ADVANCED VISION SYSTEM FOR EXPLORATION BY A PLANETARY AEROBOT

Maarten Vergauwen¹, Glyn Matthews¹, Luc Van Gool¹ and Bernard Fontaine²

¹ESAT-PSI, K.U.Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Email: firstname.lastname@esat.kuleuven.be

²Space Applications Services
Leuvensesteenweg 325, B-1932 Zaventem, Belgium
Email: bernard.fontaine@spaceapplications.com

1 INTRODUCTION

Aerial platforms can become an integral part of surface exploration missions on planets or moons with an atmosphere, like Venus, Mars or Titan. One of the most immediate applications for aerobots is ultra-high resolution imaging over extensive areas of the planet. Planetary Aerobot missions could prove very useful in the automatic detection of geological features and selection of possible landing sites for a planetary exploration mission. The Aerobot system can travel across many areas on the planet and send the appropriate data back to Earth. Unfortunately the bandwidth available for data transmission from the planet to Earth is very limited. These bandwidth restrictions would require, if the Aerobot System were to transmit all collected imagery of the planet, heavy compression on the images. This compression would definitely hamper the scientists on Earth to determine the interesting areas. The computer vision algorithms used to reconstruct the terrain in 3D would be disadvantaged as well. It is therefore imperative that the Aerobot System has some degree of autonomy and can perform computer vision operations on its own which makes it possible to detect interesting geological features on the spacecraft. During the execution of its mission, the Aerobot System has access to the uncompressed imagery, taken by its camera(s). It is therefore recommended to perform all critical computer vision processing on the system itself before the images are polluted by compression. The generation of Digital Elevation Maps is clearly a computer vision process that suffers from compression. Generation of these maps on the Aerobot and sending them, together with the compressed reconstructed parts of the images, will lead to a much better understanding of the observed areas, for the same amount of expended bandwidth.

In this paper, an Imaging and Localization Package (ILP) is described which is capable of performing the computer vision processing described above¹. All data collected by the Aerobot needs to be correlated with the position in which the measurement was acquired. On long duration missions, the Aerobot can not rely on localization performed by an orbiter or from ground; it must have its own means. During the last decade the computer vision community has made tremendous progress in acquiring 3D information from images taken by uncalibrated cameras, while at the same time self-calibrating the camera. The ILP makes use of these algorithms to compute both the calibration of the camera and the 3D reconstruction of the terrain. The specifics of an Aerobot mission, like almost linear motion of the camera, almost planar terrains, etc, require changes and new techniques in the reconstruction pipeline. Once calibration and reconstruction have been computed, scoring techniques can be applied to the data which now comprises of not only the images but of 3D information as well. The scoring algorithms should be written such that high scores for an image correspond to a large chance on interesting geological features to be found in that specific image. These scores can

¹ The work described in this paper was performed under ESA contract 17172/03/NL/CH
then be taken into account to decide on an appropriate compression factor, compression scheme and broadcast priority for the images.

2 OVERALL AEROBOT SOFTWARE SYSTEM

The overall layout of the components of the Aerobot system that are relevant to this paper is shown in fig. 1. Large boxes depict subsystems, rectangles with rounded corners depict commands and normal rectangles represent data that is sent from one component to another.

The Aerobot is equipped with a camera system that records the planetary terrain underneath. The Imaging and Localization Package requests images from this camera. Images are either taken at regular intervals or at specific times, determined by the ILP (see paragraph 3.1). The ILP then processes these images and generates 3D data comprising the camera positions and Digital Elevation Maps of the respective images (see paragraph 3.2). This data is sent to the Storage Manager. This component is in charge of managing the acquired and computed data with respect to the available storage on board. In order to do so, it makes use of functionality provided by the ILP which compresses the data based on an Image Richness Index (see paragraph 4.1). Whenever the Aerobot is in range, it can transmit data to the satellite. Upon successful transmission, the transmitted data can be deleted from the local storage.

The core of the Aerobot system is the Imaging and Localization Package or ILP. It is in charge of several tasks that have to do with image processing. Among these tasks are:

1. asking for images and/or determining which images taken by the camera are suited for further processing
2. processing given images and computing camera poses and 3D information in the form of Digital Elevation Maps
3. detecting relevant information in the images and DEMs and compressing the data.

The first two tasks together lead to a 3D reconstruction of the Aerobot trajectory and planetary terrain and will be explained in section 3. The third task is needed by the Storage Manager and is described in section 4.1.

3 ILP AND 3D RECONSTRUCTION

The core business of the Imaging and Localization Package is the computation of 3D information from the images that are taken by the on-board camera. Fig. 2 schematically shows how different subcomponents collaborate to deliver this result. All components are connected to a local database system. The advantage of this design is that local data is not unnecessarily duplicated. Current database systems furthermore implement notification techniques that can trigger other components when new data is available. The setup of the ILP heavily makes use of this feature. It also allows us to run the different components on different machines that are connected to the database, thus increasing the processing speed.

The sole input data that is used by the ILP consists of images taken by the Aerobot camera. The ILP Overlap Checker is in charge of determining which images will be used by the other components. It does so based on the overlap of this image with the previously selected image. Paragraph 3.1 describes this functionality in more detail. When the first two images have been selected by the ILP Overlap Checker, the ILP Structure And Motion (or SaM) component is automatically noticed by the local database and starts processing these images. This component is in charge of computing both the motion of the camera in the form of cameras for every image and a description of the structure of the observed scene in the form of a set of 3D points.
Paragraph 3.2 describes this state-of-the-art component. When cameras are available for two or more images, the ILP Dense Matcher is noticed by the local database and computes dense Digital Elevation Maps for the next image. This is done by dense stereo matching between the new and previous image. Mask files are automatically generated that indicate which part of the image has not been observed by previous images and is therefore crucial to be kept by the Storage Manager. More information on the ILP Dense Matcher is given in paragraph 3.3.

![Figure 2](image)

**Figure 2:** A schematic view of how the different subcomponents of the ILP collaborate. Images, received from the Aerobot Camera are the only input of the system. A local database connects all subcomponents.

### 3.1 ILP Overlap

Structure and Motion Algorithms expect sequences of images as input of their processing, as will be described in paragraph 3.2. Consecutive images in such sequences should not differ too much since matching them then becomes a very hard problem. Nor should they be too similar, i.e. recorded too close to each other, since this setup typically causes large errors in the resulting calibration and 3D reconstruction. Success of the SaM component therefore greatly depends on the overlap between consecutive images. This is the reason why we decided to implement a first component which is capable of deciding quickly whether a newly recorded image has the right amount of overlap with the previously selected image.

The ILP Overlap component is initialized with the first recorded image and a target overlap percentage, typically around 80-85%. When a new image arrives the component computes the overlap between this image and the first image. If the overlap is at or below the target percentage, the new image is selected as the next image to be processed by the SaM component and thus inserted in the local database. If the overlap is larger, two actions can be taken. If images are recorded by the Aerobot camera at regular intervals, the next image is simply awaited and the overlap between the previously selected and the new image is computed. If the Aerobot camera can be commanded to record an image at a specific time, the optimal time at which we expect to record an image with the target overlap percentage can be easily computed by extrapolating from the previously computed overlaps and time-intervals between the images.

The overlap detection algorithm is based on comparing the two images under scrutiny with a comparison function like Normalized Cross Correlation (NCC). Fig. 3 shows a schematic overview of the algorithm. First the center part of the first image is selected. A window with the same size slides over the second image and the NCC value of both windows is evaluated. From the window position for the largest NCC value, it can be computed how many pixels the image content is shifted from the first to the second image, and thus the overlap value can be calculated. This is of course only an approximate value since effects like occlusion will have an impact on the accuracy. Speed of the algorithm is an issue, especially if we want to compute the next optimal time for recording. The images are therefore subsampled a couple of times. A beneficial side-effect of the subsampling is the elimination of the danger of local maxima of the NCC values.

For simulation and testing purposes, all components of the Aerobot system run on ordinary personal computers. If not enough processing power is available, other solutions must be sought. With the increasing programmability of commodity graphics processing units (GPUs), these chips are capable of performing more than the specific graphics computations for which they were designed. They are now capable coprocessors, and their high speed makes them useful for a variety of applications [1, 2]. Computing the overlap between images boils down to comparing one image to another, shifted over many positions. Graphic cards are very good at performing such comparisons efficiently. The comparison function that is employed now becomes Sum of Intensity Differences since a
full NCC can not be implemented efficiently on the GPU.

Figure 3: Schematic overview of the overlap algorithm. First the center part of the first image is selected. A window with the same size slides over the second image and the NCC value of both windows is evaluated.

3.2 ILP Structure And Motion

The general problem of structure and motion recovery can be stated as: Given a set of images or a video sequence, compute the (internal and external) camera calibration for all views and the 3D reconstruction of the scene that is visible in these views. A possible solution to this problem that computes the result completely automatically has been developed in the ESAT-PSI lab [3] and comprises the following steps:

- First the images are pairwise related to each other. Feature points are extracted in every image and matches are found between consecutive images by comparing these features using a comparison function like Normalized Cross Correlation.
- For every consecutive pair of images the epipolar geometry is computed from the previously computed matches. Outliers are detected and removed.
- The 3D structure of the scene and the calibration of the camera is initialized for the best suited pair of images. All other cameras are consecutively computed in the same coordinate frame and the 3D pose of the feature points is computed. The resulting reconstruction is valid up to any projective transformation.
- The projective reconstruction is upgraded to metric (Euclidean up to scale) using self-calibration and a bundle adjustment procedure minimizes the total reprojection error of all points in all cameras.

3.2.1 Planar Degeneracy

Standard structure and motion algorithms, as described in the previous paragraph, first relate all images in a projective frame and then upgrade this frame through self-calibration. Unfortunately this approach suffers from the existence of critical motions [4] and surfaces [5]. This means that there are cases for which sequences are recorded in a specific fashion such that multiple solutions to the self-calibration procedure exist. Since only one of these solutions corresponds to the real world and is the one we look for, and since this solution can not be discerned from the other solutions, the result of the pipeline will in general not be correct. Examples of critical motion sequences are a motion on a line or a circle. A typical example of a critical surface sequence is one in which only a single plane is visible. The planetary testbed at ESTEC on which tests have been performed is not exactly planar but there are quite some parts where almost no 3D information outside the ground plane is available, which will certainly cause problems for standard structure and motion.

In recent years different solutions have been proposed to deal with planar structures. They vary from merely surviving the plane by detecting when only a plane is visible in the images, calibrating and reconstructing in parts where more than this plane is visible and extending this structure to the planar part [6], to effectively dealing with the planar structure (or other critical motions and surfaces for that matter) by taking into account more information on the camera intrinsic parameters than in the general structure and motion algorithm [7].

The first approach will not help us in this case because it needs some cameras of the sequence to observe more than a plane which is not guaranteed. This approach is also computationally intensive and error prone for planes are typically detected in the images by means of some model selection algorithm. Structures that are nearly but not exactly planar, like some parts of the ESTEC planetary testbed, cause problems for this approach.

Therefore the second approach has been implemented. Since it assumes the intrinsics of the camera to be known, one can estimate the
essential matrix instead of the fundamental matrix when relating two views. This essential matrix takes into account the intrinsic camera parameters and, unlike the fundamental matrix, is unique for all practical cases, even if all observed points lie in a single plane. Pose estimation of the cameras (the third step in the standard SaM process) can then be done in a metric frame which alleviates the problems we had with general structure and motion. In order to retrieve the intrinsic parameters different strategies can be envisaged. The camera can be calibrated beforehand by means of a calibration target. A more elegant approach is described below.

- Take a set of 4 or 5 consecutive images, output of the ILP Overlap Checker.
- Compute both the best fitting fundamental matrix and planar homography between every consecutive pair.
- Determine which of the two models fits the data best using a model selection algorithm, like GRIC (Geometric Robust Information Criterion) [8].
- If the fundamental matrix is clearly the better model for all pairs, the sequence of images can be processed with our normal uncalibrated SaM algorithms.
- The calibration we employ for all images from now on is inferred from the resulting camera calibration of this sequence.

3.2.2 Windowed Bundle Adjustment

One of the algorithms that is employed in step 4 of a the typical SaM pipeline is a bundle-adjustment step. This is in fact a large non-linear optimization of all reconstructed 3D points and cameras and aims to minimize the global reprojection error in the images. Small accumulated errors can be reduced by means of this algorithm. Unfortunately, since a bundle-adjustment step changes both points and cameras, such a step is typically executed at the end of the SaM process when the information for all images has been computed. In the case of an Aerobot that continuously records images, such a final phase never arrives. Omitting the bundle-adjustment step is not a good idea since it has an important regulating influence on the calibration result. That is why we implemented a windowed version of this bundle adjustment.

Since the Aerobot typically moves forward all the time and in general doesn't revisit the same area, and since there is an overlap of 80-85% between consecutive images, after approximately 5 images no information of the first image is visible in the latest image. That is why we only minimize the points and cameras of the last 5 images that have been processed by the SaM pipeline. All cameras computed before remain fixed. This strategy gives us the best of both worlds: cameras are computed continuously (with a delay of 5 images) and a bundle-adjustment is still performed.

3.2.3 Structure and Motion Results

For the automatic structure and motion algorithm to succeed, the input images should not differ too much from each other. Techniques exist to deal with wide-baseline cases but these are not applicable here since the required resources and processing time are much larger than for the normal approach. Furthermore, the ILP Overlap Manager of paragraph 3.1 makes sure that consecutive images are not too far apart.

3.3 ILP Dense Matching

The windowed bundle adjustment step of paragraph 3.2.2 delivers cameras that will not be updated anymore. Since the calibration is now computed, the calculation of dense Digital Elevation Maps (DEMs) can start. To this end, the latest pair of cameras can be rectified so that epipolar lines become horizontal and coincide with scanlines [9]. Thereafter a stereo algorithm is employed on these rectified images. The dense correspondence scheme we employ to construct the disparity maps is the one described in [10]. It operates on rectified image pairs and incorporates some extra constraints. The matcher searches at each pixel in the left image for the maximum normalized cross correlation in the right image by shifting a small measurement window along the corresponding
scan line. Matching ambiguities are resolved by exploiting the ordering constrain in the dynamic programming approach. The algorithm was adapted to yield sub-pixel accuracy by employing a quadratic fit of the disparities. The DEM for this image pair is then constructed via simple triangulation. How different DEMs can be combined is topic of discussion in section 4.2.

4 ILP AND STORAGE

4.1 Image Richness Index

All planetary exploration missions need to take into account the limited bandwidth available to send data to Earth. An Aerobot system could traverse many kilometers but the required bandwidth to send all recorded data back to Earth would almost certainly not be available. It is therefore imperative for the Aerobot system to make decisions on the importance of its recorded data. Once calibration and reconstruction have been computed, scoring techniques can be applied to the data which now comprises of not only the images but of 3D information as well. The scoring algorithms should be written such that high scores for an image correspond to a large chance on interesting geological features to be found in that specific image. These scores, called Image Richness Index (IRI) can then be taken into account to decide on an appropriate compression factor, compression scheme and broadcast priority for the images and DEMs.

4.2 DEM Combination and Compression

The DEMs that have been computed as explained in paragraph 3.3 yield the depth between the terrain and the respective camera for every pixel. Because consecutive images have an overlap of about 80-85% (see § 3.1), there exists much duplicated information. In light of the bandwidth constraints this problem must be dealt with. The solution lies in the form of masks that are constructed for all images. Fig. 4 gives a schematic overview of the process. The dense stereo matching of paragraph 3.3 yields a matching pixel in the second image for almost every pixel in the first image of an image pair. For the first image pair all matched pixels in the overlapping area correspond to newly seen 3D points and thus this entire area must be saved. The corresponding mask for the first image is therefore filled where the grey area is visible in the left of fig. 4. For the following image pairs, matters are a bit more complicated as shown on the right of fig. 4. Image 2 and 3 overlap again for a large part but only the grey area is new and the mask is only set for this region. The shaded area of image 2 is not visible in image 3 but has been reconstructed in image 1. The hatched area is visible in all three images. The already computed DEM of image 1 could be updated with this new information, e.g. using an averaging filter. In our setup we have decided against this because it again delays the time when a final DEM is ready for transmission.

It is infeasible to send all recorded and reconstructed data to Earth uncompressed. The masks we have computed can be employed in the compression step. We have chosen to make use of the JPEG2000 standard [11] for compression of our image and DEM data. This standard allows for an arbitrary shaped region of interest (ROI) to be specified. The use of wavelets in the JPEG2000 algorithm allows one to be able to select a certain area of an image to view at a high quality, while leaving the rest of the image at a lower quality. Thus the algorithm for compressing our image and DEM data first computes the IRI for an image (§ 4.1). Based on this value a certain amount of bits are accorded to the data. The JPEG2000 algorithm is then executed and compresses the data with respect to this maximum, taking into account the masks as a ROI. Fig. 5 shows some results. The top left image is the original and the top middle is the mask. On the top right the JPEG2000 compressed image is shown with ROI enabled. The bottom left image is also JPEG2000 compressed with the same bitrate but without ROI. The last two images show a detail of the bottom left part of the compressed image with and without ROI respectively.

Figure 5: The top left image is the original and the top middle is the mask. On the top right the JPEG2000 compressed image is shown with ROI enabled. The bottom left image is also JPEG2000 compressed with the same bitrate but without ROI. The last two images show a detail of the bottom left part of the compressed image with and without ROI respectively.
compressed with the same bitrate but without ROI. The last two images show a detail of the bottom left part of the compressed image with and without ROI respectively. It is clear that the ROI-enabled algorithm tries not to spend bits in areas that are not in the ROI. The detailed images clearly show artifacts that are introduced by the JPEG2000 compression without ROI that are not present in the ROI-enabled algorithm.

5 RESULTS

The computer vision software was tested on simulated (virtual) data and also on 2 kinds of real imagery. The first real imagery was from a video recording. The second came from a demonstrator that consists of a 2m wide spherical helium balloon over the planetary testbed at ESTEC, and lifting a digital camera that can be remotely controlled. A couple of consecutive pictures that were taken by the digital camera of the balloon are shown in fig 6.

Figure 6: Three consecutive pictures taken by the Sony camera on-board the aerobot system

These pictures have the standard size of NTCS frames, i.e. 640x480 pixels. Resulting reconstructions, including 3D and camera poses are shown in figures 7, 8 and 9.

Figure 7

Figure 8

Figure 9

The screenshots show combined 3D models generated from the DEMs of different images in one sequence. The ROI files are used as masks. The 3D reconstructions are created automatically and look very convincing. They show that the system is capable of performing as expected. The strategy of computing and using an ROI for every DEM proved especially useful.

6 REFERENCES


when some intrinsic parameters can vary."


