Computer Vision-Based Navigation for Autonomous Blimps

LÚCIO DE SOUZA COELHO[1], MÁRIO FERNANDO MONTENEGRO CAMPOS[1], VIJAY KUMAR[2]

[1]Departamento de Ciência da Computação - ICEx - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627
31270-010 - Belo Horizonte - MG - Brazil
mario,omni@dcc.ufmg.br
[2]General Robotics and Active Sensory Perception Laboratory
3401 Walnut Street, Suite 300C, University of Pennsylvania
Philadelphia, PA 19104-6228, USA
kumar@central.cis.upenn.edu

Abstract. Autonomous dirigibles - aerial robots that are a blimp controlled by computer based on information gathered by sensors - are a new and promising research field in Robotics, offering several original work opportunities. One of them is the study of visual navigation, that would allow an autonomous dirigible to follow precise trajectories based on environmental visual information imaged by on-board cameras. The first step in the direction of the developing of such a capacity is the creation of a Computer Vision system able to supply to the autonomous dirigible its position and orientation in tridimensional space from the acquired images. This project describes such a system, capable of estimating position and orientation from images of visual beacons - objects with known geometrical properties and recognizable by the system. Experimental results showing the correct and efficient functioning of the system are shown and have your implications and future possibilities discussed.

Keywords: Aerial Robots, Computer Vision, Perspective n-Point Problem

1 Introduction

Lighter Than Air vehicles (LTAs) [Escher (1998)], also known as dirigibles, are aircraft composed basically by one or more balloons for buoyancy and propellers for propulsion. Weighting advantages and disadvantages of LTAs, it is clearly seen that application for them is remote sensing and aerial observation using small airships. Since LTAs can fly very close to the ground during long periods of time and with a minimal interference over the environment, dirigibles fill an important gap in the spectrum of aerial observations, supplying images with better resolution and much more acquisition flexibility than those acquired through satellite or airplanes. Monitoring and surveillance [Powers (1998)] of preservation or restrict access areas are examples of tasks that demand such an observational capacity.

The tasks above can be monotonous or even dangerous, and also considering that helium (used to fill the buoyancy balloons) is an expensive element and small aircraft is preferable, autonomous LTAs would be ideal to perform them. Autonomous LTAs are aerial robots - dirigibles guided by computer controls produced as a feedback to the information gathered by onboard sensors. The aerial nature of LTAs and the type of applications addressed here make visual sensors (like CCD cameras) a natural choice for autonomous dirigibles.

Research involving autonomous LTAs is still in its infancy. Although there have been some efforts directed to building autonomous LTAs [Craig et al. (1989)], a solid scientific framework for visually guided navigation for LTAs is still far from established. Therefore, that is an open avenue for important original work. Visual navigation would allow an autonomous dirigible to follow precise trajectories based on images of the surrounding environment acquired through onboard cameras. The first step for the development of such capacity is the creation of a Computer Vision [Ballard–Brown (1982)] system able to supply the autonomous dirigible its position and orientation in tridimensional space based on the acquired images.

This paper describes a Computer Vision system developed for autonomous blimps. Using the image of a visual beacon (a calibration object with known visual and geometrical properties) as a reference, the system can estimate the spatial position and orientation of the camera used for image acquisition, then allowing an autonomous blimp to perform navigation autonomously.

2 Problem Definition

The vision system described here has as input images of a *visual beacon* with known geometric properties: the visual beacon is used to define a set of space points with known distances between them. Formally, the beacon visually assigns a set $\{P_0, P_1, ..., P_n\}$ of *characteristic* *points* where the distances of the form $|\vec{P_i} - \vec{P_j}|, 0 <= i < j < n$ are known.

Depending on the number and disposition of the characteristic points, as well as the visual marks used to assign them, it is possible, using a single image of the beacon - acquired by a camera with known parameters (focus, resolution, CCD physical dimensions) - to estimate the position and orientation of that camera. In the case of an autonomous blimp, the position and orientation of the camera (or cameras) in relation to the rest of the structure is known, in a way that the homogeneous transform describing the translation and rotation of the blimp follows directly from the obtained transform describing the camera.

Therefore, the system described in this report performs two basic tasks:

1. Given the image of a visual beacon with know geometric properties, it determines the pixels that corresponds to the characteristic points of the beacon.

2. By knowing the optical and geometric parameters of the camera used to acquire the image mentioned in the previous item, it determines the position and orientation of the camera, and consequently of the blimp, in relation to the visual beacon.

In the following section, the mathematical and computational methods used to reach the above listed goals - with emphasis on item 2 - are described.

3 Materials and Methods

In this section, the physical resources employed in this work are firstly presented in a short description, followed by a somewhat detailed discussion about the visual beacon. After that, the mathematical and algorithmic methodology, used to solve the proposed problem and developed upon the geometric properties of the beacon, is presented.

3.1 Resources

The equipment that will incorporate the Computer Vision system developed in this project is a blimp (dirigible with helium filled flexible hull) commercially obtained, remote controlled and equipped with a CCD pinhole camera installed in its gondola. The CCD camera and the propellers have an wireless connection with an workstation that will execute the software. The blimp is for use in indoor environments only.

A CCD camera light enough for onboard use in the blimp was used for image acquisition. It has a 640x480 resolution with 24 bits, RGB, per pixel.

The Computer Vision software was written in C++. Preliminary software versions were written in the Java

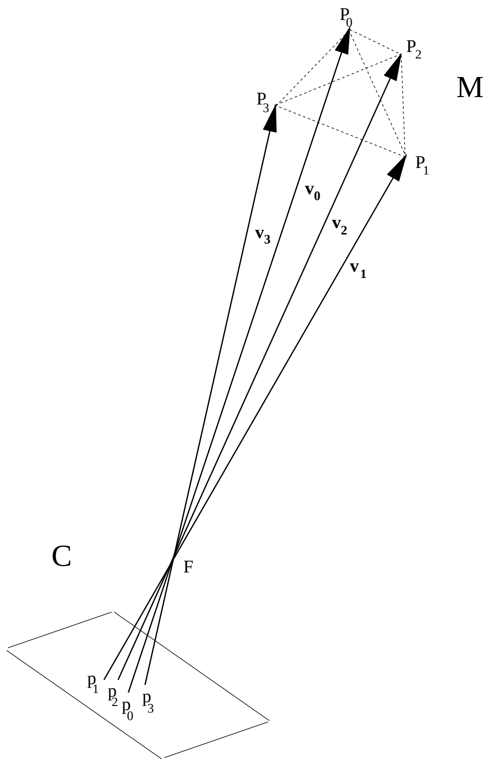

Figure 1: image projection of the vertices of a tetrahedral beacon M over the image plane of camera C.

programming language. The final version was tested running on a machine with a 266 MHz Pentium MMX processor and 64 Mbytes of RAM memory, under the Windows NT operating system

3.2 The Visual Beacon

As an introduction to the description of the visual beacons built and used in the experiments, it is first necessary to discuss about the mathematical and geometrical limitations that influenced their conceptions.

Let C be a camera with focal point F. Let M be a visual beacon with a set of 4 non-coplanar characteristic points $\{P_0, P_1, P_2, P_3\}$. Let $\{p_0, p_1, p_2, p_3\}$ be the coplanar points corresponding to the image projections of the characteristic points of M over the image plane of C. Let $\vec{V_i} = P_i - p_i, 0 <= i < 4$ be the light-ray-path vectors [1] going from the points p_i to the corresponding P_i passing through F, and $\vec{v_i} = F - p_i, 0 <= i < 4$ the vectors in the same direction of V_i, but going just until F. Figure 1 tries to illustrate that geometrical construct.

Once the vectors $\vec{V_i}$ are found, the position and orientation of C can be determined. Since the distances be-

[1]Rigorously speaking, vectors and points are different geometrical entities, but along this article they will be informally treated in a similar manner - except in cases where a distinction makes itself necessary.

tween the points P_i are known and vectors $\vec{v_i}$ are determinable if the points p_i are known, the following equation system can be specified:

$$\begin{cases} |\alpha_0\vec{v_0} - \alpha_1\vec{v_1}| = D_{0,1} \\ |\alpha_0\vec{v_0} - \alpha_2\vec{v_2}| = D_{0,2} \\ |\alpha_0\vec{v_0} - \alpha_3\vec{v_3}| = D_{0,3} \\ |\alpha_1\vec{v_1} - \alpha_2\vec{v_2}| = D_{1,2} \\ |\alpha_1\vec{v_1} - \alpha_3\vec{v_3}| = D_{1,3} \\ |\alpha_2\vec{v_2} - \alpha_3\vec{v_3}| = D_{2,3} \end{cases}$$

, where $D_{i,j} = |P_i - P_j|, 0 <= i < j <= 3$ is the distance between points P_i and P_j. The unknowns of the system are $\alpha_0, \alpha_1, \alpha_2, \alpha_3$, and $\vec{V_i} = \alpha_i\vec{v_i}$. Expanding the modulus operations on the left side of the equations, a non-linear system with six quadratic equations and four unknowns is thus obtained. The existence of six equations guarantees one solution.

Therefore, a visual beacon with tetrahedral topology - that is, having four non-coplanar characteristic points - guarantees a unique solution to the values $\vec{V_i}$ and consequently a unique position and orientation to the camera for the point set p_i determined in an image.

However, tetrahedral - and therefore tridimensional - beacons are more difficult to construct and reproduce than the bidimensional ones; in particular, practical applications of autonomous dirigibles, where the distances involved could be large and thus the visual beacon, seem to favor the use of *bidimensional* ones.

A bidimensional beacon would have to have a minimum of three characteristic points to make possible the determination of position and orientation of the camera - since with a number of points less than three the number of solutions found for position and orientation would be infinite. Nonetheless, a *triangular* beacon would imply in an equation system with just three quadratic equations, in a way that the number of solutions for a given projection of characteristic points on the image plane would be 2 or 4, as shown in [DeMenthon–Davis (1992)]. That is, for a given image of a triangular beacon, there would be two or four possible positions/orientations of the beacon with the same characteristic point projections found in the image.

However, this ambiguity can be removed if distortions in the vertex markers, caused by *perspective projection*, are taken into account. Observing the apparent size of each marker, it is possible to determine the ratios between their distances and thus to chose one among the several solutions. Hence, a triangular visual beacon, an isosceles right-triangle with $0,155m$ sides, was chosen. Each vertex is assigned by a black disk with a $0,06m$ radius contained in the triangle plane, with its center coinciding with the corresponding vertex. The disks (markers) are labeled from 0 to 2 with light-colored holes on their surfaces, and the number of holes corresponds to the number of the assigned vertex. Finally, the strategy of using dark markers against a light background is used in order to ease the segmentation step in the image acquisition phase, as described in the next section. Figure 2 shows an image of the beacon.

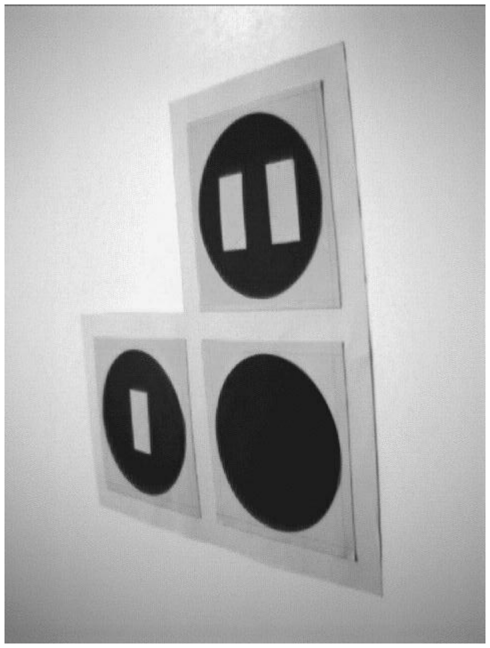
Figure 2: image of the triangular beacon against a light background.

Once the geometrical and visual details of the beacon have been described, the following sections discuss the image processing techniques used for determining the points $p_0, \ldots, p_{n-1}$ in a given image, as well as the methodologies used to determine the position and orientation of the camera based on that.

3.3 Image Processing Methodology

The stage of image processing involves standard Digital Image Processing [Castleman (1996)] and Computer Vision techniques. It is divided in the following steps:

1. **Segmentation:** firstly, using the contrast between the vertex markers and the image background, a simple threshold binarization is applied over the image, in a way that the pixels corresponding to the markers turn white (or true) and the pixels corresponding to the background turn black (or false) in the binary image [Barrera–Banon (1994)]. A fixed threshold of 50 (in a 256-level grayscale) worked for all the test images.

2. **Labeling:** the objects [Coelho–Campos (1995)] in the binary image are identified – a contour follow-

ing algorithm is used to locate the objects and determine their rectangular boundaries in the image. It is assumed that the rectangular boundaries of each object do not overlap each other.

3. **Identification of Markers:** only the three objects with largest pixel areas obtained in the previous step are considered in this one, in order to eliminate artifacts left by small binarization imperfections. The ratios between the pixel area of rectangular boundaries of the objects and the pixel area of their corresponding objects are determined and used to identify the markers, in such a way that the smallest ratio (no holes) corresponds to the P_0 marker, the largest ratio (two holes) corresponds to P_2 and the remaining one to P_1. Finally, the geometrical centers of each rectangular boundary are calculated and assumed to be the points $p_0, ..., p_{n-1}$ correspondingly assigned.

3.4 Geometrical Methodology

The application of the geometrical methodologies starts with computing of the points $p_0, ..., p_{n-1}$. An analytical solution is then calculated using those points as inputs and solving the non-linear system that describes the problem. This solution is then used as an initial condition for a local search method that supplies a final solution with better precision.

3.4.1 The Analytical Solution

In the case of a triangular visual beacon, there are different solutions of the equation system for a triangle. Therefore some information other than the simple projection of the characteristic points on the image plane has to be used in order to solve that ambiguity. That can be easily done taking into account the perspective projection inherent to the image formation process. The perspective makes the vertex markers closer to the camera to have a larger apparent size than those of the farther markers. Since the three markers used in the triangular beacon have the same shape, the same size and are in the same plane, it is easily inferred from the inverse square law that

$$\frac{|\vec{V_i}|}{|\vec{V_j}|} = \frac{\epsilon_i}{\epsilon_j} = \sqrt{\frac{A_j}{A_i}}, 0 \le i < j \le 3$$

, where $A_k, 0 \le k \le 3$ is the area in pixels occupied in the image by the marker of P_k. In particular,

$$c_1 = \sqrt{\frac{A_0}{A_1}} \Rightarrow \epsilon_1 = c_1\epsilon_0$$
$$c_2 = \sqrt{\frac{A_0}{A_2}} \Rightarrow \epsilon_2 = c_2\epsilon_0$$

Thus, substituting the above expressions, as well as $\vec{e_0} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}$, $\vec{e_1} = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}$ and $\vec{e_2} = \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}$ in the equation system

$$\begin{cases} |\epsilon_0\vec{e_0} - \epsilon_1\vec{e_1}| = D_{0,1} \\ |\epsilon_0\vec{e_0} - \epsilon_2\vec{e_2}| = D_{0,2} \\ |\epsilon_1\vec{e_1} - \epsilon_2\vec{e_2}| = D_{1,2} \end{cases}$$

analogous to the one presented in the Visual Beacon section, it is possible to find through some algebraic manipulation that

$$\epsilon_{0(1)} = \frac{D_{0,1}}{\sqrt{(x_0-c_1x_1)^2+(y_0-c_1y_1)^2+(z_0-c_1z_1)^2}}$$
$$\epsilon_{0(2)} = \frac{D_{0,2}}{\sqrt{(x_0-c_2x_2)^2+(y_0-c_2y_2)^2+(z_0-c_2z_2)^2}}$$
$$\epsilon_{0(3)} = \frac{D_{1,2}}{\sqrt{(c_1x_1-c_2x_2)^2+(c_1y_1-c_2y_2)^2+(c_1z_1-c_2z_2)^2}}$$

The system calculates those three possible values for ϵ_0. The value chosen as initial condition for the optimization algorithm is the ϵ_0 which implies in a minimum matching error. That is, the ϵ_0 (with the corresponding ϵ_1 and ϵ_2) that assigns in the space an E triangle closest to the M one marked by the beacon. The measurement of that matching is done accordingly to the minimum Euclidean distance criterion defined in the next section.

3.4.2 The Optimizing Algorithm

The optimization method used here works with the unit vectors $\vec{e_i} = \frac{\vec{v_i}}{|\vec{v_i}|}$, that is, $\vec{e_i}$ has the same direction of $\vec{v_i}$. Points defined by vectors $\epsilon_i\vec{e_i}$ form a figure E in the space and the search algorithm works finding scalar factors ϵ_i such as that E is the closest (within a local optimization and under a certain precision) to the visual beacon M with n characteristic points. The similarity between E and M is measured using the euclidean distance between the edge vectors of E and M, defined as

$$S(\epsilon_0, ..., \epsilon_{n-1}) =$$

$$= \sqrt{\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1}(|\epsilon_i\vec{e_i} - \epsilon_j\vec{e_j}| - |P_i - P_j|)^2}$$

Thus, the scalar factors ϵ_i such as $\vec{V_i} = \epsilon_i\vec{v_i}$ can be found with the minimization

$$\min S(\epsilon_0, ..., \epsilon_{n-1})$$

Since $\vec{e_i}$ are unit vectors, each ϵ_i is in fact the *distance* from point P_i to the camera [2], along the direction of $\vec{v_i}$.

The local search algorithm used to perform the above described minimization is presented below:

[2]That assertion is not precise in the sense that ϵ_i is the distance between p_i and P_i, not the distance of some central point of the camera to P_i. Since in general ϵ_i distances are much larger than distances between the p_i points, the assertion can be considered true for practical purposes and it is in fact assumed as truth during the geometrical developments performed here.

Input: $M, \vec{e_i}, 0 \leq i \leq n-1$, initial ϵ_i distances and the incremental step δ.

Output: the ϵ_i distances that best locally approximate M.

1. Let be $\Pi = \{S(\epsilon_0 + m_0\delta, \ldots, \epsilon_{n-1} + m_{n-1}\delta) | m_i \in \{-1, 0, 1\} \forall 0 \leq i \leq n-1\}$.
2. For each element $\pi \in \Pi$, if $\pi < S(\epsilon_0, \ldots, \epsilon_{n-1})$ then assign $\epsilon_i \leftarrow \epsilon_i + m_i\delta$ using the factors m_i corresponding to π.
3. If no $\pi < S(\epsilon_0, \ldots, \epsilon_{n-1})$ was found in the previous step, then stop; else, return to Step 1.

It is easily seen then, in the algorithm above, that the initial solution starts as the point $[\epsilon_0, \ldots, \epsilon_{n-i}]$ of a n-dimensional state space (with $n = 3$ for the triangular case), and the neighboring points $[\epsilon_0 + m_0\delta, ..., \epsilon_{n-1} + m_{n-1}\delta]$ are tested until one where the value of S is smaller is found; the provisory solution is then assigned to that minimizing neighbor and the optimizing step is repeated, until a point where the value of S is less than those of all its neighbors is reached. That's a local minimum, which is then considered the final solution.

4 Experiments

This section presents the description of experiments performed in order to verify the accuracy of the system in terms of position and orientation. The results obtained are presented inside the geometric framework of coordinate systems defined in the subsection below.

4.1 Coordinate Systems

The algorithm performs all the calculations with respect to the *camera coordinate system* $\{\mathrm{C}\}$. That system is an orthonormal basis [Boulos–Camargo (1987)] that has as origin the CCD matrix center, X axis parallel to the CCD width, Y axis parallel to the CCD height and Z axis coincident with the camera axis (line perpendicular to image plane passing through the focal point), pointing toward the back of the camera. This arrangement, as shown in Figure 3, is adequate for visualization and verification purposes in an image acquired by the camera. For example, in pictures shown in Figure 5, the plane XY corresponds to the plane of the page sheet showing the images, and the axis Z is perpendicular to that plane, pointing to the observer.

On the other side, $\{\mathrm{M}\}$, the beacon coordinate system - which was defined as the system used to effectively express the position and orientation of the autonomous dirigible - has as origin the point P_0, the vertex of the right angle in the triangle. The axis X is the line passing through P_0 and P_2, increasing from P_0 to P_2; the axis Y is the line passing through P_0 and P_1, increasing from P_0 to P_1; and, finally, the axis Z is perpendicular to the XY plane, passing through P_0 and pointing upwards. Figure 4 schematizes that *world coordinate system* [Craig (1989)].

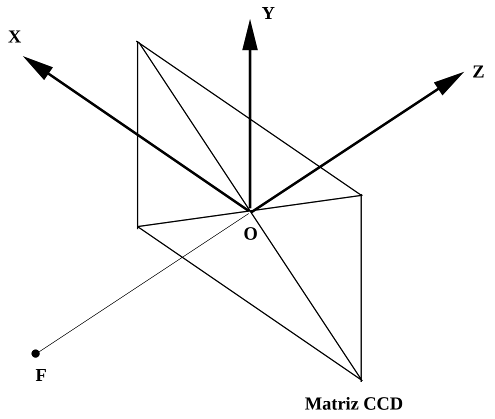

Figure 3: scheme of the camera coordinate system.

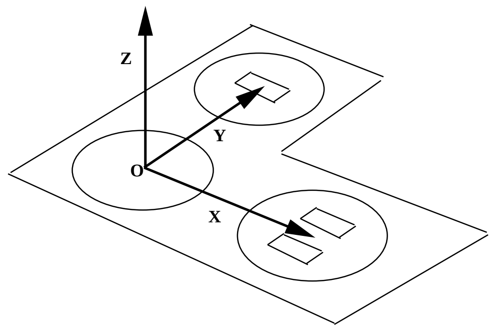

Figure 4: scheme of the beacon coordinate system.

Once the vectors $V_i, 0 \leq i \leq 2$ are obtained in $\{\mathrm{C}\}$, the position and orientation of the camera - described by a homogeneous transform - in $\{\mathrm{M}\}$ are obtained as described bellow.

The frame $\{\mathrm{M}\}$ is assembled based upon the vectors $\vec{V_i}$ found. Unit vectors $\vec{X}$,$\vec{Y}$ e $\vec{Z}$ are computed as follows:

$$\vec{X} = \frac{\vec{V_2} - \vec{V_0}}{|\vec{V_2} - \vec{V_0}|}$$

$$\vec{Y} = \vec{X} \times \vec{Z}$$

$$\vec{Z} = \frac{\vec{X} \times (\vec{V_1} - \vec{V_0})}{|\vec{X} \times (\vec{V_1} - \vec{V_0})|}$$

It can be seen that

$$\vec{Y} \neq \frac{\vec{V_1} - \vec{V_0}}{|\vec{V_1} - \vec{V_0}|}$$

In fact, $\vec{V_1} - \vec{V_0}$ is used to calculate $\vec{Z}$, and then $\vec{Y}$ is obtained based on the cross product of $\vec{X}$ and $\vec{Z}$. This somehow indirect way is used to guarantee that the calculated coordinate system basis is in fact orthonormal.

From that point on, the position of the camera follows directly as the projections of $-\vec{V_0}$ along $\vec{X}$, $\vec{Y}$ and $\vec{Z}$, that is, the position is described by the vector

$$[\ (-V_0) \bullet \vec{X} \quad (-V_0) \bullet \vec{Y} \quad (-V_0) \bullet \vec{Z}\ ]$$

To calculate the camera orientation, it is necessary to determine the projections of the vectors $\vec{I} = [\ 1 \quad 0 \quad 0\ ]$, $\vec{J} = [\ 0 \quad 1 \quad 0\ ]$ and $\vec{K} = [\ 0 \quad 0 \quad 1\ ]$ (the very orthonormal basis of $\{\mathrm{C}\}$, characterizing the axis X, Y and Z, respectively) over $\vec{X}$, $\vec{Y}$ and $\vec{Z}$, respectively. Thus, the orientation is expressed in the form of the rotation matrix

$$R = \begin{bmatrix} \vec{I} \bullet \vec{X} & \vec{J} \bullet \vec{X} & \vec{K} \bullet \vec{X} \\ \vec{I} \bullet \vec{Y} & \vec{J} \bullet \vec{Y} & \vec{K} \bullet \vec{Y} \\ \vec{I} \bullet \vec{Z} & \vec{J} \bullet \vec{Z} & \vec{K} \bullet \vec{Z} \end{bmatrix}$$

Hence, the homogeneous transform describing position and orientation of the camera under the M frame is

$$\begin{bmatrix} \vec{I} \bullet \vec{X} & \vec{J} \bullet \vec{X} & \vec{K} \bullet \vec{X} & (-V_0) \bullet \vec{X} \\ \vec{I} \bullet \vec{Y} & \vec{J} \bullet \vec{Y} & \vec{K} \bullet \vec{Y} & (-V_0) \bullet \vec{Y} \\ \vec{I} \bullet \vec{Z} & \vec{J} \bullet \vec{Z} & \vec{K} \bullet \vec{Z} & (-V_0) \bullet \vec{Z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With the coordinate systems and their relationships defined we are now able to describe experiments performed to evaluate the methodology, as well as the results that were obtained.

4.2 Experimental Data

Three images of the triangular beacon - shown in Figure 5 - were taken and the distances from the camera to the points P_0, P_2 and P_1 were measured. The distances (in meters) calculated by the algorithm (using $\delta = 0,001m$) are compared with the ones manually measured (with a tape measure) in Table 1.

The precision of the distance calculation seems to have an error margin of some percent points. In fact, maybe the precision is higher, since the manual measurement process itself tends to be somewhat imprecise. A Java applet demonstrating the distance calculations by the algorithm for the images above can be consulted at **http://www.cis.upenn.edu/ coelho/java/experiment4/**.

With respect to the transformation, the calculated values are shown in Table 2. Verifying if the signals of the coordinates correspond to the octant of the $\{\mathrm{M}\}$ frame where the camera really was in the moment of image acquisition, one can qualitatively evaluate the positions shown in the transforms above. Observing the images, it can be noticed that the camera was in the fourth

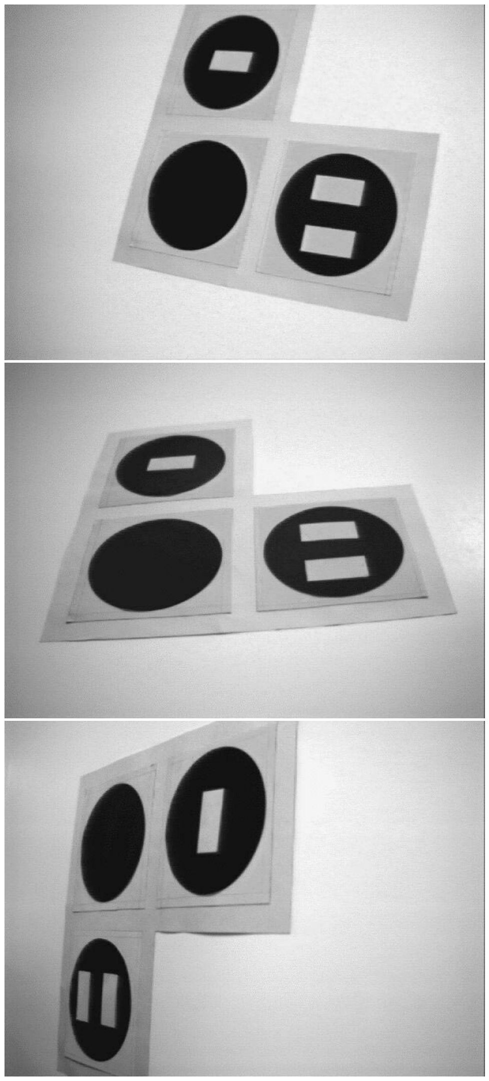
Figure 5: images 1 (top), 2 (middle) and 3 collected in the experiments with the triangular beacon.

Image	ϵ	System	Manual	Error
1	0	$0,61$	$0,61$	0%
	1	$0,65$	$0,66$	2%
	2	$0,54$	$0,56$	4%
2	0	$0,51$	$0,51$	0%
	1	$0,63$	$0,61$	3%
	2	$0,50$	$0,48$	4%
3	0	$0,59$	$0,58$	2%
	1	$0,46$	$0,46$	0%
	2	$0,58$	$0,58$	0%

Table 1: distances between camera and characteristic points obtained by the system and by manual measurement.

Image:	**Transform:**
1	$\begin{bmatrix} 0,86 & -0,10 & 0,50 & 0,34 \\ 0,23 & 0,95 & -0,20 & -0,09 \\ -0,45 & 0,28 & 0,84 & 0,50 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2	$\begin{bmatrix} 1,00 & 0,08 & 0,06 & 0,11 \\ -0,01 & 0,67 & -0,75 & -0,36 \\ -0,10 & 0,74 & 0,66 & 0,34 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3	$\begin{bmatrix} -0,05 & -0,99 & 0,11 & 0,11 \\ 0,69 & 0,05 & 0,73 & 0,50 \\ -0,73 & 0,12 & 0,68 & 0,29 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 2: transforms describing position and orientation of camera for each test image.

octant (negative Y coordinate and positive X and Z coordinates) in images 1 and 2 and in the first octant (all coordinates positive) in image 3. As a matter of fact, those were the signals found in the positions - the vectors $[\ 0,34 \quad -0,09 \quad 0,50\ ]$, $[\ 0,11 \quad -0,36 \quad 0,34\ ]$ and $[\ 0,11 \quad 0,50 \quad 0,29\ ]$, corresponding to images 1, 2 and 3, respectively - present in the transforms.

The orientation is somewhat more difficult to be qualitatively evaluated, although in more evident cases like the one in image 2, where the X axis of the camera is almost parallel to the X axis of the beacon, the first column of the transform is $[\ 1,00 \quad -0,01 \quad 0,10\ ]^T$, very close to the expected value $[\ 1 \quad 0 \quad 0\ ]^T$ for perfectly parallel axis.

Anyway, a *quantitative* evaluation of orientation was conducted for at least one image. An additional beacon image was taken with the origin of M distant $0,64m$ of the camera, contained in the Z axis of C; the X axis of the beacon was put in parallel with that of the camera, and the axis Z and Y of the beacon were rotated $57°$ in relation to the corresponding ones in the camera. Figure 6 depicts a scheme of the assembling.

With those pre-determined position and orientation, the expected transform matrix to the image would be

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{sen}57° & -\text{cos}57° & -0,35 \\ 0 & \text{cos}57° & \text{sen}57° & 0,54 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where $\sin 57° \approx 0,84$ and $\cos 57° \approx 0,54$. The transformation in fact obtained by the system was

$$\begin{bmatrix} 1,00 & -0,02 & -0,08 & -0,05 \\ -0,02 & 0,87 & -0,50 & -0,31 \\ 0,08 & 0,50 & 0,86 & 0,55 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

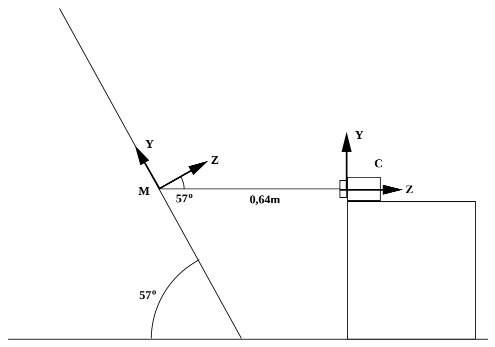

Figure 6: assembling destined to verify the correction of orientations calculated by the system.

Then, once more the values were very close to those expected also in the case of orientation. The rotation angle of the camera about the X axis corresponding to the above transformation is of approximately $60°$, against the expected $57°$.

Since the end application of the vision system here described is autonomous navigation for a blimp, issues of real time are relevant and an exposition of execution times reached by the system is worthwhile. The table below summarizes the times (in seconds) spent by each of the processing phases:

Image	**Image Processing**	**Geometric Processing**	**Total**
1	0.101	0.080	0.181
2	0.090	0.080	0.170
3	0.140	0.060	0.200
Average:	0.110	0.073	0.184

Table 3: execution times during tests.

It can be seen that the overall performance of the system allows a frame rate of approximately 5 Hz, which seems to be compatible with a real time application.

5 Discussion and Future Work

The system presented good results both in terms of accuracy and time, as shown by the experimental results. It represents a conceptual evidence favoring the viability of visual beacon-based systems for the navigation of autonomous dirigibles. Besides, the applicability of the vision system is extensible to other kinds of aerial robots (like robot planes and helicopters) and perhaps to roving robots and even conventional manipulators.

The next step to test the system under several flight conditions will be to connect the program to the onboard camera of the blimp and study the behavior of the Computer Vision system in end-application conditions. This stage is currently under development. After that, the control cycle could be closed, with the blimp following predefined trajectories based on the information calculated by the vision system.

Giving more robustness to the beacon recognition process and using larger dirigibles, capable of resist to outdoor winds, several practical applications will be possible. That robustness enhancement will consist primarily in the use of *visual tracking* [Wang et al. (1998)] techniques to keep the *focus of attention* of the system on the visual markers in unstructured environments. Techniques of *template matching* [Coelho–Campos (1997)] and *inertial navigation* [Kelly (1994)] can also be involved to deal with loss of focus of attention.

A long term possibility for further research is the use of objects already existent in the environment as visual beacons. For example, in order to fly over a city, an LTA robot could use buildings as visual beacons for its navigation. Eventually, this project could also evolve to visual navigation systems dedicated to high altitude dirigibles, which would use no more objects of euclidean geometry as beacons, but objects with fractal geometry [Mandelbrot (1988)], like mountains and rivers.

Acknowledgements

This research was supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under the process 170645/97-5 (NV) BSP and the Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

References

[Escher (1998)] Escher, R., *Airship and Blimp Resources*, http://www.hotairship.com/index.html

[Powers (1998)] Powers, J., *The Intelligent Surveillance Blimp*, http://watt.seas.Virginia.EDU/ jap6y/isb/home.html

[Craig et al. (1989)] Craig, B. et al., *Computer Blimp: A Technical History*, http://www.robotgroup.org/projects/blimphst.html

[Ballard–Brown (1982)] Ballard, D. H. and Brown, C. M., *Computer Vision*, Prentice Hall, 1982.

[Craig (1989)] Craig, J. J., *Introduction to robotics: mechanics and control*, 2nd edition, Addison-Wesley Publishing Company, Inc., 1989.

[Stroustrup (1987)] Stroustrup, B., *The C++ Programming Language*, 1st edition, Addison-Wesley Publishing Company, Inc, 1987.

[Lemay–Perkins (1996)] Lemay, L. and Perkins, C. L., *Teach Yourself Java in 21 Days*, Sams Publisher, 1996.

[DeMenthon–Davis (1992)] DeMenthon, D. e Davis, L. S., “Exact and Approximate Solutions of the Perspective-Three-Point Problem”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 11, November 1992, pp. 1100-5.

[Castleman (1996)] Castleman, K. R., *Digital Image Processing*, Prentice-Hall, 1996.

[Barrera–Banon (1994)] Barrera, J. and Banon, G.J.F., *Bases da Morfologia Matemática Para a Análise de Imagens Binárias*, UFPE-DI, Recife, 1994.

[Coelho–Campos (1995)] Coelho, L. S. and Campos, M. F. M., “Metodologia Computacional para o Modelamento e Análise de imagens de Redes de Trincas em Superfícies Metálicas”, in *Anais do XXII Simpósio de Hardware e Software - XV Congresso da Sociedade Brasileira de Computação e PANEL 95 - XXI Conferência Latino-Americana em Informática*, Gramado, Brasil, 1995.

[Boulos–Camargo (1987)] Boulos, P. e Camargo, I., *Geometria Analítica: um Tratamento Vetorial*, Segunda Edição, McGraw-Hill, São Paulo, 1987.

[Wang et al. (1998)] Wang, H., Chua, C. S. and Sim, C. T., “Real-Time object tracking from corners”, in *Robotica*, Vol. 16, 1998, pp. 109-116.

[Kelly (1994)] Kelly, A., “Modern Inertial and Satellite Navigation”, Technical Report CMU-RI-TR-94-15, Carnegie Mellon University, May 1994.

[Coelho–Campos (1997)] Coelho, L. S. and Campos, M. F. M., “Similarity-based versus template matching-based methodologies for image alignment of polyhedral-like objects under noisy conditions”, in *Anais do SIBGRAPI’97 - X Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, 1997.

[Mandelbrot (1988)] Mandelbrot, B., *Fractal Geometry of Nature*, W. H. Freeman & Co., 1988.