Experiences on the Implementation of a 3D Reconstruction Pipeline

Severino Paulo Gomes Neto[1], Márcio Augusto Silva Bueno[1], Thiago Souto Maior Cordeiro de Farias[1], João Paulo Silva do Monte Lima[1], Veronica Teichrieb[1], Judith Kelner[1], and Ismael H. F. Santos[2]

[1]Federal University of Pernambuco
Virtual Reality and Multimedia Research Group Computer Science Center

[2]CENPES Petrobrás, Rio de Janeiro, RJ

{spgn, masb, mouse, jpsml, vt, jk}@cin.ufpe.br, ismaelh@petrobras.com.br

***Abstract*—3D reconstruction from 2D images aims recovering models that represent accurately and in three dimensions features of interest of objects in a scene. This is the subject of the so called TechPetro project with the purpose of developing engineering solutions based not only on 3D reconstruction but also on markerless augmented reality. While the first allows the automatic 3D reconstruction of complex scenes captured from the real world, markerless augmented reality augments users' perception through the use of an interface that integrates in real-time 3D virtual information into the real world scene visualized by the user. In this paper experiences made on the definition and implementation of a 3D reconstruction pipeline are presented, together with some preliminary results.**

***Index Terms*—3D Reconstruction, Pipeline, Real-time, Markerless Augmented Reality.**

I. Introduction

Markerless Augmented Reality (MAR) [1] and 3D Reconstruction (3DR) [2][3][4][5] are two evolving research areas that show an impressive increase in sophistication and complexity of their applications. MAR consists in a particular branch of Augmented Reality (AR) [6] in which the need for fiducial marks is dismissed [7] in order to use the surrounding world as a marker. The fundamental intent of AR (taking advantage of users knowledge and familiarity with their own environment to perform tasks) can be actually achieved, after all, fiducial markers are intrusive objects needed only by the application that uses them, but never by the users. Although, the development of MAR techniques to deal with basic AR tasks (tracking and registration) becomes much more complex and demands higher computational power.

3DR is a research field that involves several techniques with the purpose of recovering models that represent accurately and in three dimensions features of interest (shape, structure, texture etc.) of a specific object or a set of them. In this context, there are techniques that interfere in the environment in order to achieve their goals (active techniques) as structured light, infrared and sonar based techniques, and also the ones that do not interfere (passive techniques) as the one called Structure from Motion (SfM) [8][9].

Passive techniques are often more complex, but have more flexibility because do not demand especially prepared or controlled environments. On the other hand, active techniques are not too complex, but depend on certain constraints of the reconstructed environment, object or equipment (object size, closed spaces, projection of light on a room or surface etc.) that harms their applicability in some environments without this level of control. The natural consequence of the differences between passive and active techniques is the need for knowing in detail the problem to be solved before assigning one of them as the best.

Despite of other necessary considerations, it is reasonable to accept that the use of optical sensors has a higher utilization potential, due to their popularity, easiness of handling and reduced cost (in comparison with other types of sensors).

Lately, MAR and 3DR applications became more sophisticated and complex due to the popularization of these technologies, which became from merely academic experimentation to commercial systems with costumers mainly in medical, industrial inspection, training, merchandising, and entertainment segments. The popularity of such systems is also due to another important factor: the increase of computation power. With a set of resources that are more adequate to the demands of applications developed using MAR and 3DR technologies, it became possible to immerse users meanwhile they utilize the systems in a manner very close to the one imagined since the conception of these technologies. Until a recent past, this requirement had not been achieved completely.

Initially, systems that applied MAR and 3DR technologies faced the barrier of resource restriction and were only able to reach less than satisfactory or workable levels. With higher computation power, response time decreases and visual quality increases in a manner that new and more complex demands rise (real-time responses, high accuracy, independency of operator etc.), turning early reputable applications into naive ones.

Rigorously, MAR and 3DR are independent fields that count on their specific techniques for solving the proposed challenges of each area. However, considering the fundamental question of application usage, it means, thinking about how the tasks could be performed and taking into account that the results are as good as the user's handling, the need of choosing techniques able to deal with a simple way of interaction is

clear. Precisely at this point both technologies meet.

MAR, as the traditional AR, uses cameras in order to allow tracking of features of interest. In a very similar manner, the SfM technique in the 3DR field can recover objects information by estimating camera movement, which gives a result similar to AR and MAR tracking. Therefore, it is possible to notice a very interesting intersection point where the achievement of the goals of both distinct areas is feasible accordingly to the amount of constraints imposed on the system.

MAR and 3DR must perform tracking in order to recover cameras' relative pose. Adopting a reference coordinates axes system (camera axes system on the first frame), they identify the displacement of the origin (translation, T) and the rotation imposed on each axis (rotation, R) during time. In this intent, both areas apply Computer Vision (CV) algorithms [3][4][5][10][11].

Once correct pose information (R, T) has been obtained, MAR-based systems are able to work, remaining to handle object occlusion. Determining occluding and occluded objects is important for preserving user's sense of immersion, transmitting the impression that all objects in a scene are real, although some of them can be synthetic. The way of dealing with object occlusion also has an intersection with 3DR; the extraction of depth maps allows either rendering only the non-occluded portion of virtual objects as determining characteristics of the modeled world.

3DR systems perform some other tasks that are not common to MAR, but the imposing or the relaxing of constraints allows moving from one problem domain to the other. In these circumstances, it is technically practicable to satisfy users of both profiles through the conception and implementation of an architecture planned with this purpose.

This paper aims to present considerations related to the steps on the development of this architecture, including techniques studied in order to define the problem domain (Section 2), development methodology (Section 3), and tools that can be used to accomplish this goal. In addition to the issues related to research and planning, this paper presents the current development methodology applied by the authors in comparison to the essential steps of reference pipelines (Section 4), some preliminary results (Section 5), as well as the lessons learned (Section 6) until now. At last, the paper draws some conclusions and discusses future work required to deliver a stable, well documented, extensible and easy to use architecture for developing MAR and 3DR applications.

A. The TechPetro Project

3DR and MAR technologies have been studied in the context of the TechPetro project. TechPetro is a two years project, developed by the Virtual Reality and Multimedia Research Group (GRVM) in association with CENPES Petrobras and FINEP. In this project, engineering solutions will be developed based on two technologies: 3DR from 2D images and MAR. These technologies allow the automatic 3DR of complex scenes captured from the real world, as well as the augmentation of user's perception through the use of an interface that integrates in real-time 3D virtual information into the real world scene visualized by the user.

This paper is related to TechPetro's 3DR studies. In sequence, 3DR techniques are briefly described, and preliminary results on the definition and implementation of a pipeline are presented.

II. 3DR Techniques

The techniques studied in more detail were the ones proposed by Marc Pollefeys [9] and David Nistér [8][12][13][14][15]. Pollefeys' technique presents results in 3DR with great quality and Nistér provides 3DR in real-time. Pollefeys works with projective reconstruction, self-calibration and metric reconstruction without knowing camera intrinsic parameters, while Nistér works with metric reconstruction previously knowing the camera intrinsic parameters. These techniques will be briefly detailed in the following sections.

A. Pollefeys Technique

Pollefeys developed a method to obtain 3D models from an uncalibrated monocular image sequence. This method is very flexible because it does not need any previous knowledge about the scene or the camera to build the 3D models. Therefore, there is neither restriction about scene size nor the use of camera zoom while capturing the image sequence.

Calibration of the camera setup and correspondences between images are required to build a 3D model from an image sequence. When acquiring an image sequence by an uncalibrated video camera both prerequisites are unknown and have to be retrieved only from image data.

Figure 1 presents the pipeline proposed by Pollefeys to create a texturized 3D model from an image sequence. The first step comprises generating a projective reconstruction. The epipolar geometry that relates the images is recovered from the extraction and matching of feature points between the images utilizing a robust tracking algorithm. The projective reconstruction of the feature points generates a model with a few hundred reconstructed points.

The projective reconstruction is not satisfactory for 3D modeling because orthogonality and parallelism are in general not preserved. Therefore, the second step aims a metric reconstruction which is accomplished by imposing some restrictions on the internal camera parameters, such as absence of skew, constant aspect ratio etc. By exploiting these constraints, the projective reconstruction can be upgraded to metric (Euclidean up to scale).

The objective of the third step is to estimate a dense depth map. In this step the system has a calibrated image sequence, i.e., besides the intrinsic parameters, the extrinsic parameters are also known, which are the position and orientation of the camera relative to all viewpoints. Therefore, it can be used algorithms which were developed for calibrated 3D systems like stereo rigs. The correspondence search is then reduced to a matching of image points along each image scan line, which is a much easier approach.

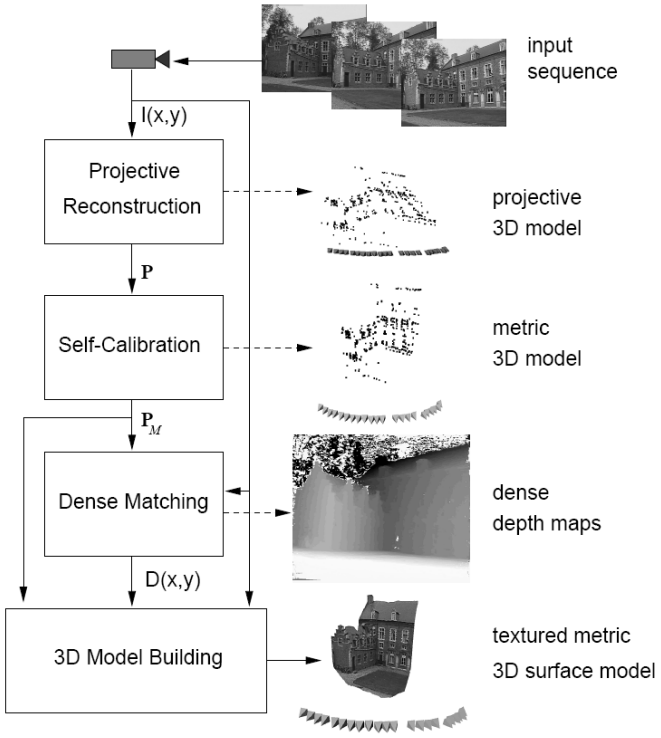

Fig. 1. Pipeline proposed by Pollefeys.

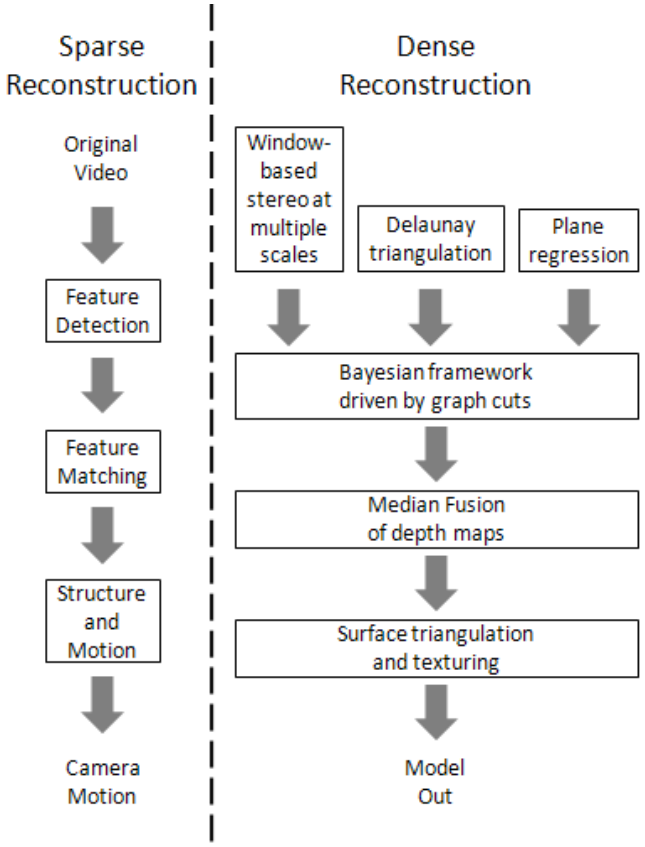

Fig. 2. Pipeline proposed by Nistér.

From these correspondences, the distance of the points to the center of the camera is obtained by triangulation. This result is refined and completed by combining correspondences from multiple images.

In the fourth and final step, the dense 3D model is built through a triangular mesh that approximates the dense depth map. Then, to make it more realistic, textures obtained from the images are mapped to this surface.

In [5], the authors show an approach similar to Pollefeys', which is an interesting reference because it presents a detailed step by step necessary to achieve both projective and metric reconstructions. An introduction to the necessary and involved mathematical concepts and several sample algorithms are presented, which makes that book a good reference for understanding the 3DR area.

B. Nistér Technique

Nistér developed a method for extracting camera pose and a sparse structure of a scene in real-time. To achieve this goal, some restrictions were imposed in comparison to the approach used by Pollefeys, such as the use of calibrated cameras and a maximum time stipulated to estimate the camera pose.

The pipeline proposed by Nistér is presented in Figure 2 (left). The first step of the processing chain is feature extraction from frames using Harris Corners. In sequence, the second step is feature matching between two consecutive frames. The feature tracking continues along the frame sequence.

The next step is recovery of scene structure and camera pose. This is done using the observations, i.e., the feature correspondence between different frames. This step is divided into four phases. The first phase deals with obtaining 2D-2D relative orientation between frames through multiview geometry. At this moment, due to noise and outliers originated from bad matches, it is necessary to use Random Sample Consensus (RANSAC), a robust estimator.

RANSAC generates hypotheses, in this case an essential matrix, from a minimum set of random observations and evaluates these assumptions with respect to all the observations found until a stop condition is satisfied. The problem with this approach is that bad assumptions generated from observations with noise or outliers are thoroughly tested, losing too much processing time. Then, Nistér developed a preemption scheme to prevent unnecessary testing of bad hypotheses and achieve this processing in real-time.

The next phase is recovery of 3D camera pose from the best essential matrix evaluated by the preemptive RANSAC. The third phase is to use triangulation to find the 3D position of the observations. Finally, the fourth and last phase of the recovery of the scene structure and camera pose is the use of bundle adjustment to minimize the reprojection errors.

The final step in recovery of the camera pose is finding its absolute orientation (stitching). At this time, a sparse cloud of points of the scene was also recovered.

According to Nistér [16], from this moment it is possible to use methods of window-based stereo at multiple scales, Delaunay triangulation, plane regression, Bayesian framework driven by graph cuts, median fusion of depth maps, and surface triangulation, and texturing to generate a dense reconstruction,

as can be seen in Figure 2 (right).

III. Methodology

Many tools were applied to develop the 3DR application presented in this work, including prototyping and visualization tools, and CV and numeric computation libraries.

The prototyping environment was the Matlab [17] Release 12. This tool was chosen because it wraps-up several numeric algorithms that are often used by 3DR applications, and made easy the code writing through its seamless math-like language. Matlab was also the choice for visualizing the 3D point cloud generated by the main application.

The developing environment was composed by Visual Studio 2005 running on Windows XP. The code is fully compatible with Unix-like systems, since all libraries are platform independent. Matlab is also distributed in Linux, Mac OS X and Solaris versions.

To build the main application, the CV programming library VW34 [18] was used. This library borrows numeric algorithm implementations from the VXL [19] library (another CV library), and includes specific 3DR methods, such as hypotheses generation and evaluation.

Image acquiring was implemented with the DevIL 1.6.8 [20] library, which deals with several image formats seamlessly. The input for the main application was created synthetically. Two types of input were generated: one with pairs of 2D points, aiming to test the algorithm core, and another with an image sequence, simulating a video that tests also the tracker and introduces noise in the system, which is found in the actual input.

All the inputs were generated with 3D Studio Max 9 [21] . Figure 3 shows samples of frames contained in the generated input.

These tools were chosen taking into account a previous evaluation, where some problems were faced during development using other libraries. These problems will be described in Section 6.

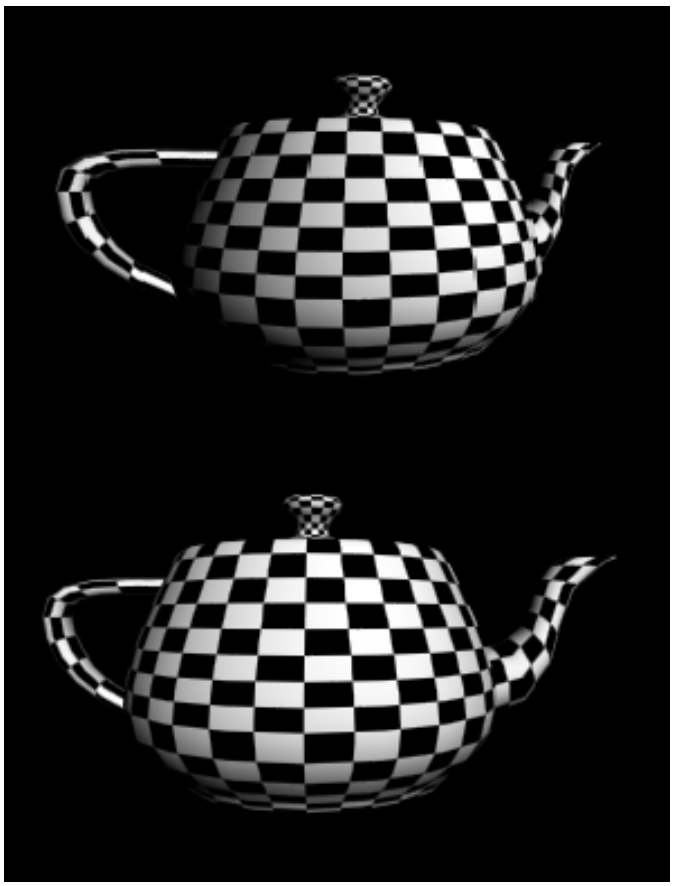

Fig. 3. Two frames from the input sequence representing different poses.

IV. Prototypes Implementation

During this research, some pipelines relative to 3DR were studied. Among the most relevant are Pollefeys' [9] and Nistér's [12] pipelines, which report solutions to 3DR with uncalibrated cameras and real-time 3DR, respectively. These pipelines have common stages, such as feature tracking, hypotheses generation, hypotheses evaluation, hypotheses refinement, pose recovery and triangulation (Figure 4). Each stage is implemented in a different way by the pipelines, since their constraints are different. There are techniques in Pollefeys' pipeline that aren't used by Nistér's pipeline, because there is no real-time implementation available for them. An example is the self-calibration technique, which extracts the intrinsic parameters for camera calibration.

Each stage of the pipelines is implemented with specific techniques. The techniques for Pollefeys' pipeline are:

- FT: Optical Flow (KLT) [22];
- HG: 8-Point [4];
- HE: RANSAC [23];

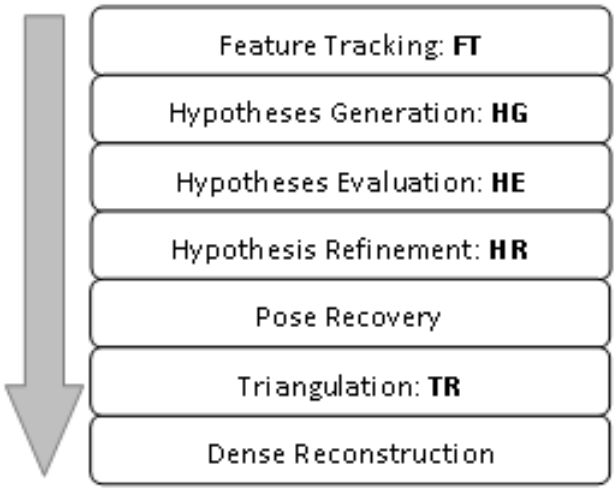

Fig. 4. 3DR pipeline stages.

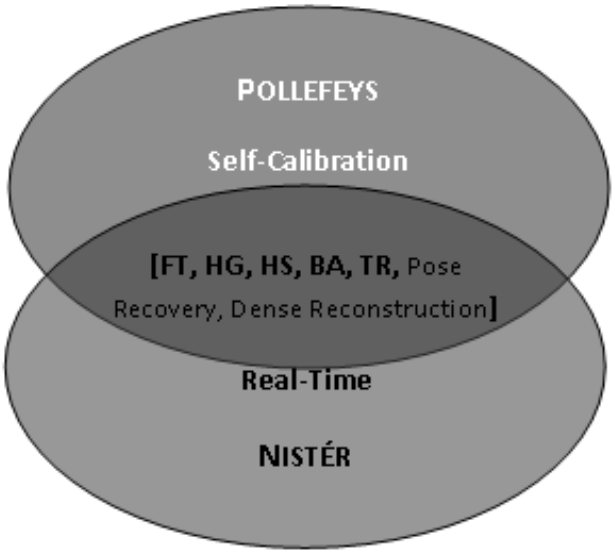

Fig. 5. Intersection between Pollefeys' and Nistér's pipelines.

- HR: Levenberg Marquadt [24];
- TR: Hartley Triangulation [4].

The techniques implemented by Nistér follow different approaches that enforce the real-time constraint:

- FT: Feature Matching;
- HG: 5-Point [14];
- HE: Preemptive RANSAC [13];
- HR: it is not possible to assign which technique Nistér made use, because in Preemptive RANSAC works no citations neither references are given;
- TR: Nistér Triangulation.

Figure 5 shows the intersection between Pollefeys's and Nistér's pipelines stages. The technique set used to implement each stage in each pipeline isn't the same, as mentioned before. There are convenient techniques that give the expected result for each pipeline. These techniques are initially applied in 2-view geometry, and in the next frames, in 3-view and N-view geometry, to get a robust reconstruction.

With the knowledge acquired from these pipelines, some prototypes were built incrementally:

- implementation of the 8-Point algorithm with synthetic 2D input;
- implementation of RANSAC as hypotheses evaluator, on top of the hypotheses generated by the 8-Point;
- replacement of 2D synthetic input by synthetic video, adding a KLT tracker;
- partial implementation of Nistér's pipeline using the VW34 library (no robust estimator was used in the Preemptive RANSAC. The triangulation follows the methods of Hartley [4] and Yi Ma [5]).

In the prototypes mentioned in this work, only 2-view geometry was used.

None of the pipelines is defended by the authors as being the best. Aiming a flexible architecture, the best choices of each pipeline were collected and employed. The pipeline of the main application (without dense reconstruction for exhibition) is:

- FT: Feature Matching (Harris);
- HG: 5-Point/8-Point;
- HS: Preemptive RANSAC;
- TR: Hartley/Yi Ma.

More results are needed to present a final pipeline.

V. Results

During prototype development, onward to the current stage on reconstruction accuracy, some efforts have been done that resulted in a relatively populated cloud of points depicting the reconstruction and allowing recognizing the subject target.

The cloud of points generated by the prototype is stored in files (one for each pair of consecutive frames, since two-view geometry is used), and the 3D points estimation was lately plotted on Matlab for checking out the reconstruction appearance.

In this manner, there are two ways of identifying incorrect reconstructions; the first one by points coordinates in the files, and the second one by observing graphically the plotted points.

Due to the expertise in manipulating points coordinates files it is possible to directly recognize the correctness of the estimation. This artifice is very useful and largely reduces the time spent verifying files in long reconstruction sequences. Table 1 presents a subset of points extracted from one file with good estimation and one with useless estimation.

Table 1. Examples of 3D points estimation extracted from files Pts3D-5 and Pts3D-6; data sequences show respectively x, y and z coordinates.

Pts3D-5 File	71.570420;40.317890;-327.349913; -1.875080; 4.742349; -37.515201; -3.035745; 3.106913; -24.060361; -13.749398;-8.021797; 78.124777; -23.060174;-8.756058; 89.990125; -1.874142;-2.400344; 25.395097; 0.378717;10.940313;-106.158284; -14.877051;16.177142;-160.838883; 103.264636;44.236046;-452.713600;
Pts3D-6 File	0.031040;0.076245;1.496716; -0.047480;0.075126;1.496474; -0.095329;0.075465;1.493502; -0.127540;0.074147;1.486454; 0.010778;0.067653;1.495625; -0.016164;0.067643;1.490876; 0.053710;0.063589;1.503444; 0.034832;0.064559;1.496519; -0.043626;0.066097;1.485419;

On the top of Table 1 (file Pts3D-5) it is possible to notice an almost chaotic variation of z coordinate values, while the bottom part shows z coordinate varying in a short range, meaning a single scale estimation of points. This kind of behavior allows to perform a triage based on reconstruction correctness as well as to compute how many useless reconstructions exist without plotting graphics on Matlab.

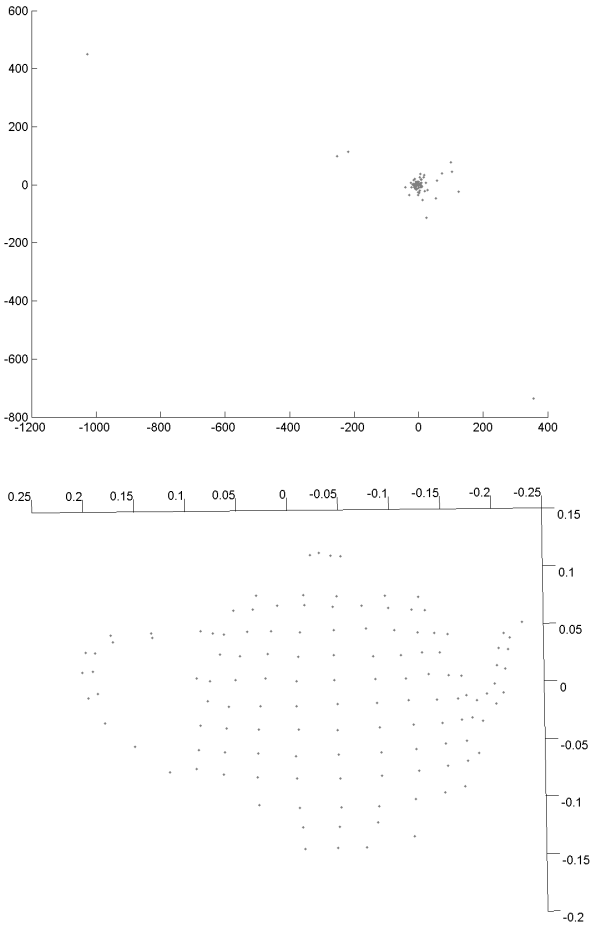

Fig. 6. Reconstruction: incorrect points from Pts3D-5 file (top) and correct points from Pts3D-6 file (bottom).

On the other hand, plotted points give a visual indication about the accuracy of reconstructions in terms of aspect, proportionality, sense of depth and presence of outliers.

Figure 6 shows a reconstruction using 3D points from Pts3D-5 file (top, incorrect) and another using Pts3D-6 file (bottom, correct). It is feasible to recognize the shape of a teapot on the bottom image despite of the top image where the shape is unrecognizable.

Using the prototype at this current stage of development, it is possible to notice a balance between the number of correct and useless reconstructions. Actually, it remains to find a way of analytically rejecting erroneous reconstructions, which implies performing a non-automatic rejection. Despite of this issue, considering only the good reconstructions, it is possible, under certain limits, to increase their quality as explained next.

The question to consider is: recovering 3D points or estimating their true position relative to the reference system implies performing a triangulation using the relative pose estimation in order to discover the depth of each point. In triangulation, the interest lays on recovering points' depth from a pair of images with coordinates given in image plane system. It means that, if a computed essential matrix (E) is correct, it is possible to recover the information about relative rotation (R) and translation (T) between the two frames and, consequently, it becomes practicable to determine the correct representative of each tracked point of this pair.

The representative of a point is a concept of projective geometry in which a point has infinite representatives. Since the first reconstruction is a projective one, it is necessary to determine the correct scalar that selects the accurate representative of each point in order to reconstruct the target object properly. This is the context where triangulation is applied.

Triangulation depends on two issues: correct estimation of relative rotation and translation, and significant translation. Related to relative pose recovery, the cited problem of rejecting analytically useless pose estimates still remains. However, concerning the need for a significant amount of translation, a basic algorithm of frame decimation [8][25] was implemented, allowing an increase in the sense of depth of the reconstructed model.

The algorithm assumes that two consecutive frames under the same conditions of light only have enough luminance disparity in two circumstances: significant translation or change of scene. The second circumstance is not a worry condition since scene changes implies in "tracking failure", what can be interpreted as a very large translation in which any step of the proposed reconstruction system is inapplicable. It is a well known constraint of this type of system and must be respected in order to work fine. Other types of systems are designed for such circumstances.

The first assumption is quite reasonable, since each pixel on an image has a tendency to be surrounded by pixels with similar colors and luminance intensity. In this manner, if a pair of frames has a difference between their mean global luminance under a threshold, the algorithm discards the second frame and takes a new one to evaluate together with the preserved frame from the last evaluation. This process is repeated until a pair of frames has a disparity that satisfies the threshold.

The sequence of images of Figure 7 shows the results of correct reconstructions. The set of points is the same in all images, but using different views. Left images refer to reconstructions without frame decimation, having a low sense of depth. Right images refer to reconstructions using the implemented algorithm of frame decimation in which an increase in the sense of depth could be noticed.

Top left and right images do not look much different due to a practically orthogonal projection. Middle images depict a top (or aerial) view, where the points represent the shape of a hemisphere of a teapot. Bottom images show a rear view of the same points.

Next section details the efforts done and problems faced along the development of the current prototype that generated the results presented here.

VI. Lessons Learned

Solving the SfM problem means, logically, performing each task of a pipeline accordingly. In order to implement those tasks the authors faced some implementation challenges as follows.

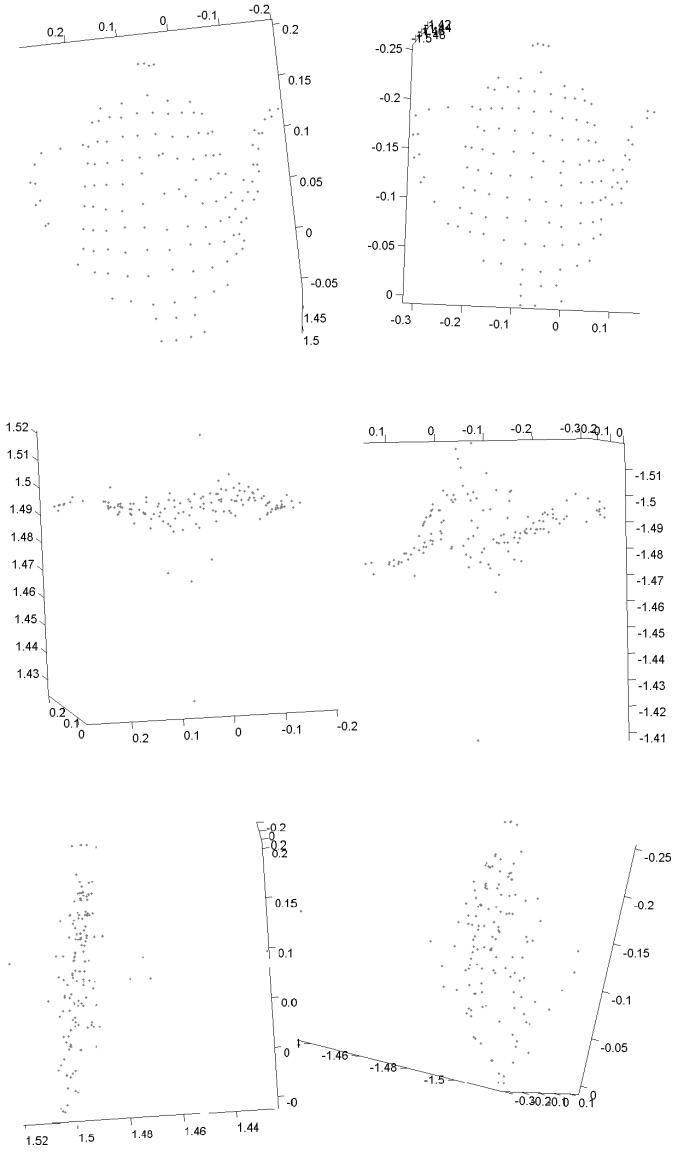

Fig. 7. Teapot points: without frame decimation (left) and after the inclusion of the basic frame decimation algorithm (right).

Taking advantage of prior familiarity with OpenCV [26] and Gandalf [27] libraries, the efforts were concentrated on implementing a module responsible for generating essential matrices (movement hypotheses) [28] through the linear algorithm of 8-Point [28] taking as reference the Matlab implementation.

Naturally, since it was a basic implementation, only a set of synthetic points coordinates were used as input data. At the end of the code translation the obtained results did not correspond to the Matlab results.

When inspecting the code translated to both libraries in order to find some incorrect piece of code, all the code looked correct. The strategy that led to the discovery of errors was to compare each intermediate result obtained when using Matlab, OpenCV and Gandalf implementation. The difference among values was encountered after the operation of Single Value Decomposition (SVD) [5].

Since the use of SVD is very frequent on SfM solutions, those libraries were discarded and the attention went to another one called VXL. This library worked correctly when solving SVD and using the hypotheses generation module. Beyond this, it brought some other advantages, such as a more complete and clear documentation, a larger set of modules and functionalities and a more flexible programming model.

After these positive results and a deeper comprehension of Nistér's works [12][13], essential for realizing real-time SfM, it had been started the design of a conceptual architecture that allows reproducing the ideas of such works. Those works have as main contribution a change in the most popular hypotheses evaluation algorithm in this area, called RANSAC [29].

Standard RANSAC uses the complete set of available observations when evaluating each generated hypothesis. Eventually, hypotheses can be early discarded if having very low scores. On the other hand, they could be elected as winner hypothesis before the end of the process if achieving rapidly the expected confidence. This case characterizes a depth-first evaluation since it explores as much as possible a hypothesis before evaluating the next.

Nistér's modification consists in implementing a hybrid evaluation (partially depth-first, partially breadth-first) with emphasis on breadth evaluation and massive cyclic rejection. The assumption that wrong hypotheses have a tendency to accumulate a low score from the beginning is accepted as true, and at the end of the evaluation of an observation block just a half of the best scored hypotheses are preserved for further evaluations. This process is repeated until the winner hypothesis is selected.

Once the choice of the best hypothesis among all the possible ones is an intractable problem and the concept of RANSAC leads to select a hypothesis with good approximation facing some constraints, the called Preemptive RANSAC could be seen as the addition of another constraint: the real-time constraint.

In order to reproduce Nistér's idea trustily the hypotheses generation method should be changed to the non-linear 5-Point algorithm [14][15][30]. When Preemptive RANSAC was presented to the community, some researchers defended it as a robust method in the general case. However, Segvic shows recently that hypotheses generation is hardly affected by the type of motion sequence imposed [31][32].

At this point a problem arises, because VXL does not implement the 5-Point algorithm. Fortunately, Segvig [31] brings a reference to another CV library called VW34, which is a subset of VXL that, in addition to the 5-Point algorithm implementation, has some other pieces of code needed to develop a SfM-based application. This is the library used for developing the current prototype presented in this work.

The set of observations (a list with coordinates of points' correspondences in a pair of frames) must be presented in camera coordinates to be used as a Preemptive RANSAC input in order to perform the hypotheses evaluation. VW34 library comes with an error on the piece of code responsible for transforming world coordinates into image coordinates (step where the z coordinates are also normalized), inside the scope

of Preemptive RANSAC.

Since in the test scenario the intrinsic camera parameters are known, the problem was solved by changing the scope from world to image coordinates transformation. A new method responsible for performing this step was implemented and works by projecting the points onto the image plane using the intrinsic camera parameters matrix (K) [4]. In this manner, the results of the tests became coherent; after all, the observations were used in the same format, but with correct coordinates.

Some comparative tests between RANSAC and another hypotheses evaluation method, named MLESAC [33], were done and both of them worked properly with pure synthetic input data (without presence of noise). However, when using a set of tracked interest points in pairs of frames the results of both methods present some problems.

It was expected that relaxing or ignoring the real-time constraint and making the process deterministic would cause Preemptive RANSAC and MLESAC to elect the same hypothesis as winner. However, the results did not lead to this.

Currently, the authors efforts are for detecting the causes of this mismatch on the results, but the question is that although they converge, incorrect reconstructions still remain and a way of analytically rejecting any incorrect hypothesis must be developed.

Despite of incorrect reconstructions, basic frame decimation was implemented in order to increase the accuracy of the reconstructed model, as described in the prior section.

Beyond the reported progresses, current efforts are related to making the reconstructions coordinate system uniform; after all, in this stage of development the reconstructions are done pair by pair, implying in a reconstruction that has a different system for each pair. For this reason, it is necessary to perform successive transformations until the system coincides with the reference one, which could be arbitrarily fixed as the canonical basis aligned with the optical axis on the first camera frame.

In despite of conceptually simple, this task has a strong dependency on the increase of pose estimation quality. This is due the fact that incorrect estimations imply in degenerated coordinate systems and using those systems for performing transformations affects all the reconstructions done from the incorrect system.

VII. Conclusion

This paper has described an ongoing work that intends to develop a prototype capable of fulfilling either MAR and 3DR demands according to the number of constraints imposed on the system.

A theoretical basis was given in order to describe the foundations of the development as well as the current results achieved. The main problems faced along the development process were also discussed.

Among the next steps in the development process it is important to note the priority in working in an analytical manner in order to discover the incorrect hypotheses, which is the main task to be performed with the purpose of solving many of the current problems. The next step is implementing a frame stitching algorithm [8][25] which allows a uniform reconstruction without duplicated points. When frame stitching works properly, some efforts will be concentrated on frame decimation refinement.

As soon as reliable results using these modifications are achieved, robust estimation will be tackled through the use of three-view and multiple view geometry [4], using trifocal and multiple view tensors [4][5] and the insertion of robust estimators.

Some other works out of the scope of this project remain, like evaluating the general complexity of the SfM problem in order to find critical points and existent bottlenecks that could be solved through new formulations like closed form solutions or innovative ones. This study could also be the basis to have the necessary confidence to assert that some steps in the pipeline may only have a significant performance increase using brute force solutions like CUDA [34].

Acknowledgment

The authors would like to thank FINEP and CENPES Petrobras for financially supporting this research (process 3662/2006).

References

[1] TEICHRIEB, V. et al. A survey of online monocular markerless augmented reality. *International Journal of Modeling and Simulation for the Petroleum Industry*, v. 1, n. 1, p. 1–7, August 2007.

[2] APOLINáRIO, E. L. *Reconstruo 3D em Ambientes com Luzes Estruturadas*. 2005. Trabalho de Graduao, Centro de Informtica, UFPE.

[3] FAUGERAS, O. *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, MA, USA: MIT Press, 1993. ISBN 0-262-06158-9.

[4] HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. New York, NY, USA: Cambridge University Press, 2003. ISBN 0521540518.

[5] MA, Y. et al. *An Invitation to 3D Vision: From Images to Geometric Models*. [S.l.]: Springer Verlag, 2003.

[6] BIMBER, O.; RASKAR, R. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. Natick, MA, USA: A. K. Peters, Ltd., 2005. ISBN 1568812302.

[7] FIALA, M. Artag, a fiducial marker system using digital techniques. In: *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*. Washington, DC, USA: IEEE Computer Society, 2005. p. 590–596. ISBN 0-7695-2372-2.

[8] NISTéR, D. *Automatic dense reconstruction from uncalibrated video sequences*. Tese (Doutorado) — Royal Institute of Technology KTH, Stockholm, Sweden, March 2001.

[9] POLLEFEYS, M. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. Tese (Doutorado) — ESAT-PSI, K.U.Leuven, 1999.

[10] DAVIES, E. R. *Machine Vision: Theory, Algorithms, Practicalities*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004. ISBN 0122060938.

[11] FISHER, R. et al. *Dictionary of Computer Vision and Image Processing*. [S.l.]: Wiley, 2005.

[12] NISTéR, D. Preemptive ransac for live structure and motion estimation. *Mach. Vision Appl.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 16, n. 5, p. 321–329, 2005. ISSN 0932-8092.

[13] NISTéR, D. Preemptive ransac for live structure and motion estimation. In: *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2003. p. 199. ISBN 0-7695-1950-4.

[14] NISTéR, D. An efficient solution to the five-point relative pose problem. *cvpr*, IEEE Computer Society, Los Alamitos, CA, USA, v. 02, p. 195, 2003. ISSN 1063-6919.

[15] NISTéR, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 26, n. 6, p. 756–777, 2004. ISSN 0162-8828.

[16] NISTéR, D. *Real-Time Motion and Structure Estimation from Moving Cameras*. 2005. Tutorial at CVPR. Available at http://www.vis.uky.edu/ dnister/Tutorials/tutorials.html.

[17] MATHWORKS. *Matlab R12*. 2008. Disponível em: <http://www.mathworks.com/>.
[18] LAB, A. V. *VW34*. 2008. Disponível em: <http://www.doc.ic.ac.uk/ ajd/Scene/download.html>.
[19] VXL 1.9.0. 2008. Disponível em: <http://vxl.sourceforge.net/>.
[20] DEVIL - A full featured cross-platform image library. 2008. Disponível em: <http://openil.sourceforge.net/>.
[21] AUTODESK. *3D Studio Max*. 2008. Disponível em: <http://www.autodesk.com/3dsmax>.
[22] LUCAS, B.; KANADE, T. *An iterative image registration technique with an application to stereo vision*. 1981. Disponível em: <citeseer.comp.nus.edu.sg/180224.html>.
[23] FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, 1981. ISSN 0001-0782.
[24] LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II, n. 2, p. 164–168, 1944.
[25] NISTéR, D. Frame decimation for structure and motion. In: *SMILE '00: Revised Papers from Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*. London, UK: Springer-Verlag, 2001. p. 17–34. ISBN 3-540-41845-8.
[26] INTEL. *OpenCV - Open Source Computer Vision Library*. 2008. Disponível em: <http://www.intel.com/technology/computing/opencv/>.
[27] GANDALF. 2008. Disponível em: <http://gandalf-library.sourceforge.net/>.
[28] LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature*, v. 293, p. 133–135, 1981.
[29] FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 726–740, 1987.
[30] LI, H.; HARTLEY, R. Five-point motion estimation made easy. In: *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006. p. 630–633. ISBN 0-7695-2521-0.
[31] SEGVIC, S.; SCHWEIGHOFER, G.; PINZ, A. Influence of numerical conditioning on the accuracy of relative orientation. In: *CVPR*. IEEE Computer Society, 2007. Disponível em: <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2007.html>.
[32] SEGVIC, S.; SCHWEIGHOFER, G.; PINZ, A. Performance evaluation of the five-point relative pose with emphasis on planar scenes. In: *AAPR/OAGM*. Austria: Schloss Krumbach, 2007. p. 33–40. Disponível em: <http://www.zemris.fer.hr/ ssegvic/pubs/oagm07.pdf>.
[33] TORR, P.; ZISSERMAN, A. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, v. 78, p. 138–156, 2000. Disponível em: <citeseer.ist.psu.edu/torr96mlesac.html>.
[34] NVIDIA. *CUDA*. 2008. Disponível em: <http://developer.nvidia.com/object/cuda.html>.