Stereo vision system for capture and removal of space debris

Francesco Rosso, Francesco Gallo, Walter Allasia, Enrico Licata
EURIXGroup
Research Office
Via Carcano 26,
I-10153, Torino, Italy
Email: {familyname}@eurixgroup.com

Paolo Prinetto, Daniele Rolfo, Pascal Trotta
Politecnico di Torino
DAUIN
Corso Duca degli Abruzzi 24,
I-10129, Torino, Italy
Email: {name.familyname}@polito.it

Alain Favetto, Marco Paleari Paolo Ariano
Istituto Italiano di Tecnologia (IIT)
Center for Space Human Robotics
Corso Trento 21,
I-10129, Torino, Italy
Email: {name.familyname}@iit.it

Abstract

In order to enable the non-cooperative rendez-vous, capture, and removal of large space debris, automatic recognition of the target is needed. Several technologies are currently available and stereo vision is one of the most suitable in the strict context of space missions, where low energy consumption is fundamental and sensors should be passive in order to avoid any possible damage to external objects as well as to the chaser satellite (e.g., scattered reflection of laser scanners may potentially be an issue). In this paper we are presenting a stereo vision system we set up in order to reconstruct the object model of space debris, making use of a programmable system-on-chip board equipped with a couple of cameras. We identified the parameters that such a system have to deal with, and implemented a prototype solution tested in lab with debris mock up and actual satellite models. Results are demonstrating that fast image pre-processing is needed for having an acceptable recognition of object depth and shape. The proposed system can be integrated with other vision techniques to improve the comprehension of debris model allowing a fast evaluation of associated kinematics to select the most appropriate approach for capture.

Keywords

space debris; epipolar geometry; distance estimation; sift; image processing; fpga; IP-core;

I. Introduction and related technologies

Space debris is becoming a critical issue and many studies and analysis have been funded in order to identify the most appropriated approach for their removal. Among the others, the CADET (see Acknowledgement) project, partially funding the presented work, is focusing the attention to specific space debris, weighting about 2 tons and spanning about 10 meters. This class of orbiting debris is the most dangerous for aircrafts and satellites, representing a threat to manned and unmanned spacecrafts as well as hazard on Earth because large size objects can reach the ground without burning up in the atmosphere, and, in case of collision, thousands of small fragments can potentially be created or even worst triggering the Kessler syndrome [1].
Example of this class of debris is the lower stage of solid rocket boosters, such as the third stage of Ariane 4 [2], the H10 module, usually left from ESA [3] as space orbiting debris.
Removal missions are urgently needed and many solutions for rendezvous are currently under evaluation. The CADET project is proposing a non-collaborative rendezvous and innovative solutions are going to be studied. The work presented in this paper is related to the first phase of the removal mission, when the object has been detected and must be modeled. In order to have the needed information about the object to be removed three main operation must be performed. First, the 3D debris shape reconstruction, then the definition of the structure of the object to be removed and the type of material composing itself, and, finally, the kinematic model of the debris.
Since space applications impose several constraints regarding allowed devices, many devices commonly used for the 3D object reconstruction cannot be used (e.g., laser scanners and LIDARs [4], [5]). The device to be used must be characterized by very low power consumption. Moreover, they have to potentially be passive, not only for power constraints, but also because passive components are more robust against damages caused by unforeseen scattering of laser light.
The easiest (and probably cheapest) device suitable for space missions is the digital camera acquiring visible wavelengths. Either based on CCD or CMOS technology, their power consumption is affordable as well as the further processing of provided images. Digital cameras can be used for 3D reconstruction implementing several techniques. Stereo vision can be developed making use of a pair of identical cameras fixed on a well designed support. Due to the rotation motion of booster debris over central axis, silhouette [6] and shape from motion can be potentially candidate, also. Finally the 3D shape can be inferred from the Lambert's law [7] of reflectance and processing the shading stored in pixel depths. Even if we plan to evaluate the feasibility of all these techniques during the project lifetime, the first we have tested and presented in this paper is based on stereo vision.
In the following, Section II describes the state of the art about stereo vision techniques and related math; Section III is providing the overall system architecture and the integrated hardware details; Section III-B presents the 3D model reconstruction and

algorithms implemented; Section IV reports the results we have achieved with the proposed solution and, finally, Section V wraps up the work done and introduces candidate future work that we plan to evaluate.

II. Mathematical background

In the last years, the interest in creation of virtual worlds based on reconstruction of contents from the real world objects and scenes is increased more and more. Current research tends to study two different approaches: the image based and the model based. The first tries to deceive the observer with targeted image reprojections, while the second one aims at automatically building a (semi-) complete 3D model of the scene. This second group contains the stereoscopic technique. This technique mimics the human visual system with two (or more) points of view, and provides in output the so called dense map: an image that point out the distance from the observer of each pixel composing the input image.
This section provides a survey of the state-of-the-art procedures able to reconstruct a dense map, that allows to provide dense and full 3D reconstructions of objects from multiple views.
There are several steps that are required to compute dense map of objects from a set of images. These steps could be grouped in two main categories or "phases of execution": (1) calibration, one shot phase, and (2) effective dense map computing, real-time phase.

A. Calibration

Before starting the actual description about the calibration, a brief overview of epipolar geometry is required.
Epipolar geometry is used in stereo vision to limit the searching space when looking for matching points in both images. A point $P(x, y, z)$ in 3D space is seen by the left view, which we will call the source image, as a point $p(x^T, y^T)$, which is on the line between the left camera's focal point and point P. This line is seen by the right view, which we will call the search image, as a line. This is called an epipolar line. Given both the cameras internal and external calibration matrices and a point we can generate an epipolar line corresponding to this point in the search image. This constrains the search space to this 1D line. However, this means that for each pixel in the source image, we have to calculate the corresponding epipolar line in the search image. It would be much more convenient if each epipolar line was on the same line as the pixel it corresponded to. It is possible to transform the images in such a way that the epipolar lines are parallel and horizontal. This process is called rectification and is reached with the application at run-time of rectification matrices, output of calibration phase, on each input image.
Camera calibration defines the pose estimation of a camera in terms of interior and exterior orientation parameters. The term pose estimation in computer vision refers to determination of the camera position and orientation using correspondence of 3D reference points and their images. This task can be accomplished using a technique that compute the camera position and orientation with regards to a known object using 4 or more coplanar feature points. Commonly, in order to have artificial coplanar feature points, the known object is represented by a chessboard, but it is also possible to extract natural feature points from images with a scale-invariant feature transform algorithm.
The more is the number of extracted features, the greater will be the accuracy of the dense map, but at the same time, the higher will be the time to compute the calibration.
The output of this first step is composed by a set of matrices describing (1) the interior orientation parameters, (2) the exterior orientation parameters and (3) the rectification matrices to be applied to input images in order to overlap features points and get horizontal epipolar lines.

B. Dense map computing

This second real-time phase is composed by different steps: (1) application of rectification matrices, (2) search of features on epipolar lines and (3) distance estimation.

- **Rectification**
  This first mandatory step aims at overlapping features of the input images in order to let them "comparable" through the epipolar geometry. It is a simple but time-consuming step that moves pixels taking them from one place in the original image and locating them in another position in a new rectified image.

- **Search of Features**
  This second step is able to find the matches between the features extracted from the left and right images.
  Literature proposes a series of methods to solve this problem (i.e., *Stereo Global Matching*, *Stereo Block Matching*, etc.). Such methods present a lot of difficulties related to photometric variations, image sensor noise, specularity, foreshortening and the uniqueness constraint, perspective distortions, textureless regions, repetitive structures and textures, reflections, transparency, occlusions.
  In this first phase of the project, we decided to evaluate only the disparity of the features extracted with SIFT [8] algorithm to avoid listed difficulties. The features set has been manually postprocessed in order to validate the features correspondence extracted automatically.

- **Distance Estimation**
  The Distance between the observer (i.e., two cameras composing the stereovision system) can be computed exploiting

the dense stereo approach.
Dense stereo combines the two images obtained from the rectification process and takes the position of the pixels in the left image to output the corresponding pixel location in the right image.
Triangulation, as shown in Fig. 1, requires knowing the focal length of the camera (f), the distance between the camera bases (b), and the center of the images on the image plane (c1 and c2). From these information it can be computed the dense map, in which points closer to the camera are almost white whereas points further away are almost black. Points in between are shown in gray-scale, which get darker the further away the point gets from the camera. Disparity (d) is the difference between the lateral distances to the feature pixel (v2 and v1) on the image plane from their respective centers. Using the concept of similar triangles, the distance from the cameras (D) is calculated as D = b * f / d.

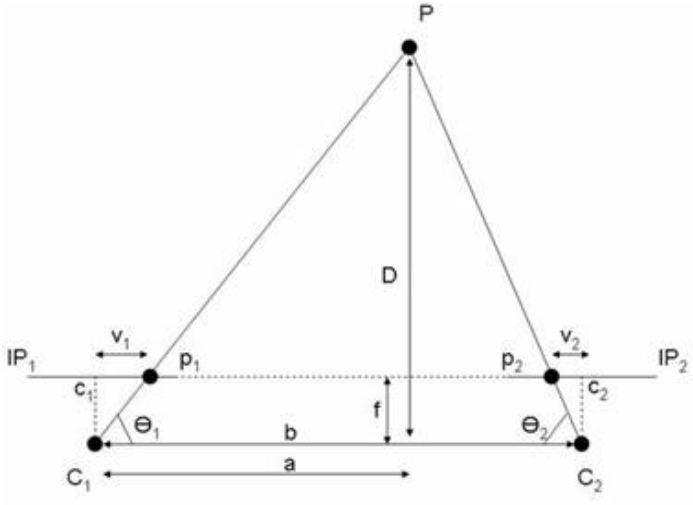

Figure 1: Trigonometric for Distance Estimation from Disparity

Within this paper, the automatic dense map generation is not performed, while a manual evaluation of the disparity between matching SIFT features has been made. This approach gives us the possibility to evaluate the exact number of valid features.

III. Proposed Architecture

The proposed architecture receives in input two images from a stereovision camera (i.e., a camera that provides in output two images of the same scene taken at the same time from two different points of view), and provides in output the Debris 3D-model. The Debris 3D-model is provided in output as *X3D* file. The *X3D* file is an ISO standard XML-based file format for representing 3D computer graphics.
Since the current prototype of the proposed architecture is composed by cooperative hardware and software routines, this first prototype has been developed on the *Xilinx Zynq-7000 System-on-Chip (SoC)*.
This SoC integrates on the same chip a tightly coupled ARM processing system and 7 series programmable logic. For these reasons this architecture is able to fit all the requirements of the modern embedded systems, in terms of low system cost, high performance, and greater flexibility.
Fig. 2 shows the internal architecture of the current prototype, highlighting the modules implemented in the reconfigurable logic region and the ones implemented as software routines running on the ARM processor.
The stereovision system is composed of two cameras, that provide grayscale images with a 640x480 pixels resolution.
The two data streams from the stereovision camera are separately managed by two input controllers (i.e., *Input Controller* in Fig. 2) implemented in the reconfigurable logic. Each input controller manages the communication between the stereovision camera and the system, through a USB Controller. In addition, each *Input Controller* organizes the input pixel stream in 32-bit packets. Since the the pixelwise resolution of the stereovision camera is 8 bit, every output packets from the controller contains 4 pixels. The output packets associated with the right and left image are provided in input to the *Right Image Preprocessing* and the *Left Image Preprocessing* (see Fig. 2), respectively.
Since the illumination conditions in the space cannot be predicted a priori and, at the same time, the proposed architecture must be always able to properly work, an image preprocessing is required. The preprocessing aims at enhancing the quality of input images in terms of illumination and contrast. This operation allows to increase the features extraction capability of the 3D model also in bad illumination conditions.
This task can be performed exploiting *spatial-domain* image enhancement techniques. Among the available spatial-domain techniques, the *Histogram Equalization* [9] is the best one to obtain a high contrasted image with an uniform tonal distribution. This technique modifies the intensity value of each pixel to produce a new image containing equally distributed pixel intensity values. Thus, the output images have similar tonal distribution, reducing the effect of the illumination variations.

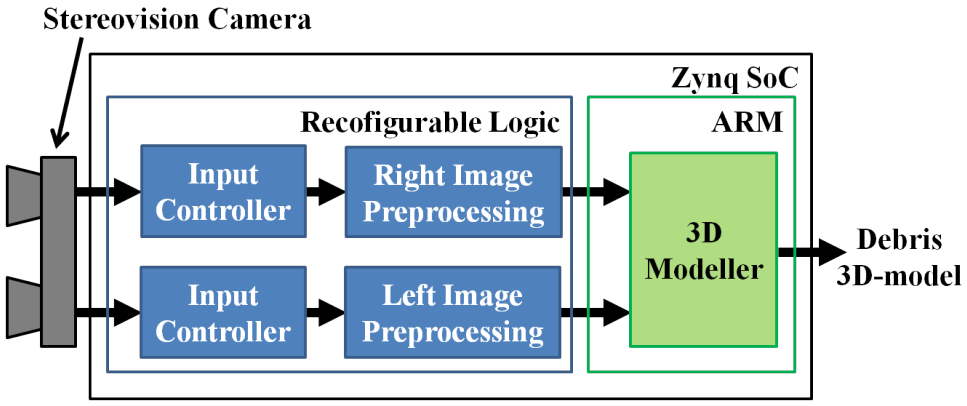

Figure 2: Proposed architecture

However, this technique does not work well in every condition. In facts, it works properly on images with backgrounds and foregrounds that are both bright or both dark (smoothed image histogram), but becomes ineffective in the other cases.
If the image histogram is peaked and narrow, the Histogram Stretching [9], that allows to redistribute the pixel intensities to cover the entire tonal spectrum, provides better results. Instead, if the image histogram is peaked and wide, it means that the input image already has a good level of details and it contains an object on a solid color background (i.e., the image can be provided in output without modification).
Thus, in order to design a system able to autonomously work, the pre-processing system must be able to manage these three different cases, and to provide in output the best image, only.
This task can be accomplished exploiting SAFE [10]. SAFE is a high performance FPGA-based IP-core that is able to enhance an input image autonomously selecting the best image enhancement technique (i.e., HE, HS, or no enhancement) to be applied. Finally, the left and right enhanced images are provided in input to the 3D modeler.

A. Preprocessing System

As aforementioned, the two image preprocessing systems consist in two SAFE instances. SAFE is a high performance FPGA-based IP core that receives in input a grey scale image and outputs the enhanced image.
This IP-core receives the input pixels from the Input Controller (see Fig. 2) through a 32-bit input interface (i.e., four pixels can be received each clock cycle). In addition, SAFE receives in input two parameters: *HW* and *BW*. *HW* defines the threshold associated with the histogram width (i.e., the distance between the minimum and maximum intensity inside the image histogram). *BW* defines the threshold referred to the difference between two consecutive image histogram bar (HB) values. These two parameters are required for automatically selecting the best image to be provided in output (i.e., equalized image, stretched image, or input image without modifications), depending on the input image statistics.
Fig. 3 shows the internal architecture of SAFE.

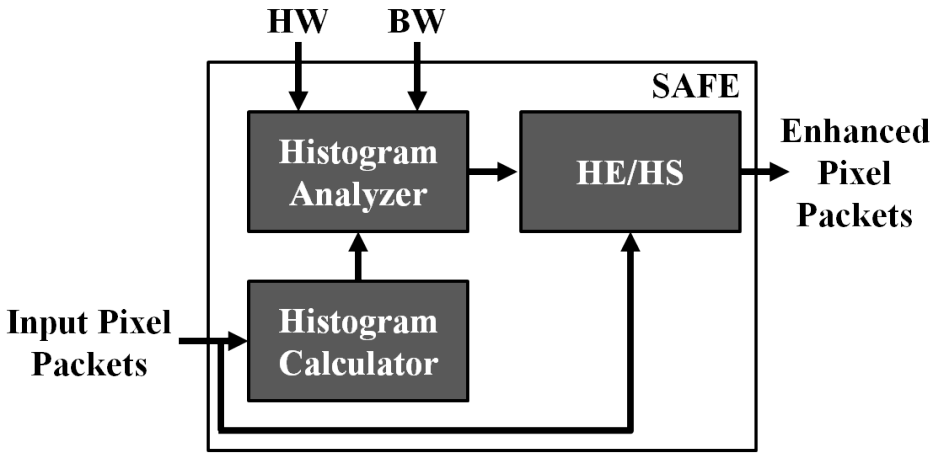

Figure 3: SAFE block diagram

The *Histogram Calculator* counts the occurrence of each pixel intensity, in order to compute the histogram bar values. In this

way, when a complete image has been received, it is able to provide in output the histogram associated with the received image. The *Histogram Analyzer* performs the operations reported in Alg. 1. Basically, it analyzes the image histogram in order to extract: the maximum difference between two consecutive histogram bars (*max_BW* in Alg. 1), and the histogram width (*real_HW* in Alg. 1). This two values are compared with the input thresholds (i.e., *HW* and *BW*), in order to select the best image to be provided in output.

Algorithm 1 Histogram Analyzer operations

$found \leftarrow FALSE$
$max\_BW \leftarrow 0$
$previous\_HB \leftarrow HB(0)$
for $i = 0 \rightarrow 255$ **do**
  if $(HB(i) \neq 0) \wedge (found = FALSE)$ **then**
    $min\_HB \leftarrow HB(i)$
    $found \leftarrow TRUE$
  end if
  $actual\_BW \leftarrow \mid HB(i) - previous\_HB \mid$
  if $actual\_BW > max\_BW$ **then**
    $max\_BW \leftarrow actual\_BW$
  end if
  $previous\_HB \leftarrow HB(i)$
end for
$found \leftarrow FALSE$
for $((i = 255 \rightarrow 0) \wedge (find = FALSE))$ **do**
  if $HB(i) \neq 0$ **then**
    $max\_HB \leftarrow HB(i)$
    $found \leftarrow TRUE$
  end if
end for
$real\_HW \leftarrow max\_HB - min\_HB$
if $(real\_HW < HW) \wedge (max\_BW > BW)$ **then**
  $Output\_image \leftarrow Stretched\_Image$
else
  if $(real\_HW > HW) \wedge (max\_BW > BW)$ **then**
    $Output\_image \leftarrow Input\_Image$
  else
    $Output\_image \leftarrow Enhanced\_Image$
  end if
end if

The *Equalizer / Stretcher* performs both HE and HS on the input image, but it provides in output only the best image (i.e., equalized image, stretched image, or input image without modifications) depending on the information provided by the *Histogram Analyzer*.
In addition, in order to increase the timing performance and to avoid the internal buffering of the image, the output image is selected on the basis of tone statistics gathered from the previous image (i.e., histogram width and bar width associated to the previous processed frame). Since the input frame rate is relatively high (i.e., 30 fps), this choice introduced a negligible error (i.e., lower than 0.39% on the intensity of the pixels).
For the sake of completeness, in the sequel, the mathematical operations required by the two enhancement algorithms are summarized.
The HS transformation function is reported in (1).

$$I_{Stretched}(x, y) = C \cdot (I(x, y) - min\_HB) \tag{1}$$

where $I_{Stretched}(x, y)$ is the stretched pixel intensity in the (x, y) position, $I(x, y)$ is the pixel intensity in the (x, y) position, min_HB is the minimum intensity in the previous image histogram, and C is a scale factor equal to:

$$C = \frac{2^{bpp} - 1}{max\_HB - min\_HB} \tag{2}$$

where max_HB is the maximum intensity value in the previous image histogram and bpp is the pixel resolution.
The transformation function performed during HE is:

$$I_{Equalized}(x, y) = k \cdot \sum_{j=0}^{I(x,y)} HB_j \tag{3}$$

where $I_{Equalized}(x, y)$ is the equalized pixel intensity in the (x, y) position, HB_j is the value of the j-th HB and k is equal to:

$$k = \frac{2^{bpp} - 1}{1024 * 1024} \tag{4}$$

B. Features extraction for distance estimation

In this first prototype, the component for distance estimation is implemented in software (not FPGA), exploiting some primitives contained in the OpenCV library.
Once the calibration step is completed (it is not the purpose of this paper to describe the calibration phase), two couples of matrix are given. Such matrices are of the same size of the input images and contain the distortion coefficients to be applied to each pixel on both *x* and *y* axes. The output of this process is composed by a couple of images with perfectly horizontal and overlapped epipolar lines.
On this couple of images is now optionally possible to apply a canny filter in order to reduce pixels correlation due to diffused light (see Section IV).
The extraction and matching of SIFT features is now performed. Thanks to the epipolar geometry and the rectification a constraint on the *y* value of the matching features position is feasible, allowing to strongly reduce the number of false positive matches.
As final step, the disparity, i.e. the difference of *x* value of the matching features position, can be computed.

IV. Experimental Results

The experimental results reported in this section aims at demonstrating: (1) the high performances achieved by the hardware implementation of the adopted image preprocessing architecture, and (2) the matching capability of the proposed architecture under different illumination conditions.

A. Preprocessing System

The proposed architecture has been implemented on a *ZedBoard*, that is a low cost development board for the Xilinx Zynq-7000 SoC. This board is equipped with a *Zynq-7000 SoC XC7Z020-CLG484-1* device, that embeds a dual-core *ARM Cortex-A9 processor* and a *Xilinx 7-series* reconfigurable logic region.
The image preprocessing module, i.e., SAFE, has been synthesized on the reconfigurable logic region of the Zynq-7000 SoC. Table I shows the area occupation of each internal module composing SAFE.

Table I: Resource Usage for *XC7Z020-CLG484-1*

Module	**FPGA Area Occupation**			
	FFs	**LUTs**	**BRAMs**	**DSPs**
Histogram Calculator	216 (0.20%)	265 (0.41%)	4 (1.43%)	0 (-)
Histogram Analyzer	314 (0.30%)	219 (0.41%)	2 (0.71%)	0 (-)
HE/HS	788 (0.74%)	434 (0.82%)	4 (1.43%)	20 (9.1%)
Total	1,318 (1.24%)	918 (1.73%)	10 (3.6%)	20 (9.1%)

Obviously, since in the proposed architecture two instances of SAFE are required (see Fig. 2), the total required logic resources are two times the data reported in the above table.
The overall number of clock cycles required to preprocess a complete input image is equal to 262,656, and the maximum working frequency achievable by SAFE on the Zynq-7000 SoC is 70 MHz. Thus, the preprocessing task can be completed in 3.75 ms. This data demonstrates that the delay introduced by SAFE is negligible and it allows to far exceeds the typical frame rate of 25 fps of the other building blocks, and of the camera, as well.

Figure 4: H10 1:10 model

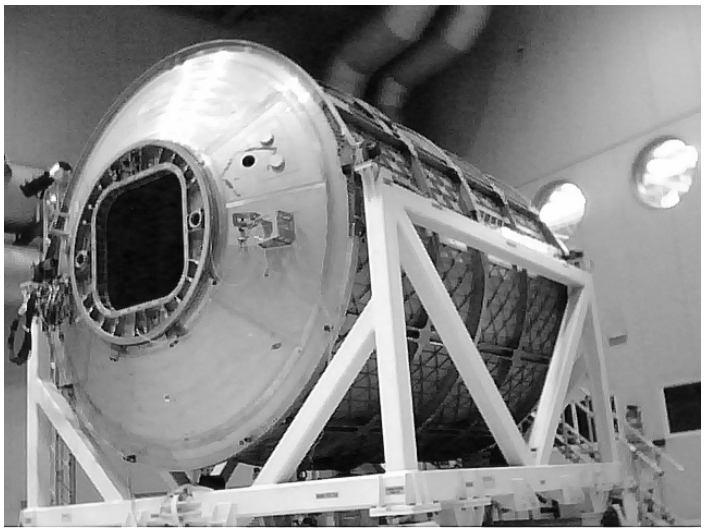

Figure 5: ISS 1:1 module. Courtesy of ALTEC and ASI

B. Features Extraction for Distance Estimation

The performances of the proposed architecture have been evaluated on two different images. The first one (Fig. 4) represents a H10 model caught with direct light, while the second one (Fig. 5) reproduce ISS real module illuminated with diffused light.

Moreover, on the ISS real model image, a canny prefilter has also been applied in order to decorrelate pixels from each other, due to the diffused illumination.
From each test image the SIFT matching features have been extracted before and after the SAFE preprocessing.
The number of manually validated SIFT matching features and the ratio between the number of features recognized with and without SAFE are reported in Table II.
It can be noticed that the number of validated SIFT matching features strongly increases with an image preprocessing such as SAFE. As a consequence, the increased number of matching features allows to increase the accuracy of the distance estimation.
For the sake of completeness, Figures from 6 to 11 show the adopted test images with the associated matching features.

V. Conclusion

This paper presented an architecture that could be potentially adopted to perform the 3D-model computation of a space-debris. The presented experiments demonstrate the benefits when extracting SIFT features from images enhanced with SAFE. The main one is the increased number of matching features and, as a consequence, the increased accuracy of the distance estimation between object and cameras.

Table II: Number of valid SIFT features

#features		*noSAFE*	*SAFE*	*ratio*
H10		5 (Fig. 6)	22 (Fig. 7)	4.4
ISS	*noCANNY*	721 (Fig. 8)	1208 (Fig. 9)	1.7
	CANNY	58 (Fig. 10)	140 (Fig. 11)	2.4

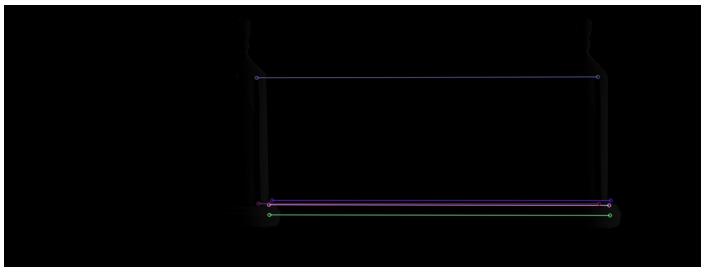

Figure 6: H10 valid SIFT features.

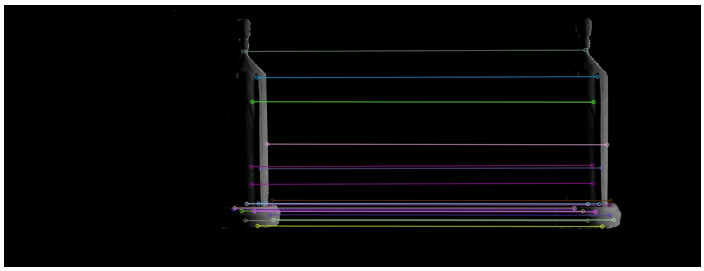

Figure 7: H10 enhanced with SAFE valid SIFT features

Figure 8: ISS valid SIFT features. Courtesy of ALTEC and ASI

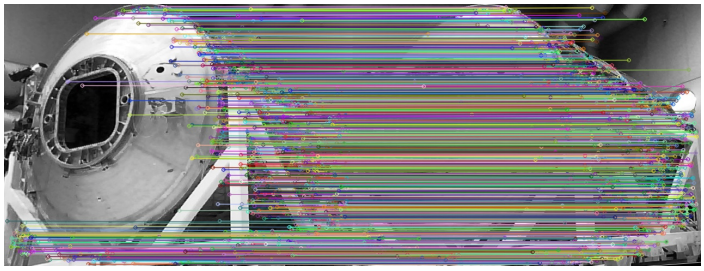

Figure 9: ISS enhanced with SAFE valid SIFT features. Courtesy of ALTEC and ASI

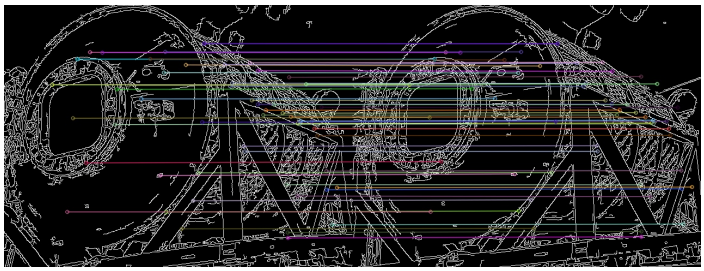

Figure 10: ISS valid SIFT features. Courtesy of ALTEC and ASI

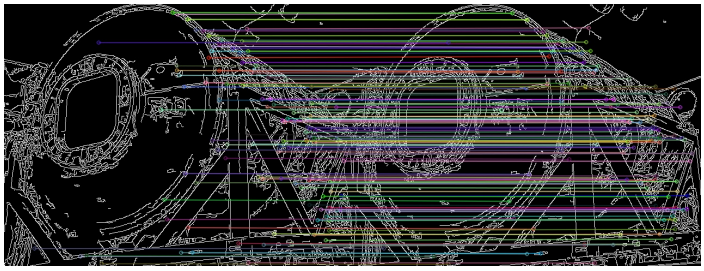

Figure 11: ISS enhanced with SAFE valid SIFT features. Courtesy of ALTEC and ASI

The self adaptivity to different environmental conditions and the negligible delay introduced by the SAFE FPGA-based implementation allow to meet the requirements of space missions in terms of flexibility and real-time behaviour. Moreover, the low area occupation of SAFE allows to exploit the free hardware resources to integrate in the FPGA device other IP-cores able to compute tasks currently implemented in software. In this way, the timing performances of the entire system could be enhanced.

ACKNOWLEDGMENT

The presented work has been partially supported by CADET, a project co-funded by Regione Piemonte according to call for proposal: POR FESR 2007/2013, linea di attività I.1.1. "Piattaforme innovative" AEROSPAZIO FASE II. The authors want to thank ALTEC [11] and ASI [12] for their availability and assistance in acquiring images for the ISS model.

REFERENCES

[1] K. Donald, "Collisional cascading: The limits of population growth in low earth orbit," *Advances in Space Research*, vol. 11, no. 12, pp. 63–66, 1991.

[2] W. N. Ch. Bonnal, "Ariane debris mitigation measures: Past and future," *Acta Astronautica*, vol. 40, no. 2–8, pp. 275–282, 1997.

[3] ESA, "The european space agency." http://www.esa.int/Our_Activities/Launchers/Ariane_42.

[4] L. H. A.P. Cracknell, *Introduction to Remote Sensing*. Taylor and Francis, 1991.

[5] E. Naesset, "Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data," *Remote Sensing of Environment*, vol. 80, no. 1, pp. 88–99, 2002.

[6] A. Rivers, F. Durand, and T. Igarashi, "3d modeling with silhouettes," in *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pp. 1–8, 2010.

[7] B. L. V. K. Fuwa, "The physical basis of analytical atomic absorption spectrometry. the pertinence of the beer-lambert law," *Analytical Chemistry*, vol. 35, no. 8, pp. 942–946, 1963.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, (Washington, DC, USA), pp. 1150–, IEEE Computer Society, 1999.

[9] R. Lakshmanan, M. Nair, M. Wilscy, and R. Tatavarti, "Automatic contrast enhancement for low contrast images: A comparison of recent histogram based techniques," in *Proc. of 1st International Conference on Computer Science and Information Technology (ICCSIT)*, pp. 269–276, 2008.

[10] S. Di Carlo, G. Gambardella, P. Lanza, P. Prinetto, D. Rolfo, and P. Trotta, "Safe: a self adaptive frame enhancer fpga-based ip-core for real-time space applications," in *Proc. of 7th International Design and Test Workshop (IDT)*, 2012.

[11] ALTEC, "Advanced logistics technology engineering center." http://www.altecspace.it/en/. Last visited: 24 Apr. 2013.

[12] ASI, "The italian space agency." http://www.asi.it/en. Last visited: 24 Apr. 2013.