INVESTIGATION OF THE ACCURACY OF A MACHINE VISION ROBOTIC ARM SYSTEM FOR RENDEZVOUS AND DOCKING OPERATIONS

David Valle[1], Luis Mollinedo[1], Alberto Medina[1]

[1]GMV
Isaac Newton 11, P.T.M. Tres Cantos, E-28760 Madrid (Spain)
dvalle@gmv.com, lmollinedo@gmv.com, amedina@gmv.com

ABSTRACT

One of the most important elements in space missions are the sensors required to acquire navigation data. RVD (RendezVous and Docking) operations or astronauts tracking by a support vehicle are some examples of missions that need reliable sensors. On the other hand, computer vision is a growing application in the sensors industry for space missions that could fit with the mentioned space applications.

This paper presents an investigation of the performances of a machine vision system suitable to be used as main sensor for GNC (Guidance, Navigation and Control) algorithms for RVD or other operations. These performances are later evaluated in a simplified although demanding machine vision real environment composed by a camera, a robotic arm and a docking target base.

The main purpose of this paper is to investigate on that computer vision techniques that fulfil robotics and spacecraft navigation functional requirements while decreasing the cost of the required sensors. One of the drawbacks at the current time is that several techniques of image processing are very high-CPU consuming tasks and many boost performances must be added to reduce processor load and to improve the accuracy of the achieved navigation.

The implemented test bench, with the design of the software application (particularly the integration between the robotic arm and the camera with the different software elements) and the algorithms for the navigation, just as the tests will be presented next.

1. INTRODUCTION

Regarding relative navigation field within RVD scenarios, the following sensors can be considered as a reference baseline (in absence of GNSS coverage) from the point of view of performance. In the table showed below, green shadowed blocks indicate active participation on navigation loop at each range interval, while orange shadowed blocks indicate secondary uses.

Table 1. Relative navigation sensors in planetary RVD operations

<table>
<tr><th colspan="2" rowspan="2">Sensor</th><th colspan="5">Range region</th></tr>
<tr><th>1 m – 1 km</th><th>1 km – 5 km</th><th>5 km – 30 km</th><th>30 km – 410 km</th><th>410 km – …</th></tr>
<tr><td colspan="2">NAC</td><td colspan="2">Pictures</td><td>Medium/Fine LOS</td><td>Fine LOS</td><td>Contingency search</td></tr>
<tr><td rowspan="2">RFS2</td><td>L Mode</td><td>X</td><td>X</td><td>X</td><td>Fine ranging & coarse LOS</td><td>X</td></tr>
<tr><td>S Mode</td><td>CAM sensor</td><td colspan="2">Fine ranging & medium LOS</td><td>X</td><td>X</td></tr>
<tr><td colspan="2">LIDAR</td><td colspan="2">Fine ranging & Fine LOS</td><td>X</td><td>X</td><td>X</td></tr>
</table>

Acronyms: NAC (Navigation Aided Camera), RFS2 (Dual Mode Radio Frequency System), LIDAR (Light Detection And Ranging), LOS (Line Of Sight).

Looking at commercially-available LIDAR systems, they show resolution in the order of magnitude of centimetres. A more detailed view shows that a standard LIDAR system with a weight of 40 kilograms and a resolution of 2 centimetres can work in an operating range of 1 kilometre. If a higher operating range is needed, it is necessary to increase the power of the system, for which the total weight of the LIDAR system is increased too.

Regarding systems based on optical sensors, the conclusion obtained is that they have a resolution of several centimetres while keeping the weight under the tens of kilograms.

Through a comparison between LIDAR sensors and optical sensors, the next conclusions have been obtained:

- Systems with LIDAR sensors present a higher accuracy with regard to optical sensors, since the first ones are more accurate than the second ones in the computation of the distance between two objects. On the other hand, optical sensors also allow to calculate this distance as well as to be more versatile (forms, size, dimensions, etc. can be obtained by means of image processing).
- LIDAR sensors are usually weighter than optical sensors, reason why they limit the useful charge of some spacecrafts (ATV, Soyuz …) which are launch to space. In this sense, an optical sensor is preferable, since it offers minors dimensions and minor weight.
- On the contrary, a system LIDAR is generally more robust than a system based on an optical sensor, but also more expensive, which could limit the additional instrumentation of the spacecraft.

Finally and based on this comparison, the optical sensor has been chosen to develop the machine vision system.

2. COMPUTER VISION: APPLIED IMAGE PROCESSING TECHNIQUES

The objective of the computer vision in this system is to provide 2D navigation for the RVD operation. All the computer vision techniques used are based on the image processing. By processing images obtained from optical sensors the vision system will be able to detect and to track the docking area within the target satellite.

Most common technique when doing target detection is pattern matching; i.e. we already know how our target looks like and we look for this pattern inside the processed image, Therefore, we need first to design a representative pattern of the object class to search and later we must apply the pattern matching. This technique, of great potential and simplicity in detection, recognition and tracking of objects, presents two disadvantages in the system:

- High computational cost, of the order of

  $$O(WHwh) \tag{1}$$

  where WH is the image size and wh is the pattern or template size. For example, when the image resolution increases the double, the time multiplies by 16.
- High sensitivity to pattern dimensions like size, rotations, scale, etc.

On the one hand, to solve the high computational cost (keeping the image characteristics in terms of resolution, colour and depth) the following techniques are used (only for the image processing):

- RGB to gray scale conversion: the image processing is performed in gray scale. For it, it is necessary to convert the RGB image (3 channels) into a gray scale image (1 channel).
- Image segmentation: to get a region of interest (ROI), where the object will be searched later and where:

  $$resolution(ROI) \leq resolution(image) \tag{2}$$

  In order to apply this technique, first the object in the complete image is searched, and it is supposed that the object to search will be in a region next to the present one of the found object later. Thus and starting off from the central point of the object detected at moment t, image is segmented at $t+1$ in width and height according to a required size which will be always inferior to the one of the captured image obviously.

  As disadvantage we face that if target moves very fast, the object could go out of previous ROI; thus, it is very important to adjust properly the size of the ROI.
- Image resolution reduction by means of downsampling step of Gaussian pyramid decomposition. First it convolves source image with the specified filter and then downsamples the image by rejecting even rows and columns.

  Disadvantage: quality is lost when the image resolution is reduced by means of the convolution and this affects to the image processing harming the detection.

On the other hand, to solve the problem of sensitivity to rotations and scale target motion, it is possible to decide on three possible techniques:

- To use several patterns, with different sizes and rotations.
- To perform a multiscale search. The process can be applied scaling the pattern to 110%, 120%, …
- To use some selective attention technique or heuristic. For example, to use colour or edges to focus attention on certain parts of the image, although this is not always possible.

In our application case, it has been chosen to apply a scaling factor to the pattern. This scaling factor can be configurable by the user. In addition, also a heuristic one has been used to focus attention on certain parts of the image.

Canny edge detector is used to reject some image regions that contain too few or too much edges and thus can not contain the searched object. This speeds up the search considerably.

The next figure shows some of the image processing techniques used for the development of our system:

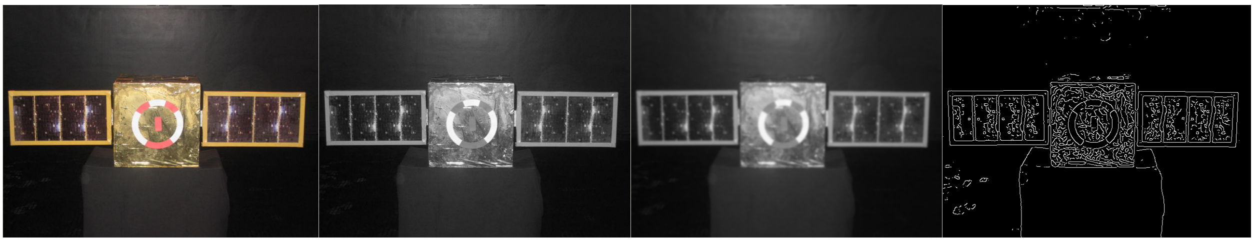

Fig. 1. From left to right, applied image processing techniques: original RGB image, RGB to gray scale conversion, noise reduction with a 9x9 Gaussian mask and Canny edge detector (thresholds 0-255, aperture 5)

3. TEST BENCH: A RVD-LIKE EXAMPLE OPERATION APPROACH

Ground Test Case Scenario

In order to simulate a RVD-like example operation in the test bench and to prove this application, we have supposed that the robotic arm with the camera is the chaser and a moving person in front of the camera is the target; the target can be static or in motion. In order to get navigation data, the face of the person will be detected with suitable patterns.

Therefore, the object to search as target will be a human face. In a RDV operation the satellite docking pattern would be considered as a set of rectangles. This assumption is the same one that the Haar-like classifier does when it looks for a human face. The detection procedure is described in the following chapter. In the following pictures, there is shown a comparison between a satellite docking pattern with the same number and position of rectangles than in the case of a human face:

Fig. 2. Comparison between satellite docking and human face pattern

Test Bench Configuration

In order to test this concept, a dynamic test bench, composed mainly by a camera fixed to a robotic arm, has been set up at GMV premises. The hardware and software configuration of this test bench is described next in the following tables:

Table 2. Test bench hardware configuration

Hardware	Observations
Mitsubishi PA-10 robot	6 DOF, repeatability 0.1mm
Sony DFW-X700 digital camera	IEEE 1394, ½” progressive scan CCD, 1024 x 768, colour, 15fps
Pentax H1214-M lens	f12mm, F/1.4-16, FOV 35.69º
Matrox Meteor II/1394	Frame grabber for camera connection
Personal computers	2 PCs to control the machine vision system

Acronyms: DOF (Degrees Of Freedom), CCD (Charge-Coupled Device), FOV (Field Of View).

Table 3. Test bench software configuration

Software	Observations
Microsoft Windows 2000 & XP	Operating systems
Microsoft Visual C++ 6.0	Integrated development environment
MIL Lite 7.5	Libraries for image capture, display and archiving
Intel Open CV 1.0	Open source image processing libraries
CORBA ACE-TAO 5.4	Middleware for communications

Acronyms: MIL (Matrox Imaging Library), CV (Computer Vision), CORBA (Common Object Request Broker Architecture).

Test Bench Set-Up

The set-up of the test bench is shown below:

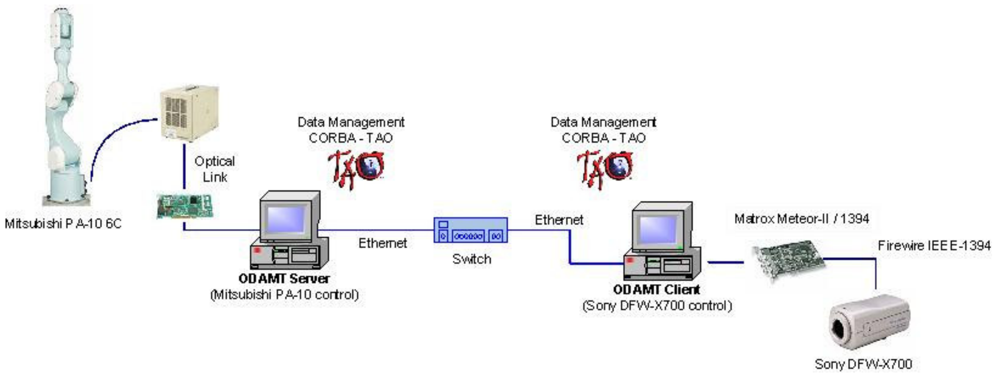

Fig. 3. Machine vision system hardware architecture

The Sony DFW-X700 camera is mounted as a tool on the 6^{th} arm of Mitsubishi PA-10 robot arm. The SW application is in charge of keeping always the camera focused on any human face under current detection. Once the application begins to work it enters into scanning mode. On this mode, the camera does a ± 90º scanning by rotating robot arm around its Z-axis in order to increase horizontal and vertical field of view up to 180° at least. This scanning mode is also invoked when the vision system looses the target.
If any human face is detected, the camera stops scanning mode and begins to track the human face. The application sends to the robot the position where the face it's located, and track the face while don't go out o the FOV. Based on the face tracking (position and attitude estimation with respect to our target), a control law (PD) is used to close the loop of the chaser (through the robotic arm) and align the chaser docking axis with the target position. The application performs this action computing the distance from the face centre to the frame centre and commanding a motion/rotation on this direction (through the robot arm). This motion analysis is performed in XY plane and the motion tracking is done by moving the robot arm around Z-axis and Y-axis.

Object Detection

First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundreds of sample views of a particular object (faces in this case), called positive examples, that are scaled to the same size (say, 20x20), and negative examples (arbitrary images of the same size).

After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a "1" if the region is likely to show the object (face), and "0" otherwise. To search for the object in the whole image the classifier procedure can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. Therefore, the scanning procedure must be done several times at different scales in order to look for an object of unknown size.

The word "cascade" in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). The basic classifiers are decision-tree classifiers with at least 2 leaves. Haar-like features are the input to the basic classifiers, and are calculated as described below:

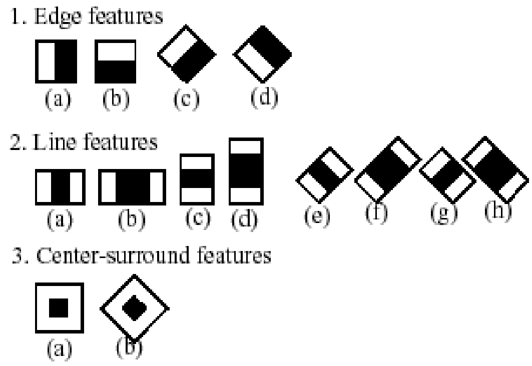

Fig. 4. Haar-like features which make the pattern or classifier

The feature used in a particular classifier is specified by its shape (1a, 2b etc.), position within the region of interest and the scale (this scale is not the same as the scale used at the detection stage, though these two scales are multiplied). For example, in case of the third line feature (2c) the response is calculated as the difference between the sum of image pixels under the rectangle covering the whole feature (including the two white stripes and the black stripe in the middle) and the sum of the image pixels under the black stripe multiplied by 3 in order to compensate for the differences in the size of areas. The sums of pixel values over a rectangular region are calculated rapidly using integral images.

Motion Analysis And Tracking

Once the object is detected, a motion vector is created from the central pixel of the image to the central pixel of the detected object, that is to say, the face (representing the docking port). The goal is to minimize this vector at every moment of time, controlling the object deviation from the central position of the image through the appropriate control commanding, providing to the user the tracking sensation. The conversion from the “world image” to the “world chaser” from pixels to degrees is as follows:

$$\vec{r}_x = \vec{p}_x \cdot \frac{FOV_x}{2 \cdot frame\_size_x}, \text{ where } FOV_x = 2 \cdot \arctan\left(\frac{CCD_x}{2 \cdot F}\right) \tag{3}$$

$$\vec{r}_y = \vec{p}_y \cdot \frac{FOV_y}{2 \cdot frame\_size_y}, \text{ where } FOV_y = 2 \cdot \arctan\left(\frac{CCD_y}{2 \cdot F}\right) \tag{4}$$

and where: CCD_x = camera sensor width (mm)
CCD_y = camera sensor height (mm)
F = lens focal length (mm)

The computed r vector is the angle value in degrees that the chaser must correct to maintain the target object centred in the image continuously, getting the object tracking. The command and control through the robotic arm is done by means of Cartesian coordinates associated to the two last joins (pitch and roll) to the robot: the pitch angle is associated to the Y-axis and the roll angle with X-axis.

For fast target motion, the time of operation of the application is critical and shall be compatible with the control loop:

$$operation\_time = capture\_time + detection\_time + analysis\_time + sent\_time$$
$$robot\_motion\_time \leq operation\_time \tag{5}$$

During our tests we have identified the interval times with a CPU Intel Core2 1.86Ghz 1Gb RAM shown in the next table. As expected the bottleneck in the whole process is the pattern detection. This process can be speed up if the pattern constraints are relaxed but then the number of false positives increments exponentially.

Table 4. Operation duration times in ms.

Operation	Duration time [ms]
Image capture	30
Pattern detection	220
Vector analysis	30
Commanding	10

The following figure illustrates a image sequence with the detection and tracking process. On the first image, the object is detected in the target and motion vector is created. Then, this motion vector is minimized until the object centre matches into the frame centre.

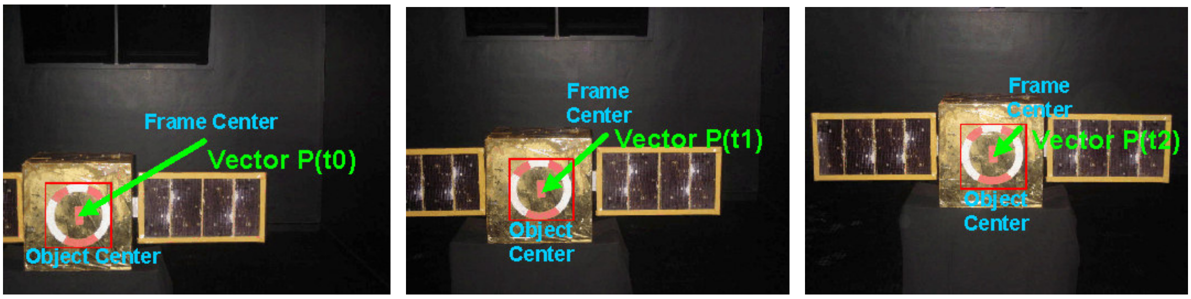

Fig. 5. From left to right, detection and tracking sequence of a satellite docking target

Software Application

The machine vision system software is based in a client-server application model:

- Client application: the software is in charge of image capturing, pattern detection, movement analysis and pattern tracking. This software runs in a standalone PC
- Server application: this application controls the robot motion. Through a CORBA interface it receives the new position command for the robot measured in degrees and it command the robot to this position by means of a velocity controller mode. This software runs in a second dedicated PC.
- As middleware between client and server we do use CORBA ACE-TAO. This middleware is oriented for real-time environments and it allows to synchronize different application running in different operative systems.

4. RESULTS AND CONCLUSIONS

This test bench has been tested first with many static images from individual and group human faces and then in real life with persons. The camera system started from the scanning motion looking for faces; when the software was able to detect a face the system start to track it and it was able to finalise the tracking in less than 5 s. In the following figure we can see the motion vector expressed in pixels in both Cartesian coordinates x and y. After some seconds the vector motion is almost reduced to zero and the tracking finalises.

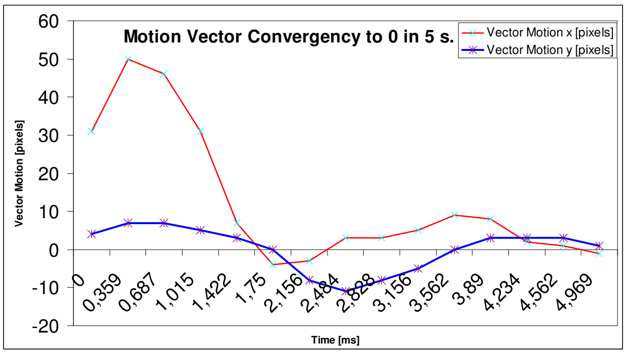

Fig. 6. Motion vector convergency graph

As conclusions, we can state that the camera navigation process has been achieved using pattern detection with the following performances:

- Success index: at least 95% of image detected faces are true faces. This index could be increased to a 99% increasing the complexity of the pattern detection process. As drawback, we have that the processing time is increased and then our test bench becomes too slow.
- Robustness: the maximum angular velocity that keeps tracking is about 12 deg/s. This angular velocity corresponds to a normal human walking. Below this velocity the test bench is able to keep the tracking without loosing at any time the detection of the pattern.
- Processing time: the whole processing time is around 300 ms and is mainly due to the pattern detection. This pattern detection should be reduced at least by a factor of 2 when using the docking pattern.
- System expandability: the use of CORBA-TAO middleware allows to plug-in additional elements very easily (hardware and software) in the implemented application, including the capability of detects a wide range of objects with new patterns.

Finally and taking into account the results obtained from the test bench, optical sensors and computer vision techniques are completely suitable for RVD operations.

REFERENCES

[1] Gerhard X. Ritter and Joseph N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, 2nd ed., CRC Press, 2001.

[2] John C. Russ, *The Image Processing Handbook*, 3rd ed., CRC Press, 1999.

[3] Bahram Javidi, *Image Recognition and Classification. Algorithms, Systems and Applications*, Marcel Dekker, Inc., 2002.

[4] BC-CARMS, *LiDAR – Overview of Technology, Applications, Market Features & Industry*, University of Victoria, BC, June 2006.

[5] Jacob Fraden, *Handbook of Modern Sensors,* 3rd ed., Springer, 2004.

[6] Paul Viola and Michael Jones, "Robust Real-Time Face Detection"*, International Journal of Computer Vision*, vol. 57, pp. 137-154, 2004.

[7] Rainer Lienhart and Jochen Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", *IEEE ICIP 2002*, vol. 1, pp. 900-903, September 2002.

[8] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Accepted Conference on Computer Vision and Pattern Recognition*, 2001.