\boldsymbol{T}_{\mathcal{C}} = \begin{bmatrix} \sqrt{1-n_x^2} & \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{-n_x n_z}{\sqrt{1-n_x^2}} & 0 \\ 0 & \frac{n_z}{\sqrt{1-n_x^2}} & \frac{-n_y}{\sqrt{1-n_x^2}} & 0 \\ n_x & n_y & n_z & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}} = \begin{bmatrix} \sqrt{1-n_x^2} & 0 & n_x & -n_x d \\ \frac{-n_x n_y}{\sqrt{1-n_x^2}} & \frac{n_z}{\sqrt{1-n_x^2}} & n_y & -n_y d \\ \frac{-n_x n_z}{\sqrt{1-n_x^2}} & \frac{-n_y}{\sqrt{1-n_x^2}} & n_z & -n_z d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

The update of $\{\mathcal{N}\}$ as the robot moves along is beyond the scope of this work, where $\{\mathcal{N}\}$ is just relative to the robot's position, and not truly world fixed. Using the robot's odometry, the inertial sensors and landmark matching this update can be accomplished.

## 6 Collineation of ground plane points

Consider a world point ${}^{\mathcal{N}}\boldsymbol{P}$ that belongs to the ground plane (i.e. $Z = 0$). We can rewrite the plane equation in (9) as

$$\begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} .{}^{\mathcal{N}}\boldsymbol{P} = 0 \tag{13}$$

and we can express the ground point as

$${}^{\mathcal{N}}\boldsymbol{P}_{gnd} = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{14}$$

To find its projection in the camera image plane we can rewrite equation (3) as

$$s\boldsymbol{p}_i = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{R} \, \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{r_1} \, \boldsymbol{r_2} \, \boldsymbol{r_3} \, \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{r_1} \, \boldsymbol{r_2} \, \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{15}$$

From the above equation we can see that there is a fixed mapping between ground plane points and image points. This mapping is called an homography or collineation of points. A ground plane point is related to the right camera image by an homography $\boldsymbol{H_r}$:

$$s\boldsymbol{p}_{ri} = \boldsymbol{H_r}.\widetilde{\boldsymbol{P}} \tag{16}$$

where $\boldsymbol{p}_{ri}$ is the right projective image point, $s$ an arbitrary scale factor, $\widetilde{\boldsymbol{P}} = \begin{bmatrix} X & Y & 1 \end{bmatrix}^{\mathsf{T}}$ and

$$\boldsymbol{H_r} = \boldsymbol{C_R} \begin{bmatrix} \boldsymbol{r_1} \, \boldsymbol{r_2} \, \boldsymbol{t} \end{bmatrix}_R \tag{17}$$

and similarly we have for the left image

$$s\boldsymbol{p}_{li} = \boldsymbol{H_l}.\widetilde{\boldsymbol{P}} \tag{18}$$

with

$$\boldsymbol{H_l} = \boldsymbol{C_L} \begin{bmatrix} \boldsymbol{r_1} \, \boldsymbol{r_2} \, \boldsymbol{t} \end{bmatrix}_L \tag{19}$$

We can consider a direct mapping $\boldsymbol{H}$ of ground plane points between the stereo pair. $\boldsymbol{H}$ can be obtained by calibration using know ground plane points [14], or using equations (16) and (18), provided that $\boldsymbol{H_r}$ or $\boldsymbol{H_l}$ are invertible. For the direct mapping $\boldsymbol{H}$ of right image points to the left image we have

$$s\boldsymbol{p}_{li} = \boldsymbol{H}.\boldsymbol{p}_{ri} = \boldsymbol{H_l}.\boldsymbol{H_r}^{-1}.\boldsymbol{p}_{ri} \tag{20}$$

To obtain $\boldsymbol{H}$ we must first compute $\boldsymbol{H_l}$ and $\boldsymbol{H_r}$. From (19) and using ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}}$ obtained from the inertial data and ${}^{\mathcal{L}}\boldsymbol{T}_{\mathcal{C}} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{L}}{}^{-1}$ obtained form the geometric setup, we can write

$$\boldsymbol{H_l} = \boldsymbol{C_L} \begin{bmatrix} \boldsymbol{r_1} \, \boldsymbol{r_2} \, \boldsymbol{t} \end{bmatrix}_L = \boldsymbol{C_L} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} .{}^{\mathcal{L}}\boldsymbol{T}_{\mathcal{C}}.{}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{N}}. \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

and from the previous equations we obtain

$$\boldsymbol{H_l} = \begin{bmatrix} f\frac{-\sin\theta + n_x^2 \sin\theta + n_x n_y \cos\theta}{\sqrt{1-n_x^2}} & -f\frac{n_z \cos\theta}{\sqrt{1-n_x^2}} & fn_x d\sin\theta + fn_y d\cos\theta + \frac{1}{2}fb\cos\theta \\ f\frac{n_x n_z}{\sqrt{1-n_x^2}} & f\frac{n_y}{\sqrt{1-n_x^2}} & fn_z d \\ \frac{\cos\theta - n_x^2\cos\theta + n_x n_y \sin\theta}{\sqrt{1-n_x^2}} & -\frac{n_z \sin\theta}{\sqrt{1-n_x^2}} & -n_x d\cos\theta + n_y d\sin\theta + \frac{1}{2}b\sin\theta \end{bmatrix} \tag{22}$$

and proceeding analogously for the right image we obtain

$$\boldsymbol{H_r} = \begin{bmatrix} -f\frac{-\sin\theta+n_x^2\sin\theta-n_xn_y\cos\theta}{\sqrt{1-n_x^2}} & -f\frac{n_z\cos\theta}{\sqrt{1-n_x^2}} & -fn_xd\sin\theta+fn_yd\cos\theta-\frac{1}{2}fb\cos\theta \\ f\frac{n_xn_z}{\sqrt{1-n_x^2}} & f\frac{n_y}{\sqrt{1-n_x^2}} & fn_zd \\ -\frac{-\cos\theta+n_x^2\cos\theta+n_xn_y\sin\theta}{\sqrt{1-n_x^2}} & \frac{n_z\sin\theta}{\sqrt{1-n_x^2}} & -n_xd\cos\theta-n_yd\sin\theta+\frac{1}{2}b\sin\theta \end{bmatrix} \tag{23}$$

From (20) we have that the homography between left and right images of ground plane points, for the geometry established for our system, is given by

$$\boldsymbol{H} = \boldsymbol{H_l}.\boldsymbol{H_r}^{-1} = \begin{bmatrix} -\frac{2n_xb\cos\theta\sin\theta+n_yb+2d-4d\cos^2\theta}{-n_yb+2d} & \frac{2bn_z\cos\theta}{-n_yb+2d} & f\frac{-2\cos\theta(2d\sin\theta+bn_x\cos\theta)}{-n_yb+2d} \\ 0 & 1 & 0 \\ 2\frac{2d\sin\theta\cos\theta-n_xb+n_xb\cos^2\theta}{f(-n_yb+2d)} & \frac{2bn_z\sin\theta}{f(-n_yb+2d)} & -\frac{2n_xb\cos\theta\sin\theta+n_yb-4d\cos^2\theta+2d}{-n_yb+2d} \end{bmatrix} \tag{24}$$

This equation will be fundamental for the stereo correspondence described in section 8.

## 7 Image vertical line detection

Knowing the vertical, the vanishing point of all image lines that correspond to world vertical features is know. This vanishing point is at infinity when there is no tilt, and the vertical lines are all parallel in the image. For small tilt values, the vertical lines can be taken as parallel. Based on this assumption, the vertical line segments found in the image will be parallel to the local image vertical $\hat{\boldsymbol{n}}_i$, the normalized image projection of the vertical $\hat{\boldsymbol{n}}$ as defined in equation (7).

In order to detect vertical lines we extract the edges in the image using an optimized Sobel filter [15]. The filter estimates the gradient $\boldsymbol{\mathcal{D}}$ as

$$\boldsymbol{\mathcal{D}} = \begin{bmatrix} \mathcal{D}_x \\ \mathcal{D}_y \end{bmatrix} \qquad \text{where} \qquad \mathcal{D}_x \approx \tfrac{1}{32}\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \qquad \text{and} \qquad \mathcal{D}_y \approx \tfrac{1}{32}\begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \tag{25}$$

The optimized Sobel filter has a lower angle error than the standard Sobel filter. By choosing an appropriate threshold for the gradient magnitude, the potential edge lines can be identified.

$$\mathcal{D} = \sqrt{\mathcal{D}_x^2 + \mathcal{D}_y^2} > \mathit{threshold} \tag{26}$$

To only obtain the vertical edges we compare the pixel gradient with the vertical. The dot product of the gradient with a vector normal to the vertical should be null, so by setting a tolerance threshold value the detected edge points can be taken as vertical or not.

$$\boldsymbol{\mathcal{D}}.\hat{\boldsymbol{h}}_i < \mathit{tolerance} \tag{27}$$

where $\hat{\boldsymbol{h}}$ is the horizontal unit vector, perpendicular to $\hat{\boldsymbol{n}}$ in the image plane. i.e.

$$\hat{\boldsymbol{n}}_i.\hat{\boldsymbol{h}}_i = 0 \tag{28}$$

But since this can lead to erroneous results since the pixel gradient provides a very local information and is affected by the pixel quantization, a large tolerance is used. In order to extract the vertical lines in the image, all edge points that satisfied equation (27) where mapped to a rectified image table (equation (29)), so that continuity could be tested along the vertical edge direction. So each edge point $\boldsymbol{p}_i = (u, v)$ contributed to the table at position

$$\mathit{vert\_points}\,(x, y) = \left(\boldsymbol{p}_i.\hat{\boldsymbol{h}}, \boldsymbol{p}_i.\hat{\boldsymbol{n}}\right) \tag{29}$$

The minimum line length and allowable gaps is set and each column of the table parsed. The end result is a set of lines, given by their end-points in the original image.

Figure 3 shows the results obtained. The white lines show the extracted vertical edges and the dark line the horizon.

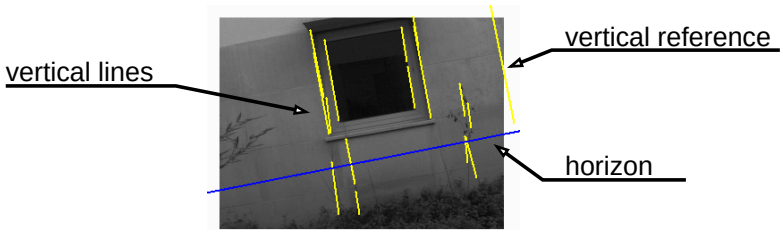



Fig. 3. Detected vertical lines.

## 8 Stereo correspondence of vertical lines and 3-D position

In the previous section a method was proposed for vertical image lines detection. But in order to find out if they truly correspond 3D vertical features they have to be matched across the stereo pair of images. Making the assumption that the relevant vertical features start from the ground plane, and since we known the homography of the ground plane image points from equation (24) a common unique point is identified, the lower point or *foot* of each vertical feature in one image should map to the corresponding *foot* in the other image.

An algorithm for the 3D reconstruction of vertical features can now be presented. For each detected line in the right image, map its *foot* to the other image using equation (24). The correspondent point and its neighbourhood in the left image can then be tested for a match with the original point of interest in the right image by parsing all the left image lines and testing

$$\boldsymbol{p}_{li} \pm \delta = \boldsymbol{H}.\boldsymbol{p}_{ri} \tag{30}$$

where $\delta$ is the allowed neighbourhood window, $\boldsymbol{p}_{li}$ the *foot* of the left image lines being parsed and $\boldsymbol{p}_{ri}$ the *foot* of the right image line.

If there is a match, the point belongs to the ground plane and must be the *foot* of a true 3D vertical world feature. From equation 16 the 3D position ${}^{\mathcal{N}}\boldsymbol{P}$ of the *foot* of this vertical element is given by

$${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \boldsymbol{H}_{\boldsymbol{r}}^{-1}\boldsymbol{p}_{ri} \tag{31}$$

where ${}^{\mathcal{N}}\widetilde{\boldsymbol{P}} = \begin{bmatrix} X & Y & 1 \end{bmatrix}$ and ${}^{\mathcal{N}}\boldsymbol{P} = \begin{bmatrix} X & Y & 0 & 1 \end{bmatrix}$.

With the system mounted on the mobile robot, the vertical features can be charted on a world map, constructed as the robot moves in its environment. This map is constructed in the robot's navigation frame of reference $\{\mathcal{N}\}$ -see section 5.

## 9 Results

### 9.1 Experimental Setup

For this work we needed a pair of cameras with a stereo rig capable of controlling camera vergence, and inertial sensors to measure the systems attitude.

An inertial system prototype built at our lab was used. The system is based on low-cost inertial sensors and is intended for robotic applications. The sensors used in the prototype system include a three-axial accelerometer, three gyroscopes and a dual-axis inclinometer.

For this work we are only extracting the system's attitude from the accelerometer data when it is motionless. This is done by keeping track of the gravity vector $\boldsymbol{g}$. The gyros are intended to track all rotations, so that the attitude can be known when accelerations other than gravity are present [16]. The accelerometer sensor, 34103A tri-axial capacitive accelerometer by Summit Instruments, comes with a factory calibration table, providing sensitivity, offset and alignment data. See [17] for complete details of the inertial system prototype.

The system uses two commercial pan&tilt stepper motor units, built by Directed Perception, onto which two Sony XC-999 NTSC cameras, with 12 mm lens, were mounted. The pan&tilt units are used only for panning, enabling the system to verge with a common base line. So apart from an initial reset and alignment,

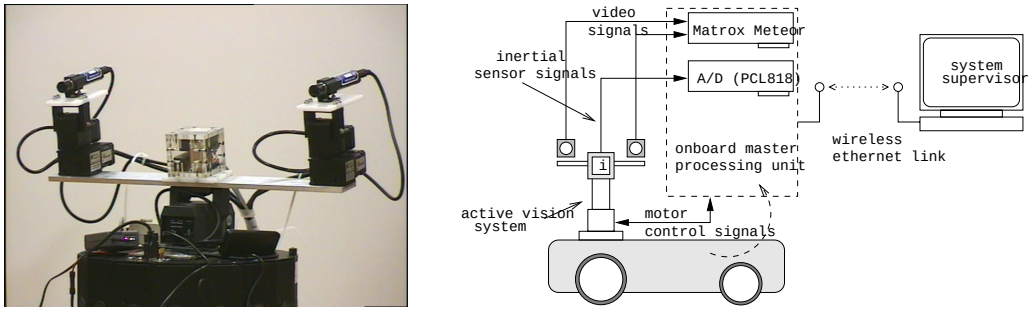



Fig. 4. a) Vision System with Inertial Sensors; b) System Architecture.

only the pan will be used. The cameras are positioned with their optical center over the panning axis to ensure a constant common baseline.

The inertial system prototype was mounted onto the active vision system as seen in figure 4. A large baseline of 50*cm* had to be used due to the initialization on the pan&tilt units, that forces a full swing motion of the motors. By having a large baseline, the robustness of the stereo algorithms is also improved. The vergence resolution is 0.05413 degrees per step.

The whole systems runs on a PC running a public domain Unix operating system, the FreeBSD. An Advantech PCL818HD data acquisition card was included to handle the inertial sensor acquisition. A Matrox Meteor framegrabber was used to capture the images from the cameras. The FreeBSD driver for the Meteor was reprogrammed to handle the Advantech board and enable sensor data acquisition synchronous with the captured frames. The pan&tilt units connect to the PC's RS232. This system, while not enabling top performances, is very flexible and reconfigurable.

To study the integration of inertial information and vision in artificial autonomous mobile systems, the system was mounted onto a mobile robot platform. Figure 4 shows the architecture of the complete system. The the pitch and roll can be adjusted in the mechanical mount, but are not actively controlled. To accomplish the visual fixation of a ground plane point, the target point, pitch and roll are adjusted as required.

### 9.2 Vertical world feature detection

With this experimental setup, the vertical world feature detection algorithm was tested. Figure 5 shows one stereo image pair with the identified image vertical lines and the matched *feet* of the true 3D vertical lines.

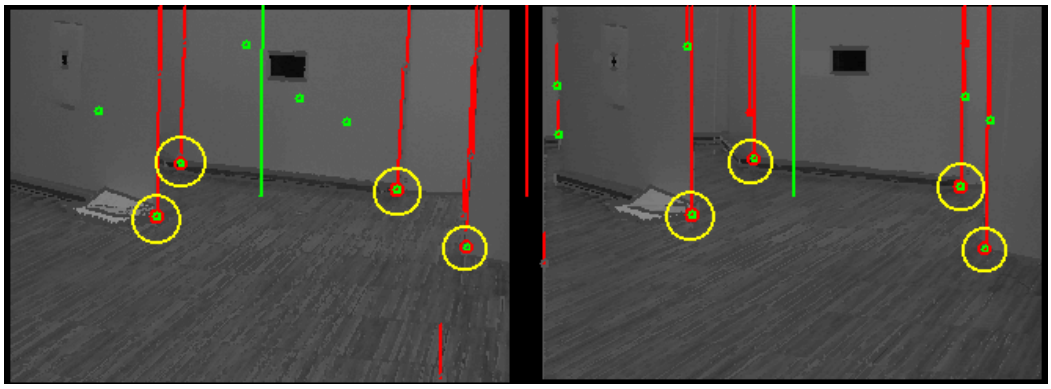
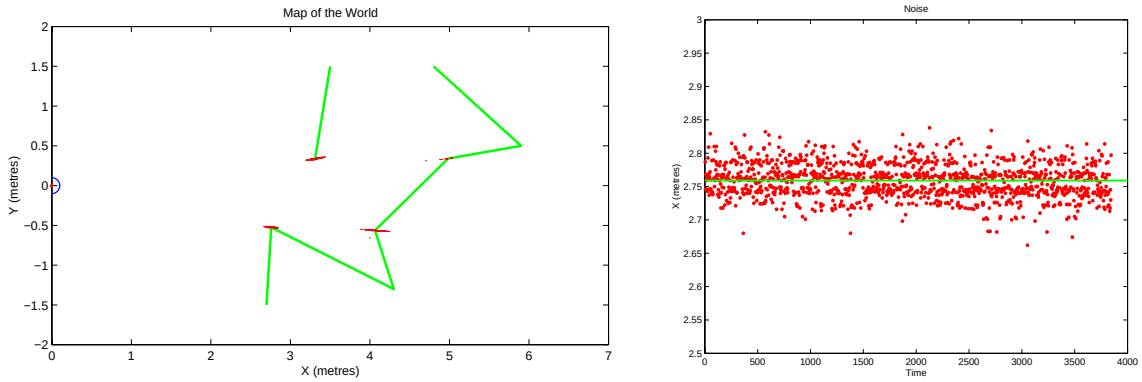


Fig. 5. Stereo images with detected edges and vertical lines.

### 9.3 Mapping of detected vertical world features

The detected verticals in the above image, obtained with the system at rest, were than mapped as shown in figure 6a. The straight lines represent the vertical panels providing the vertical features, and the robot is represented by the semicircle on the left.

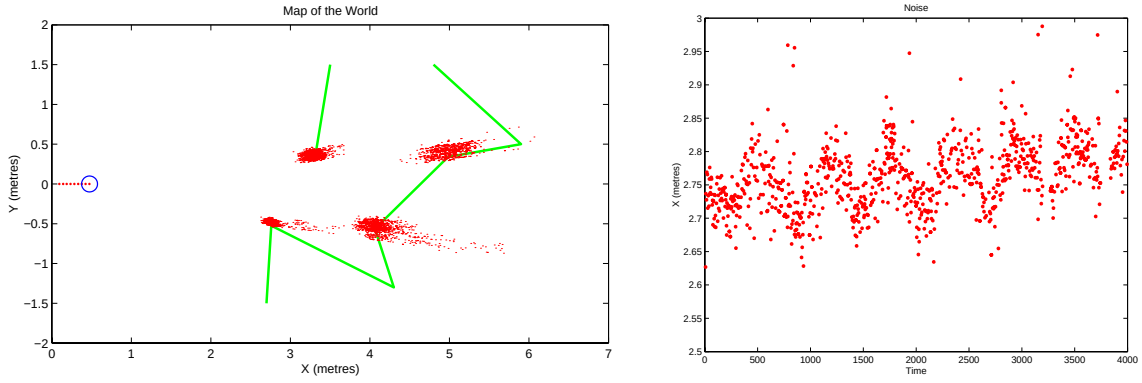



a) b)

Fig. 6. a) World map; b) Noise in $X$ coordinate for one of the points.

The distribution pattern clearly suggests that the $X$ coordinate is more unstable. Figure 6b shows the noise in the $X$ coordinate for one of the points. The noise has a clear quantisation pattern, due to the accelerometer discrete sampling. With suitable filtering of the detected raw points, and outlier removal, a more stable output can be obtained.

The problem is worse when the robot moves along a trajectory, because the vision system vibrates and oscillates even after the motors stop. The robot was set to move 0.5 $m$ forward and backwards several times, capturing base points of vertical features at the end positions, when the motors were stopped. The constructed map can be seen in figure 7a.




a) b)

Fig. 7. a) World map with robot movement; b) Noise in $X$ coordinate for one of the points.

The actual distance travelled by the robot was read from the robot's odometer, that has a 1/10 $in$ resolution, but can be adversely affected by physical conditions. Figure 7b shows the noise in the $X$ coordinate for one of the points.

The noise is due to the fact that the vision system vibrates and oscillates, even after the motors stop. But there is also a clear oscillation pattern with a long time constant, due to the two positions from which measurements we taken. The image pixel quantisation and odometer errors are the key factors producing this pattern.

Besides improving the signal filtering of the accelerometer data, to filter out the vibration induced errors, the raw maps can be improved with suitable filtering of the detected points and outlier removal.

## 10 Conclusions

This article presented our recent results on map-building and 3D reconstruction, using two different sensor modalities: inertial sensing and vision. The process described uses inertial and visual information to detect world vertical features. The horizon, as well as the reconstruction of the horizontal plane, was used to establish correspondence between stereo image vertical lines, enabling the detection of true 3D world vertical features. These features are used to build a *metric map*, which can be useful to improve mobile robot autonomy and navigation.

## References


1. Hans Moravec, Alberto Elfes, "High resolution maps from wide angle sonar", IEEE Int. Conf. On Robotics and Automation, pp.116-121, 1985.
2. S. Wei, Y. Yagi, M. Yachda, "Building local floor map by use of ultrasonic and omni-directional vision sensor", IEEE Int. Conf. On Robotics and Automation, pp 2548-2553, 1998.
3. Brian Yamauchi, Alan Schultz, William Adams, "Mobile robot exploration and map-building with continuous localization", IEEE Int. Conf. On Robotics and Automation, pp 3715-3720, 1998.
4. A. Silva, P. Menezes, J. Dias, H. Arajo, "Grid-Based framework for sensorial data integration in mobile robots", SPIE's International Symposium and Education Program on INTELLIGENT SYSTEMS & ADVANCED MANUFACTURING - Special Session on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems, Pittsburgh, Pennsylvania USA, October 14-15, 1997
5. J. Vandorpe, H. Van Brussel, H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder", IEEE Int. Conf. On Robotics and Automation, pp 901-908, 1996.
6. J. Gonzalez, A. Ollero, A. Reina, "Map building for a mobile robot equipped with a 2D laser rangefinder", IEEE Int. Conf. On Robotics and Automation, pp 1904-1909, 1994.
7. J. Lobo, Jorge Dias, Recovering 3D Structure from Images and Inertial Sensors, 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation, 5-7 December, 2000, ESTEC Noordwijk, The Netherlands
8. T. Viéville and O.D. Faugeras. Cooperation of the Inertial and Visual Systems. In Thomas C. Henderson, editor, *Traditional and NonTraditional Robotic Sensors*, volume F 63 of *NATO ASI*, pages 339–350. SpringerVerlag Berlin Heidelberg, 1990.
9. Thierry Viéville, François Romann, Bernard Hotz, Hervé Mathieu, Michel Buffa, Luc Robert, P.E.D.S. Facao, Olivier Faugeras, and J.T. Audren. Autonomous navigation of a mobile robot using inertial and visual cues. In M. Kikode, T. Sato, and K. Tatsuno, editors, *Intelligent Robots and Systems*, Yokohama, 1993.
10. Kenichi Kanatani. *Geometric Computation for Machine Vision.* Oxford University Press, 1993. ISBN 0-19-856385-X.
11. Jorge Lobo and Jorge Dias. Fusing of image and inertial sensing for camera calibration. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2001*, Baden-Baden, Germany, August 2001 (accepted for publication).
12. Jorge Dias, Carlos Paredes, Inácio Fonseca, and A. T. de Almeida. Simulating Pursuit with Machines. In *Proceedings of the 1995 IEEE Conference on Robotics and Automation*, pages 472–477, Japan, 1995.
13. Jorge Dias, Carlos Paredes, Inacio Fonseca, Helder Araujo, Jorge Baptista, and Anibal Traca de Almeida. Simulating Pursuit with Machine Experiments with Robots and Artificial Vision. *IEEE Transactions on Robotics and Automation*, 3(1):1–18, February 1998.
14. Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge Univercity Press, 2000. ISBN: 0521623049.
15. Bernd Jahne. *Digital Image Processing.* Springer-Verlag, 1997. ISBN 3-540-62724-3.
16. R.P.G. Collinson. *Introduction to Avionics.* Chapman & Hall, 1996. ISBN 0-412-48250-9.
17. Jorge Lobo and Jorge Dias. Integration of Inertial Information with Vision towards Robot Autonomy. In *Proceedings of the 1997 IEEE International Symposium on Industrial Electronics*, pages 825–830, Guimaraes, Portugal, July 1997.



This article was processed using
the TEX macro package with SIRS2001 style