ISIS technical report series, Vol. –, — 2005

A Review of 3D Reconstruction from Video Sequences

MediaMill3D technical reports series

Dang Trung Kien
Intelligent Sensory Information Systems
Department of Computer Science
University of Amsterdam
The Netherlands

This report gives an overview of 3D reconstruction from video sequences. It is to be distributed with the MediaMill3D reconstruction system as an introduction to the theoretical aspects underneath.

Draft version 0.1

Intelligent Sensory Information Systems
University of Amsterdam
The Netherlands

Contents

Intelligent Sensory Information Systems
Department of Computer Science
University of Amsterdam
Kruislaan 403
1098 SJ Amsterdam
The Netherlands

tel: +31 20 525 7463
fax: +31 20 525 7490
`http://www.science.uva.nl/research/isis`

Corresponding author:
Dang Trung Kien
tel: +31(20)525 7508
`dang@science.uva.nl`
`http://www.science.uva.nl/~dang`

1 Introduction

This report is an overview of 3D reconstruction from video sequences. It is one of the MediaMill3D technical reports series that documents literature (and also engineering issue) of the MediaMill3D system [11].

The system is made to serve as a test bed and a software prototype in the "Crime Scene Investigation using hand-held cameras" project [81]. in summary, its functions are to reconstruct a 3D model of a crime scene from video sequences, add text and voice annotations, and fuse information from different video sequences. Later on it will be embedded into the MediaMill Video Search system [67].

In this document, the literature of 3D reconstruction (first function of MediaMill3D) is reviewed in order to:

- Summarize the literature on 3D reconstruction from video sequences and their evaluation.
- Conclude on the most suitable methods to integrate into MediaMill3D.
- Identify shortcomings and define solutions and additional functionalities

The target of our system is the geometric model of the scenes. So here we consider geometric reconstruction and not photometric (or image-based) reconstruction, which directly generates new views of a scene without (completely) reconstructing the 3D structure.

With the stated purposes stated and application context we set the limits for the report as:

- ***Static scenes***: There is no moving object or the movement of objects is relatively small.
- ***Uncalibrated cameras***: The input data is captured by an uncalibrated camera, i.e. the camera's intrinsic parameters such as focal length is unknown.
- ***Varying intrinsic camera parameters***: The camera intrinsic parameters (e.g. focal length) can vary freely. Together with the previous, this assumption adds flexibility to the system.

2 Overview of 3D Reconstruction From Video Sequences

First we introduce our application-oriented description of *3D reconstruction from video sequences* (shortly called *3D reconstruction* in this document).

1. The process starts with the data capturing step, in which a person moves around and captures a static scene using a hand-held camera.
2. The recorded *video sequence* is then pre-processed (e.g. selecting *frames*), removing noise, normalizing illumination).
3. After that, the video sequence is processed to produce a 3D model of the scene.

4. Finally, the 3D model can be rendered, or exported for editing using 3D modeling tools.

In this report, we especially focus on the third step. The problem of how to capture a scene (first step) is shortly discussed from a theoretical point of view in section 4. The second and fourth step are not the subject of this document since they are quite general tasks. So from this point in the document, the term "3D reconstruction" denotes *only the third step*.

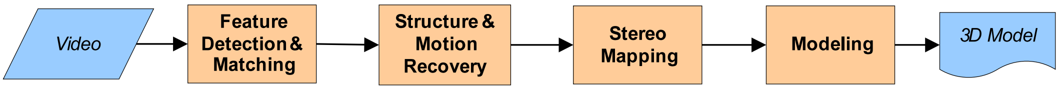


Figure 1: Main tasks of 3D reconstruction

The 3D reconstruction (step 3) can be divided into 4 main tasks (figure 1), which are discussed in the following sections.

1. **Feature detection and matching** (Section 3): The objective of this step is to find out the same features in different images and match them.

2. **Structure and Motion Recovery** (Section 4): This step recovers the structure and motion of the scene (i.e. 3D coordinates of detected features; position, orientation and parameters of the camera at capturing positions).

3. **Stereo Mapping** (Section 5): This step creates a dense matching map. In conjunction with the structure recoved in the previous step, this enables us to build a dense depth map.

4. **Modeling** (Section 6): This step includes procedures needed to make a realistic model of the scene (e.g. building mesh models, mapping textures).

This is the most general view of the 3D reconstruction process. Other diagrams of reconstruction process could be easily mapped to it. For example, in the framework of Pollefeys [52] (figure 2): (A) and (B) are mapped to the input, (C) and (D) are mapped to (1), (E) and (F) are mapped to (2), (G) is mapped to (3), (H) is mapped to (4), (E) is mapped to the output.

Some define the input as an image sequence but in figure 1 we define it as a video sequence since our practical objective is a system that does reconstruction from video. By defining it like that, we want to clearly state that the intermediate step to go from video to image sequences (i.e. frame selection) is a part of the reconstruction process.

3 Feature Detection and Matching

This process creates relations used by the next step, structure and motion recovery, by detecting and matching features in different images. Until now, the features used

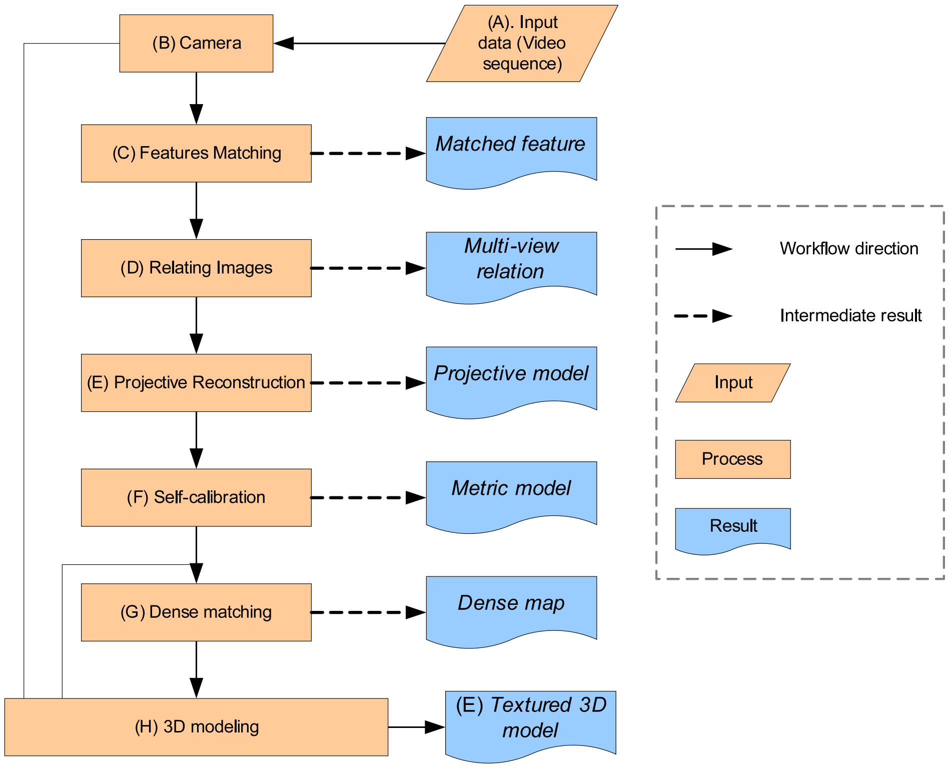


Figure 2: Pollefeys 3D modeling framework [52]

in structure recovery processes are points (e.g. [19, 53]) and lines (e.g. [15]). So here features are understood as points or lines.

Here are some important concepts that are repeatedly used in the coming text:

- ***Detectors.*** Given an image a *feature detector* is s arocess to detect features from the image. The most important information a detector gives is the location of features, but other characteristics such as the scale can also be detected.

  Two characteristics that a good detector needs are *repeatability* and *reliability*. Repeatability means that the same feature can be detected in different images. Reliability means that the detected point should be distinctive enough so that the number of its matching candidates is small.

- ***Descriptors.*** Suppose we have two images (from two different views) of a scene and already have extracted some features of them. To find corresponding pairs of features, we need *feature descriptors*. A descriptor is a process that takes information of features and image to produce descriptive information i.e features' description, which are usually presented in form of features vectors. The descriptions then are used to match a feature to one in another image.

  A descriptor should be *invariant* to rotation, scaling, and affine transformation so the same feature on different images will be characterized by almost the same value and *distinctive* to reduce number of possible matches.

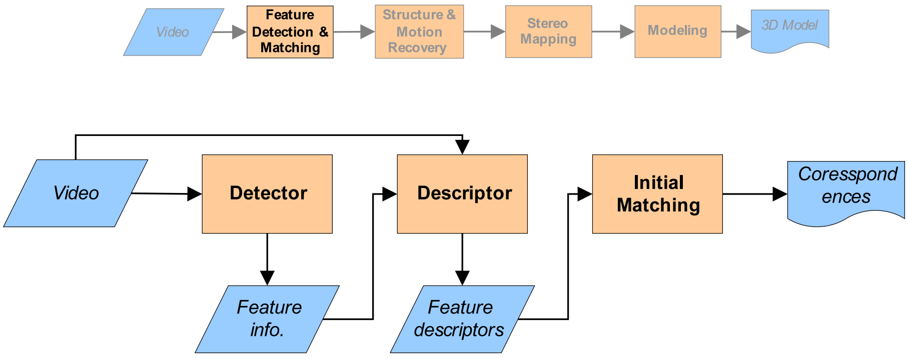


Figure 3: Feature detection and matching process

The roles of detectors and descriptors in the feature detection and matching step are given in figure 3.

The following sub-sections summarize research on interest points and lines using the two concepts detectors and descriptors.

3.1 Interest Points

In this document, a point feature is called an *interest point.* A definition of interest points, given in [62], is “any point in the image for which the signal changes two-dimensionally”. This definition however cannot cover all kinds of interest points. Hence in this report we define it as a point that can be identified using same point detection process.

3.1.1 Point detectors

Classification. In [62] Schmid et al classify point detectors into three categories: contour based, intensity based, and parametric model based ones.

- ***Contour based detectors***. These detectors first extract contours from images and then find points that have special characteristics, e.g. junctions, endings, or curvature maxima. A multi-scale framework can be utilized to get more robust results.

- ***Intensity based detectors***. These detectors find interest points by examining the intensity change around points. To measure the change, first and second derivatives of images are used in many different forms/ combinations.

- ***Parametric model based detectors***. Points are found by matching models/ templates (e.g. of L-corners) to an image.

Related works and evaluation. A repeatable detector is one that is invariant to changes, i.e. rotation, scale, affine transformation, and intensity.

The Harris corner detector [16] is well-known and is invariant to rotation and partially to intensity change. However, it is not scale invariant. Scale invariant detectors such as [41, 33] search for features over scale space. Lowe's SIFT [41] searches for local maxima of Difference of Gaussian (DOG) in space and scale. Mikolajczyk and Schmid [33] use Harris corners to search for features in the spatial domain and then use a Laplacian in scale to select features that are invariant to scale. An affine invariant detector is defined by Tuytelaars and Van Gool [78]. Starting from a local intensity maximum, it searches along rays through that point to find local intensity extrema. The link formed by those extrema defines an interest region[1], which is later approximated by an ellipse. By searching along many rays and using ellipses to represent regions, the detected regions are affinely invariant. Their experiments show that the method can deal with view changes up to 60 degrees.

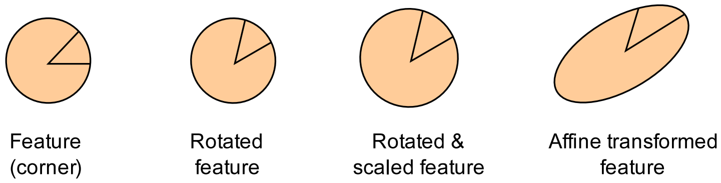


Figure 4: A feature under different transformations

The criterion for repeatability evaluation is the ratio of number of repeated points over the total detected points in the common part of two images. The evaluation in [62] unfortunately does not cover scale and affine invariant detectors. Among the examined ones, the Improved Harris corner, which is improved from the original by employing a more appropriate differential operator, gets the best score.

In [62], a *local jet* (a set of image derivatives) is used to characterize interest points. Reliability of a detector is measured by the diffusion of local jets. The more diffusive the descriptive values, the more distinctive (reliable) the detector. This diffusion is measured using entropy. The Improved Harris also gets the best place.

Conclusion. It is difficult to say which detector should be used in practice since we do not have an exhaustive evaluation. In practice, speed is also a decisive factor. Unfortunately none of available evaluation considers this criteria but intuitively more sophisticated detectors are slower. In the next paragraph we will see that the final matching rate does not depend much on the chosen detector. So an average but fast detector and a good descriptor is probably a wise choice.

3.1.2 Point descriptors

In this sub-section we discuss point descriptors and their application to matching for 3D reconstruction.

[1] Note that via scale, and affine transformation, a point is usually no longer a point but becomes a region. So in literature, for robust detectors we see "interest regions" instead of interest points. When using matching regions for 3D reconstruction, we can simply use the centroids of regions for computation.

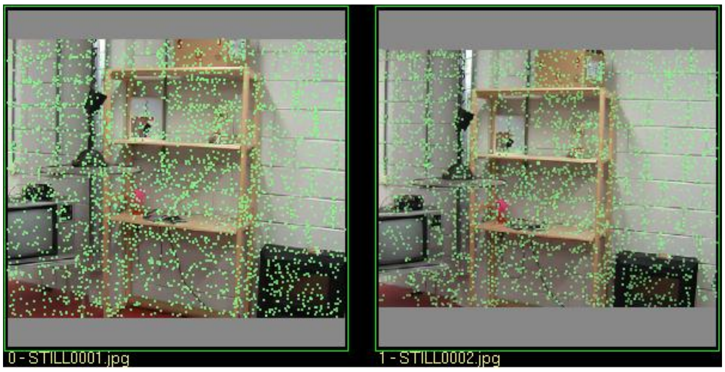


Figure 5: A interest points detected by SIFT (green marks). Pictures produced by MediaMIll3D

Classification. In [45], Mikolajczyk and Schmid classify point descriptors into the following categories:

- ***Distribution based descriptors***. Histograms are used to represent the characteristics of the region. The characteristics could be pixel intensity, distance from the center point [38], relative ordering of intensity [82], or gradient [40].
- ***Spatial-frequency descriptors***. These techniques are used in the domain of texture classification and description. Texture description using Gabor filters is standardized in MPEG7 [57].
- ***Differential descriptors***. The descriptor used in [62] to evaluate detectors' reliability is an example of a differential descriptor, in which a set of local derivatives (local jet) is used to describe an interest region. Others are described in [4, 59].
- ***Moments***. Van Gool et al. [14] use moments to describe a region. The central moment of a region in combination with the moment's order and degree forms the invariant.

Related work and Evaluation. The invariance of descriptors is obtained in many ways for different changing factors. For example, in [40, 33] maxima of local gradients with different directions are used to identify the orientation. Other sets of rotation invariants can be used to characterize the region, e.g. The Fourier-Mellin transformation used in [4]. Scale and skew determined in the detecting phase are used to normalize image patches [4, 59].

Mikolajczyk and Schmid have done an evaluation of descriptors in [28, 45]. ROC (Receiver Operating Characteristic - the ratio of detection rate over false positive rate) and *recall* (the ratio of correct matches over possible matches) criteria are used respectively. The results coincide. The SIFT descriptor [41], which is invariant to scaling and partially to view change, and SIFT-based methods [30, 45] are in the top group. This evaluation also shows that using region-based detectors in particular the scale and affine invariant detectors give slightly better results.

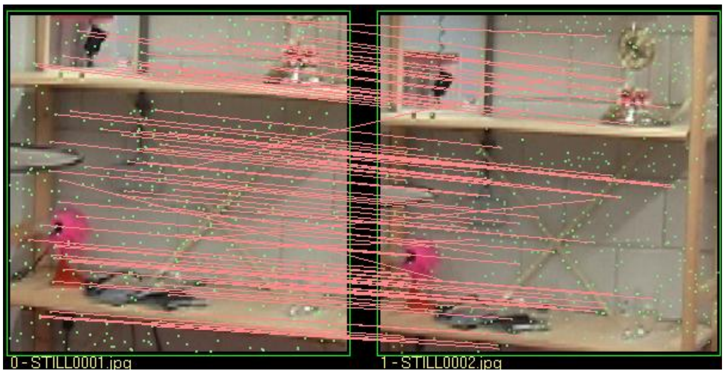


Figure 6: Correspondences initially matched based on SIFT descriptors. Pictures produced by MediaMill3D

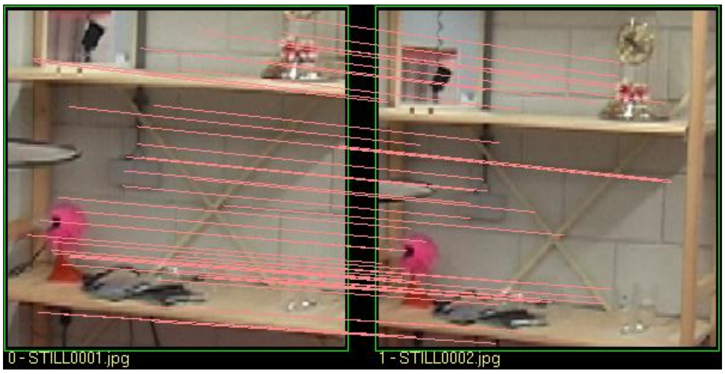


Figure 7: Correspondences after being filled by the fundamental matrix. Pictures produced by MediaMill3D

Conclusion. It is proved the evaluations that the choice of the descriptor is more important than the choice of the detector. However, note that many good descriptors use information produced by complex detectors (e.g. scale). SIFT and SIFT-based descriptors are attractive because of their performance. From this fact, SIFT is used in the the MediaMill3D system.

3.2 Lines

Two-view projective reconstruction can only use point correspondences. But in three or more view structure recovery it is possible to use line correspondences. In this sub-section we discuss line detection and matching for 3D reconstruction.

3.2.1 Line detection

Line detection usually includes edge detection, followed by line extraction.

Edge detection. Many edge detection schemes are available in the literature (Shin et al [65] counted 22 new algorithms proposed in 4 journals from 1992 to 1998). The key to solve the problem is the intensity change, which is shown via the gradient of the image. Edge detectors usually follow the same routine: smoothing, applying edge enhancement filters, applying a threshold, and edge tracing.

Evaluations of edge detectors are inconsistent and not convergent [65, 13] for reasons such as unclear objective and varying parameters. Shin el al have done a series of evaluation in different tasks [65, 66], in which the application acts as the black box to test algorithms. One of them is structure from motion. The evaluation shows that overall, the Canny detector [8] is most suitable because of its performance, fastest speed, and low sensitivity to parameters variation. However, the structure from motion algorithm used there [75] is not a three-view one and uses line segments rather than lines as in [23]. Also the "Intermediate processing" (line extraction and corresponding) that would affect the final result is fixed. Thus the result is not concrete enough.

Line Extraction. Extracting lines could be done in several ways. The Hough transform [27] is famous in curve fitting. Despite of having a long history Hough transform and its extensions are still used widely in recent literature (e.g. [46, 2, 50]). A simpler approach connects line segments with a limit of angle changes and than uses the least median square method to fit the connected paths into lines (e.g. [69, 65]).

As with edge detection, no complete and concrete evaluation of line extraction is available.

3.2.2 Line matching

Lines can be matched based on attributes such as orientation, length, or extent of overlap. Some matching strategies such as nearest line, or additional view verification can be used to increase the speed and accuracy. Optical flow can be employed in the case of short baseline [35]. Matching groups of lines (graph-matching) is more accurate than individual matching [79].

Beardsley et al [5] use the geometric constraints in both two-view and three-view cases to match lines. The constraints are found by a robust method with corresponding points. Schmid and Zisserman extend this idea for matching both lines and curves [63].

The evaluation of line matching algorithms for reconstruction is missing in literature. Authors give evaluations to compare their work with previous works (e.g. [35]) but those are not complete enough to draw a conclusion.

Conclusion. Lines and generally highly structured features give stronger constraints. Lines are many and easy to extract in scenes with dominant artificial objects, e.g. urban architectures. However, the fact that evaluations on line extraction and matching for structure recovery are not complete and concrete probably is the reason why the theory of three-view reconstruction with lines are available for a long time but methods in structure recovery usually use point correspondence. One of the few works that uses line correspondences and trifocal tensors is of Breadsley [5] but lines are not used directly. Still point correspondences are used first to recover geometry information.

In conclusion, we recognize the potential of using line correspondence. However, there should be a concrete evaluation of line extraction and matching before we can use it directly. In MediaMill3D, first we will follow the method of Breadsly et al [5] that uses point correspondence to recover the initial structure before exploiting line correspondences.

3.3 Summary and Conclusion

In this section we summarized the literature of feature detection and matching for the structure recovery problem. Two kinds of features are examined: points and lines.

With points, many detection and matching methods are available in literature with concrete evaluations. Choosing a good matching scheme is more important than the detection scheme. SIFT and SIFT extensions show superior results in the evaluation of Mikolajczyk and Schmid [45]. Based on that result we decide to use SIFT and in the the future its extension for our work.

Line detection and matching schemes and their evaluations are not available at the proper level, especially not in the context of 3D reconstruction. We identify it as possible reason of lacking applications of line correspondence in structure recovery. Still we believe that the benefit of using line correspondence has potential. We conclude that before directly applying line correspondence there should be a concrete evaluation. In the first implementation, our system will use lines as additional features to validate the geometric information and to help the dense matching and modeling tasks.

4 Structure and Motion Recovery

The second task *Structure and motion recovery* recovers the structure of the scene and the motion information of the camera. The motion information is the position, orientation, and intrinsic parameters of the camera at the captured views. The structure information is captured by the 3D coordinates of features.

Given feature correspondences, the geometric constraints among views can be established. The projection matrices that represent the motion information then can be recovered. Finally, 3D coordinates of features, i.e. structure information, can be computed via triangulation (figure 8).

In the following subsections we discuss the problem of structure and motion recovery from multiple views and techniques to benefit from a large amount of data, i.e. reconstruction from video sequences, and the degeneracy problem.

4.1 Multiple View Geometry and Stratification of 3D Geometry

This subsection gives a brief overview of multiple view geometry and the concept of geometric stratification that are required to understand the following subsections.

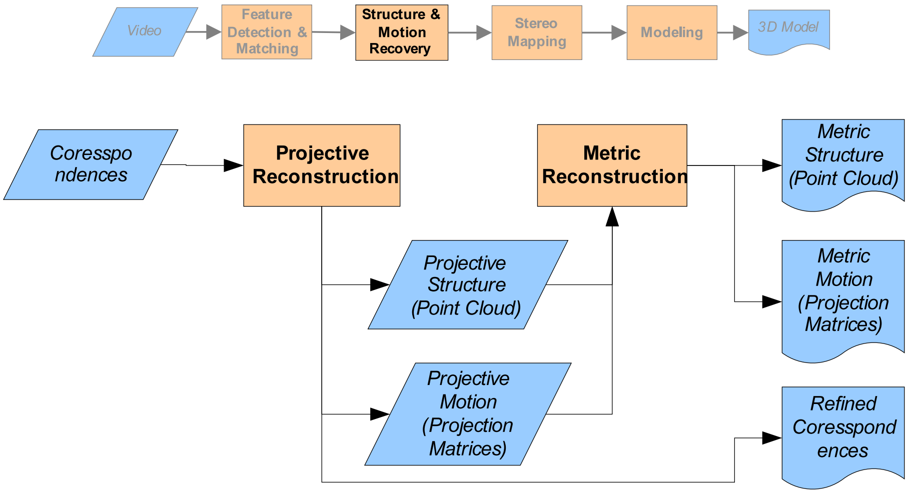

Video
Feature Detection & Matching
Structure & Motion Recovery
Stereo Mapping
Modeling
3D Model
Coresspo ndences
Projective Reconstruction
Metric Reconstruction
Metric Structure (Point Cloud)
Projective Structure (Point Cloud)
Metric Motion (Projection Matrices)
Projective Motion (Projection Matrices)
Refined Coresspond ences

Figure 8: Structure and motion recovery process

4.1.1 Multiple view geometry

The research in 3D reconstruction from multiple views started with two views. This is quite natural since humans also see the world through a two-view system. Initial research assumed calibrated cameras, i.e. intrinsic parameters of a camera and the relative positions of two cameras if a stereo system is employed are known. All those parameters are acquired via a calibration process.

For the calibrated case, the *essential matrix* E [25] is used to represent the constraints between two *normalized views.* Given the *calibration matrix* K (a 3x3 matrix that includes the information of focal length, ratio, and skew of the camera), the view is normalized by transforming all points by the inverse of K: $\hat{x} = K^{-1}x$, in which x is the 2D coordinate of a point in the image. The new calibration matrix of the view is now the identity. Then with a corresponding pair of points (x, x') in homogeneous coordinates, E is defined by a simple equation: $\hat{x}'^{T}E\hat{x} = 0$.

The research has later been extended to the uncalibrated case. During the 1990s, the concept of *fundamental matrix F* was introduced and well studied by Faugeras [47] and Hartley [18]. The F matrix is the generalization of E and the defining equation is very similar: $x'^{T}Fx = 0$.

The difference is that in uncalibrated reconstruction, the K matrix is unknown and thus the view coordinate cannot be normalized therefore in the equation x is used instead of $\hat{x}$). F is still "fundamental" for research of multiple view geometry since it is simple yet very informative. Its relations with other ways of expressing constraints can be found in [17].

Some principle concepts in two-view geometry are explained in figure 9. X, x, and x' are a 3D point and its two projections respectively. C and C' are two camera centers. The line segment that connects them is called the *baseline.* The line X, C, and C' define a planes called the *epipolar plane.* l and l' are the *epipolar lines* of the two projections of X. The projection of the camera centers on the other images, e

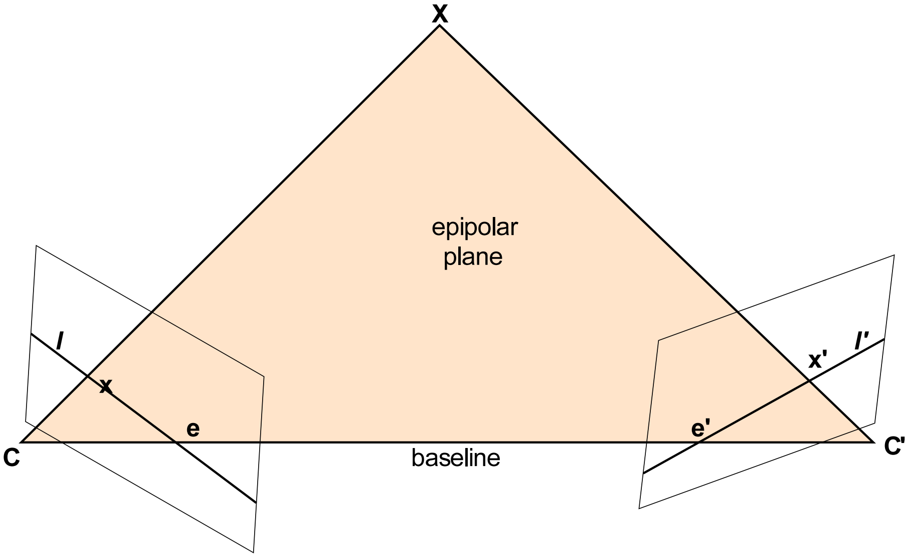


Figure 9: Two-view geometry

and e', are named *epipoles*. The relation among all these elements forms the *epipolar constraint*.

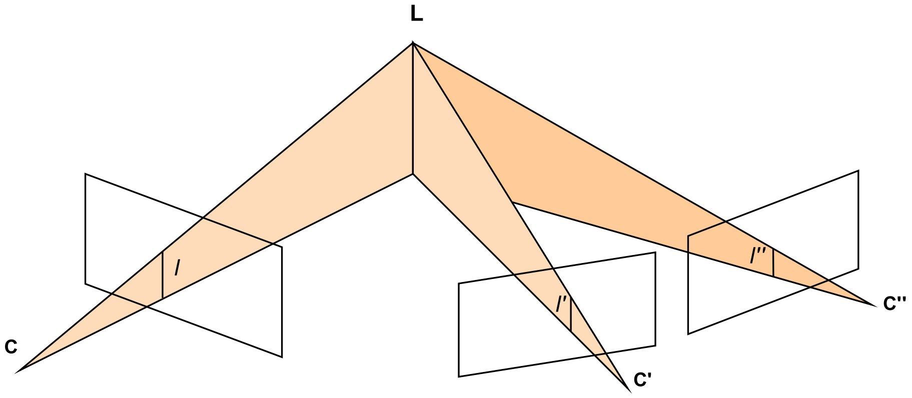


Figure 10: Line correspondence among three view - basis to define trifocal tensors

Three-view geometry is also developed during the 1990s. The geometry constraints are presented by *trifocal tensors* that capture relation among projections of a line on three views. The trifocal tensor defines a richer set of constraints over images. Apart of a line-line-line correspondence, it also defines point-line-line, point-line-point, point-point-line, and point-point-point constraints. Furthermore, it introduces the homography to transfer points between two views. In figure 10, points on lines l are transfered to points on line l'' by the homography induced by the plane (C', l'). Unlike the fundamental matrix, which defines a point to line relation, i.e. a one-to-many relation, line correspondences defined by trifocal tensors are one-to-one. This is one of the advantages of trifocal tensors identified not only by Hartley and Zisserman [17] but also by Faugeras, Luong and Papadopoulo [48].

4.1.2 Stratification of 3D geometry

Geometry stratification. F matrices and trifocal tensors form the constraints among multiple views. Suppose that we have enough information, i.e. correspondence of features, to find the F matrix, or the tensor. It alone is not enough to solve the structure and motion problem. The ambiguity of reconstruction is because of the lack of prior information. In calibrated reconstruction, the information of the camera's parameters are known before hand by a calibration process and are unchanged. In human vision system, our cameras are also well calibrated. But in uncalibrated case to make a Euclidean reconstruction, the missing information must be recovered via a further step called self-calibration, which is discussed in section **??**. This process employs the idea of geometric stratification [12] and invariant objects.

To make a metric reconstruction, i.e. a reconstruction in the metric space (Euclidean space up to a scale), one should find the corresponding reconstruction in lower stratum: the affine and projective space respectively. The affine space is the metric space without the measurement of angles (parallelism, ratio, centroid still exist). In the projective space, the concepts of parallelism, ratio, and centroid do not exist anymore. Tangency still exists in this space.

A calibration process uses calibrated objects, i.e its characteristics are known, to find out the camera's parameters. In the uncalibrated reconstruction, invisible but always existing objects are the replacements. They are the plane at infinity, the absolute conic and the absolute quadric. The plane at infinity and the absolute conic are invariant in affine and metric space respectively. The absolute dual quadric encodes information of the others and is used in Pollefeys' metric reconstruction method [53].

Characteristics of geometric strata are summarized in table 1 and can be found in [52] or [17]. In section 4.3 we will discuss different methods of reconstruction from video sequences in the uncalibrated case.

4.2 Projective Structure and Motion

Reconstruction with only knowledge of feature correspondences is only possible up to a projective reconstruction and there are many ways to obtain projection matrices from a geometry constraint, i.e. a fundamental matrix or a focal tensor. Hence projective reconstruction is mainly the recovery of fundamental matrices or focal tensors. Methods, implementation hints, and evaluations are well discussed by Hartley and Zisserman in [17].

If the input, i.e. feature correspondences, includes outliers, robust methods such as RANSAC, LMS can be employed to reject them. Then iterative minimization using Levenberg-Marquart should be used to improve the result. Choosing error functions for the minimization is very important since the algebraic error, i.e. the estimation error computed directly from the geometric constraint equations, does not express geometric meaning. Geometric or Sampson distance are advised [17].

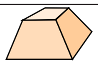
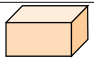
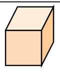
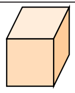
Stratum	DoF	Trans.Matrix	Distortion	Invariants
Projective	15	P		*Intersection, Tangency of surface, Cross-ratio*
Affine	12	$\begin{bmatrix} A & \\ 0^T & 1 \end{bmatrix}$		*Parallelism, Centroid, Plane at infinity*
Metric	7	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$		*Relative distance, Angle, Absolute conic*
Euclidean	6	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		*Absolute distance*

Table 1: Characteristics of geometric strata [52, 17]. *The matrix P is a 4×4 invertible matrix. The matrix A is a 3×4 matrix. The R matrix is 3×3 rotation matrix. s is the scaling factor. t is a 3D translation vector*

With recovered focal tensors, projective reconstruction is already available. There are many decompositions from tensors to projection matrices [42]. The most commonly used one assumes that the first camera projection matrix is $P_1 = [I\ 0]$ and derives the other view's projection matrix based on the constraint.

In case of more then two views, the decomposition into projection matrices must be done with homographies induced from the same reference plane in order to have a consistent structure. This could be based on the work of Luong and Viéville [43] or based on trifocal tensors as described in [3] by Avidan and Shashua.

To avoid complex equations, one could use the updating method of Pollefeys [53]. After having the initial structure, the new projection matrix is computed from a linear equations system, which is formed from correspondences of already reconstructed 3D points and their projections on new frames. Six points are needed for this computation. The problem of accumulation error is solved by jumping matching, i.e. divide a sequence into K-frame blocks and match the starting frame of the blocks.

Using first view coordinates instead of world coordinates simplifies the equations of reconstruction because the projection matrix of the first view is extremely simple ($P = [I\ 0]$). However, it makes the computation unstable, and sensitive to noise [76]. That is the motivation for the factorization method, first introduced by Tomasi and Kanade for orthogonal projection, which produces a consistent set of projection matrices directly from the correspondences. The method is later extended by Sturm and Triggs [71] for perspective projection. Further developments to solve the problem of initialization, missing trajectories and continuous reconstruction are given in [1, 44]. An evaluation method is given is [74]. Theoretically, factorization gives better results compared to the structure update technique. Yet there is no explicit experimental verification of this. Whether factorization is better (more accurate and effective) than structure update with a good frame selection is still a question.

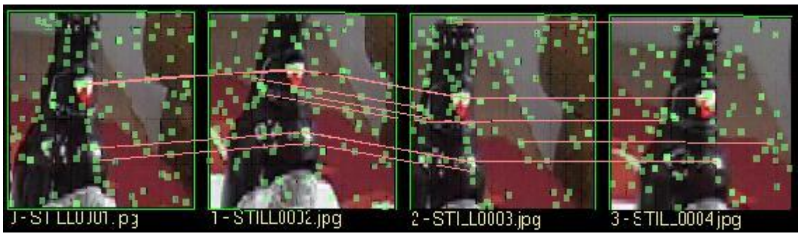


Figure 11: Tracking features over frames. Same features detected over frames are required to initiate and build up the projective structure. Pictures produced by MediaMill3D

Conclusion. Both methods, structure update and factorization, should be evaluated further. While factorization has been the research direction in recent years, the updating method has the practical advantage that it is successfully implemented by Pollefeys in a complete framework. In MediaMill3D, we implement the former, the later will be examined easily since this step is modularized. We also will examine the effect of frame selection techniques on the final result of both methods.

4.3 Metric Structure and Motion

The process of upgrading from projective structure to a metric one is called *self-calibration* or *auto-calibration.*

Without additional constraints, it is impossible to upgrade to a metric reconstruction [55]. The development of research on self-calibration goes from methods with strict unrealistic assumptions of camera motion and intrinsic parameters [37, 20] to the flexible, practical ones with minimal and realistic assumptions (e.g. self-calibration even with only the condition of squared pixels) as in [24, 55, 56].

Available methods. The iterative methods of Pollefeys [55] and Heyden [24] directly recover a metric structure from projective one. However, Pollefeys builds up the method from an analysis of the absolute quadric equation (remember that the absolute quadric encodes characteristics of both the affine and metric strata), whereas Heyden derives the solution from the projection matrix equation.

Hartley, on the other hand, from the comment that iteration is tricky [56] and a direct upgrade method has the difficulty of constraint enforcement [17], proposed a full stratified method. The method [56] first upgrades to a affine structure by an exhaustive search for the plane at infinity with branch cut based on the so called cheirality constraint, i.e. the reconstructed points must be in front of the camera [21]. Later the affine structure is upgraded to a metric one by finding the image of the absolute conic as in [37].

Evaluation. Heyden [24] gives experimental results on synthetic data. Pollefeys and Hartley both give results on real images. However, the experiments were not complete enough to convince the robustness of the methods. Also there is no comparison among them. It is not clear from the papers how to detect and deal with (nearly) critical sequences. In Pollefeys' method, convergence and semi-definite constraint are not enforced explicitly during the process. The application context of those methods is historical and architectural objects, in which geometric accuracy is not highly demanded and the distortion almost does not exist on images (in fact Cornelis et al proposed a method to deal with distortion in the uncalibrated case but for some cases the method is unstable and unreliable [9]).

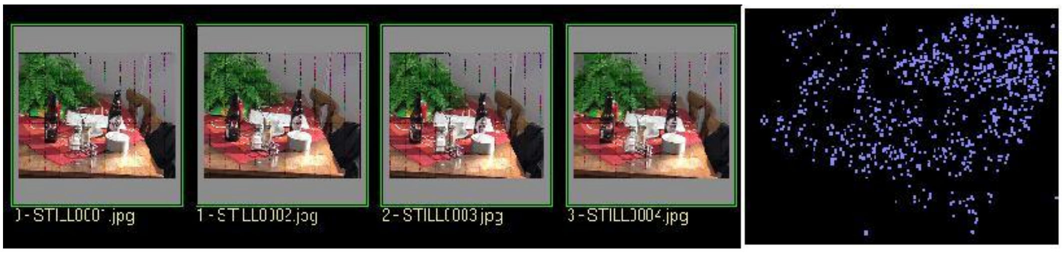


Figure 12: A sequence of a part of the NFI's crime scene and the metric structure recovered. Pictures from MediaMill3D system

Conclusion. In summary, methods exist for metric reconstruction but the robustness, accuracy, and flexibility can still be improved. To apply those methods in our application, we need a thorough examination of them with respect to our application context, i.e. the dimension of the scene and the required accuracy.

4.4 Advantages and Problems of Using Video Sequences

Research in 3D reconstruction started with looking for an answer to the question whether it is possible to do 3D reconstruction from images. But in practice, it is more natural to use video sequences since it eases the capturing process and provides more complete data. But also problems arise. The following text discusses the advantages and the problems of using video sequences as input.

4.4.1 Advantages

The most important advantage of using input of video sequences is the higher quality one can obtain. Both geometric accuracy and visual quality can be improved by exploiting the redundancy of data. Intuitively, more back-projecting rays of a point's projections limits the possible 3D coordinates of the point. The best texture found by selecting the best view or super-resolution can be used to get better visualization quality. Image sequences also enable some techniques to deal with shadow, shading, and highlights such as described in [53].

Other advantages are the automaticity and flexibility. Capturing data by a hand-held camera is more comfortable since a person does not have to worry about missing information or to consider if the captured information is enough for reconstruction. And on the processing time, instead of manually selecting some images from a video, it is better to have a system that can do everything automatically.

4.4.2 Problems

To take advantage of the use of video sequences we have to deal with some problems, ranging from pre-processing (frame selection, sequence segmentation), during processing as has been seen in previous sub-sections, to post-processing (bundle adjustment, structure fusion).

- **Frame selection.** Among a number of frames, selecting good frames will improve the reconstruction result. Good frames are ones that have proper geometric attributes and good photometric quality. The problem is related to the estimation of views' position and orientation and photometric quality evaluation.

- **Sequence segmentation.** Reconstruction algorithms assume that a sequence is continuously captured. The sequence should be broken into proper scene parts and reconstruct separately and fuse later.

- **Structure fusion.** Results of processing different video segments (generated either by different captures or by segmentation) must be fused together to create a final unique result.

- **Bundle adjustment.** The reconstruction process includes local updates (e.g. feature matching, structure update) and bias assumptions (e.g. use of first-view coordinate system). Those lead to inconsistency and accumulated errors in the global result. There should be global optimization step to produce a unique consistent result.

For all mentioned problem, there exists solutions to certain levels. Yet no solution is absolutely perfect.

4.5 Critical Cases

A critical case happens when it is impossible to make a metric reconstruction from the input data. It either because of the characteristics of the scene or the capturing positions.

In practice, metric reconstruction from video sequences captured by a person using a hand-held camera hardly falls into an absolute critical case. However, nearly critical cases are common in practice, e.g. a camera moving along a wall or on an elliptic orbit around the object. That is why studying critical configurations and detecting those cases is extremely important to create a robust reconstruction method, or select the most suitable method for the case.

There are two kinds of *critical cases*: (i) *critical surface* or *critical configuration*, and (ii) *critical motion sequences* (of camera). The first class depends on the observed points and the viewpoints. The study of this kind started very early (1940s) and still is subject of recent research, e.g. by Kahl el al in 2001 [29]. The later depends only on camera motion, i.e. can happen with any scene. Sturm's paper [70] on fixed intrinsic parameter cases provides the basis for further research. He also suggested a "brute force" approach to select the best algorithm. Pollefeys gives a practical approach that examines the condition number of the equation system [51]. This however only helps to reject the case but not to take the proper method for it.

Some important notes about critical cases are:

- Normal cases in some conditions, e.g. calibrated or fixed intrinsic parameters, can turn into critical ones when conditions change.
- The more uncalibrated the camera, i.e. less cameras' parameters are known, the more ambiguous the reconstruction will be [51].

In conclusion, research of absolute critical cases is concrete but a method to early detect near cases is not available. It will be an important contribution if one could develop such a method since it will pave the way to build an effective metric reconstruction method.

4.6 Summary and Conclusion

There are many options for each step in the reconstruction process. Some steps are well evaluated while others need further evaluation.

Both accuracy and robustness should be improved in order to make image-based 3D reconstruction more applicable. Using more data can improve the quality. But also many problems come along when using video sequences.

To build our system, which aims for a quality-demanding application, we have chosen to implement the practically best algorithms for each step and will have to do research and experiments to find out how to improve the quality of the system, either by improving the algorithms or conforming to a good capturing guideline.

5 Rectification and Stereo Mapping

The structure created after the second phase is very discrete and not enough for visualization. Also a dense depth map must be established in order to build the 3D model. This task can be divided into two sub tasks: *rectification* and dense *stereo mapping*. The first one exploits the epipolar constraint to prepare the data for the second one by aligning a corresponding pair of epipolar lines along the same scan line of images thus all corresponding points will have the same y-coordinate in two images. This makes the second task, roughly search and match over the whole image, faster. Furthermore, other optimization techniques are used in the second task to achieve a correct dense matching before triangulation to build the depth map. Figure 13 illustrates the process.

5.1 Rectification

The first class of rectification methods is *planar rectification*, e.g. [22]. Both images are projected onto a plane that is parallel to the baseline. This class of rectification works fine with traditional application of rectification in which there is no forward movement of the camera, e.g. aerial images. It fails in the case of a forward moving camera, which is quite common when a video sequence is captured by a hand-held

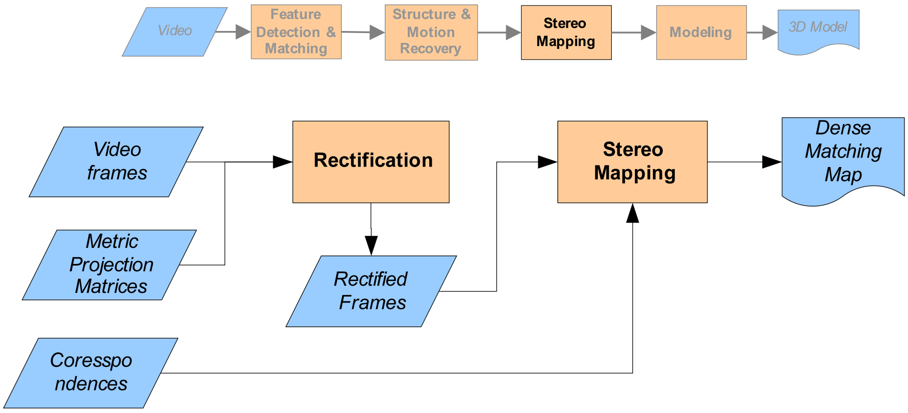


Figure 13: Stereo mapping process

camera, e.g. moving along a street or corridor. In this case, planar rectification will create an unbound image.

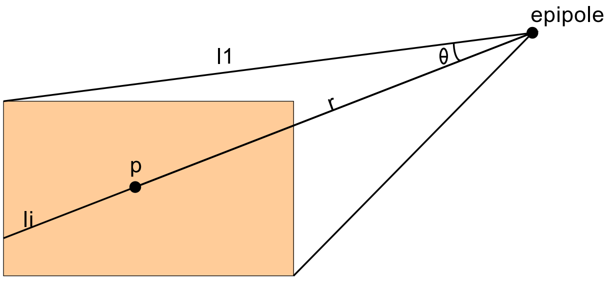


Figure 14: Polar rectification. A point p is encoded by a pair of (r, θ)

The second class of rectification methods is *non-planar rectification*. The first invented method in this class is cylinder rectification proposed by Roy et al [58]. Images are projected on a cylinder whose axis is the baseline. The unbound images' size problem is solved. However, the cost is the complexity, which is not a desired characteristic of just a preparation step. Later on, Pollefeys proposed a method called polar rectification [54] that solves the problem while keeping the simplicity. Each pixel is coded by two components, the scan line that it lies on, which is an epipolar line, and its distance to the epipole. The method does not require projection of pixels but only scanning and recoding. A later work by Oram [49] refines this method to reduce feature distortion and complete the solution for the epipole at infinity case.

In the MediaMill3D system, since we do the reconstruction from video sequences captured by hand-held cameras, the polar rectification is implemented.

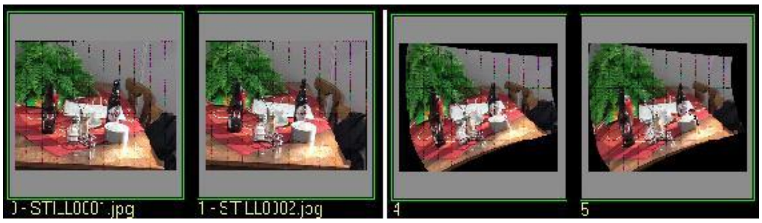


Figure 15: Two consecutive frames before and after rectifying. Pictures produced by MediaMIll3D

5.2 Stereo Mapping

Stereo mapping is the task of establishing a dense matching map between points of different calibrated views. This task is not trivial. The number of papers, different constraints, and strategies applied that can be found in [60] proves that. In the following paragraphs we skim through the taxonomy, and evaluation of stereo mapping based on D. Scharstein and R. Szeliski's work in [60].

5.2.1 Taxonomy

The traditional definition of this task considers only two rectified views and the matching map is presented as the *disparity map* $d(x, y)$ with respect to the *reference image*. The task normally include fours subtasks:

1. **Matching cost computation.** Differences of any pair of pixels from two different images are computed using a cost function, e.g. square intensity difference or absolute difference. The computation range, i.e. the pixels to be compared to a pixel of the reference image, is limited based on a geometric constraint, e.g. the epipolar constraint.

2. **Cost aggregation.** This step increases the accuracy since the cost after aggregation contains information over a region instead of only one pixel. Aggregation in many cases is a local smoothness enforcement.

3. **Disparity computation/optimization.** The decision on matches are made in this step. It can be a simple winner-take-all iteration through all pixels or as complex as a global optimization over all pixels.

4. **Disparity refinement.** Refinements include sub-pixel disparity estimation, e.g. by curve fitting, and post-processing such as applying a median filter to clean up mismatches.

Algorithms can either be *local* or *global*. Local or window-based algorithms decide each match based on matching cost within a limited window region while global ones apply smoothness assumption over whole images and solve the optimization problem. Step (3) of local algorithms is simply a winner-take-all. While in global methods step (3) is the most important one where global smooth constraint is enforced.

Global methods are classified into three sub-classes depending on how subtask (3) is implemented.

- **Global optimization.** The smoothness constraint enforcement is interpreted as an energy-minimization problem. The energy function encodes both intensity and smoothness error. The smoothness function must also preserve the discontinuity at edges. A match is then decided by finding the minimum. A variety of algorithms are applied including Markov Random Fields, simulated annealing, graph-cut, and belief-propagation.
- **Dynamic programming.** Methods in this class solve the minimization along scanlines. Its advantage is the high speed. Three problems of this class are how to define the cost for occluded pixels and in line smoothing constraint, and how to achieve inter-scanline consistency. There are proposed techniques to overcome the first two (e.g. [31]). The last one remains a main problem.
- **Cooperative algorithms.** Cooperative algorithms perform local optimization iteratively. This finally creates a result that is similar to a global optimization.

Scharstein also lists out algorithms that fall out of the taxonomy, for instance those that use optical flow in a hierarchical framework or use multi-valued representation for disparity maps [6, 7].

5.2.2 Evaluation

The most complete and updated evaluation is the one of [60][2]. About forty algorithms are evaluated based on the bad pixel percentage at non-occlusion regions, textureless regions, and discontinuous regions. Currently, the overall best method is *symmetric belief propagation with occlusion handling* [72].

5.2.3 Multiple view mapping

3D reconstruction systems do not use a few images but sequences of them. The ability of simply handling problems of two-view vision, e.g. occlusion or reflection, is what we can gain from multi-view vision. That ability is already acknowledged in some research. Some recent stereo methods [34, 80] use multi-view information. But due to an assumption of occlusion they use only information from close views.

Far views' information is helpful in detecting occlusion by a large object. The *space carving theory* [36] shows an interesting fact of reconstruction from sequences: the more views added, the more complete the model. It proves that the occlusion problem can be addressed effectively with multiple-view information. The framework of [53] updates the depth map and simply detects outliers for example caused by reflection using the linking maps, which represent the correspondences over a sequence.

[2] The evaluation is frequently updated and available on the Internet at http://cat.middlebury.edu/stereo/.

So in short, a practical system, such as the one we are building, should attack the problem on the basis of using video sequences.

5.3 Summary and Conclusion

The number and diversity of methods for stereo matching indicates that it is an important part of 3D reconstruction, especially to create a good model. The problem until now is usually examined from the two-frame and epipolar constraint view. The question is whether it is efficient to apply a complex algorithm in an early step while the problem could be solved by other possibly cheap means in a later stage. With the trend of using video sequences for metric reconstruction, stereo mapping should be done over multiple views. The *correspondence linking* technique of Pollefeys [52], for example, establishes the trajectories of points over views to effectively identify reflection, occlusion and wrong matches. Also it is a fact that thick occlusion cannot be detected from two views but from multiple views it is possible.

In MediaMill3D, we will combine a global two-view stereo algorithm (e.g. [72]) with multi-view information presented by linking maps. This approach would combine the ability of enforcing smoothness of two-view approaches and the ability of detecting occlusion of multi-view approaches.

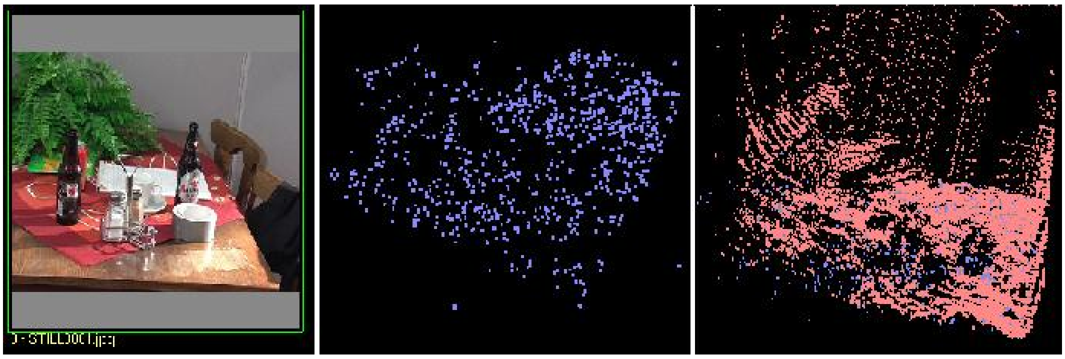


Figure 16: Sparse structure (blue), and dense structure (red) of the same scene. Pictures produced by MediaMIll3D

6 Modeling

The final step is to map texture on the model. The sub-tasks in the modeling step are summarized in figure 17. Triangulation is quite a simple task. Points of each stereo map are triangulated to generate depth maps. Those maps are used to construct the mesh of the scene and finally, with texture extracted from frames, the complete textured model can be built. We ignore it and discuss only mesh building and the texturing process in the following text. Since the triangulation process is simple we ignore it here. Triangulation methods and comparison can be found in [17].

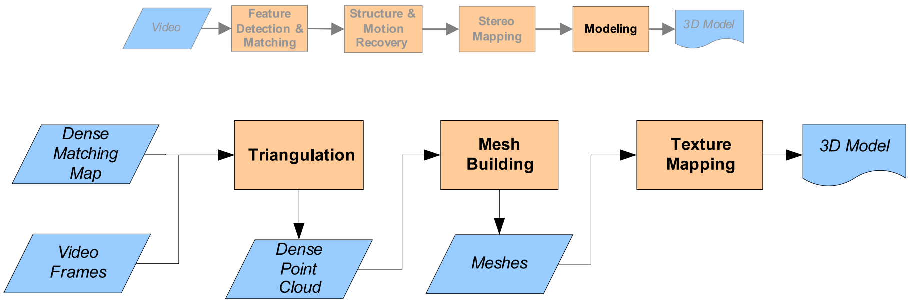


Figure 17: Modeling process

6.1 Mesh Building

The simplest method to build a mesh is first overlaying a triangle mesh on a frame and then using the depth map to compute 3D coordinates of vertices. Each 2D triangle image patch is mapped to the corresponding 3D triangle. This approach however can be applied only to a limited set of close frames since it is impossible to accurately track all points over many frames. Hence its main disadvantage is that it cannot be applied in complex scenes that needs information from different view angles for full reconstruction.

More sophisticate approaches should deal with fusion of meshes constructed from different depth maps. Research on mesh fusion was mostly to support ranging devices, e.g. laser scanners, and usually includes registration before the actual fusion. In our case, we do not have to do registration since all the depth maps are reconstructed based on the same metric frame.

Methods of mesh fusion, or multiple view integration, can be categorized along two dimension: (i) structured or unstructured points; (ii) surface based or volumetric based.

Unstructured fusion (e.g. [73, 10]) does not make the assumption of spatial connectivity, which could be counted as an advantage. But due to this, the methods do not perform well with curved surfaces. *Structured fusion* (e.g. [77]) uses the structure information, e.g. spatial connectivity. This class of method usually performs better than the former.

Surface-based approaches (e.g. [73]) remove or merge mesh elements, i.e. triangles, that overlap each other. The overlap check process usually requires back projection to a 2D plane. *Volumetric approaches* (e.g. [10]) divides the space into voxels and updates their labels based on information from depth maps. After that an isosurface extraction will remove voxels that are before or behind the surface. Surface-based methods' results are meshes so the process can go directly to mesh reduction (e.g using [32, 64]) while volumetric methods require an intermediate polygonization step (e.g. using marching cube [39] or marching triangle [26]).

6.2 Texture Mapping

The texture mapping can be done as follows:

1. Back project the 3D mesh, a set of wire-frame patches, into each frame.

2. Extract texture patches (photometric information) of each wire-frame patches over different video frames.

3. Use photometric and geometric information, i.e. the angles between the line of sight of views and the normal of the wire-frame patch), to create the mapping texture patch.

4. Map the texture patches to the corresponding wire-frame patches.

The process described above already includes the idea of super resolution done in a priory strategy. In the posterior strategy first one super frame is created from the sequence, e.g. by using the idea of *space-time super-resolution* [61], then step (1), (2) with only the super frame and (4) are applied respectively.

Step 3, the super resolution step, can discard artifacts such as highlights and reflections. It can also increase or simply take the best resolution patch for mapping.

In case the scene has close light sources, they should be detected and information of those sources should be taken into account in step 3. In fact, the close light source problem is even more serious since it affects the reconstruction right from the beginning. Light source estimation requires known shapes or surfaces [68] while the surface available at this stage already suffers from the lack of light source information. We have not seen a proper method to solve this problem.

6.3 Discussion

3D reconstruction [3] suffers from the assumption of Lambertian surfaces, i.e. transparent and reflective objects are usually incorrectly reconstructed. The modeling step is the right place to handle this problem since the redundancy of data can help to detect those objects. The refection and transparency problem can be solved to some level as mentioned above. This is an advantage of using cameras over laser scanners.

Modeling is hard to evaluate. Until now the evaluation considers visual assessment, a costly process, or actually no metric evaluation at all. The lack of evaluation is probably due to the fact that most of the modeling methods are created to work with range capturing devices that generate densely accurate data. Maps generated by reconstruction with video sequences are however sparser, due to the limit of resolution and noise filling, and also less accurate. Hence evaluation in this case is more important.

Until now existed 3D reconstruction systems is only at scene level, i.e. one mesh for the whole scene. As a result, the main application is just navigation. A further step into object level is very useful since it enables interaction with the scene.

[3] Either with laser scanners or cameras

7 Conclusion and Discussion

In this document we have given an overview of reconstruction from video sequences over the four main steps: feature extraction and matching, structure and motion recovery, stereo mapping, and modeling. Each step or even sub-step is already a field of research so we did not go into detail. Only an overview of methods and evaluation was given. We also identified problems.

Feature detection and matching is the most well studied step. Methods are plentiful and well evaluated. The correspondences are however never 100% correct. This results in the employment of robust estimation and bundle adjustment methods in the following steps.

We recognize the potential benefit of using video sequences as input in the structure and motion recovery step but also problems arise with that. As working with sequences was called "black art" [17], the only way to conquer it is to study it in practice.

The stereo mapping and modeling task seem less difficult but this might be because modeling with point clouds generated from stereo vision was not examined critically enough.

In general, methods exist for every step. The quality and robustness of the process at large, and especially of the structure and motion recovery step, is the main concern. Although we described the process as an almost sequential one, which is a desired characteristic since it enables a pipeline framework, practical solutions often require loops and feedbacks at different levels. This, on the other hand, makes the quality analysis and management of the process difficult.

References

[1] G. Sparr A. Heyden, R. Berthilsson. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 1999.

[2] N. Aggarwal and W.C. Karl. Line detection in images through regularized hough transform. *IEEE International Conference on Image Processing*, 3, 2000.

[3] S. Avidan and A. Shashua. Threading fundamental matrices. *European Conference on Computer Vision*, 1998.

[4] A. Baumberg. Reliable feature matching acrosswidely separated views. *Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[5] P.A. Beardsley, P.H.S. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. *4th European Conference on Computer Vision*, 2, 1996.

[6] J. R. Bergen, P.Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. *European Conference on Computer Vision*, pages 237–252, 1992.

[7] J. S. De Bonet and P.Viola. Poxels: Probabilistic voxelized volume reconstruction. *International Conference on of Computer Vision*, pages 418–425, 1999.

[8] J. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6), 1986.

[9] K. Conelis, M. Pollefeys, and L.V. Gool. Lens distortion recovery for accurate sequential structure and motion recovery. *European Conference on Computer Vision*, 2002.

[10] B. Curless and M. Levoy. A volumetric method for building complex model from range images. *Proceedings of SIGGRAPH*, 1996.

[11] T.K. Dang. *Technical report: MediaMill3D – System Overview*, 2005.

[12] O. Faugeras. Stratification of 3-d vision: projective, affine, and metric representations. *Journal of the Optical Society of America*, 12(3), 1994.

[13] L.A. Forbes and B.A. Draper. Inconsistencies in edge detector evaluation. *Computer Vision and Pattern Recognition*, 2, 2000. no tensor, 2 views.

[14] L.V. Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. *European Conference on Computer Vision*, 1996.

[15] R.I. Harley. A linear method for reconstruction from lines and points. *5th International Conference on Computer Vision*, page 882, 1995.

[16] C. Harris and M. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference*, 1988.

[17] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision – 2nd edition.* Cambridge University Press, 2004.

[18] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. *Lecture Notes In Computer Science*, 588, 1992.

[19] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6), 1997.

[20] R.I. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1), 1997.

[21] R.I. Hartley. Cheirality. *International Journal of Computer Vision*, 26(1), 1998.

[22] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2), 1999.

[23] R.I. Hartley and F. Schaffalitzky. Reconstruction from projections using grassmann tensors. *8th European Conference on Computer Vision*, 2004.

[24] A. Heyden and K. Åström. Minimal conditions on intrinsic parameters for euclidean reconstruction. *Asian Conference on Computer Vision*, 1998.

[25] H.C. Longuet Higgins. A computer algorithm for reconstructing a scene from two projection. *Nature*, 1981.

[26] A. Hilton, J. Illingworth A.J. Stoddart, and T. Windeatt. Marching triangle: Range image fusion for complex object modeling. *Intl. Conf. On Image Processing*, 1996.

[27] P.V.C. Hough. Machine analysis of bubble chamber pictures. *International Conference on High Energy Accelerators and Instrumentation*, 1959.

[28] C. Schmid K. Mikolajczyk. A performance evaluation of local descriptors. *Conference on Computer Vision and Pattern Recognition*, 2, 2003.

[29] F. Kahl, R. Hartley, and K. Åström. Critical configurations for n-view projective reconstruction. *Computer Vision and Pattern Recognition*, 2, 2001.

[30] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. *Proc. Conference on Computer Vision and Pattern Recognition*, 1, 2004.

[31] C. Kim, K.M. Lee, B.T. Choi, and S.U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. *Computer Vision and Pattern Recognition*, 2005.

[32] R. Klein, G. Liebich, and W. Straer. Mesh reduction with error control. *Proceedings of the 7th conference on Visualization*, 1996.

[33] K.Mikolajczyk and C.Schmid. Indexing based on scale invariant interest points. *International Conference on Computer Vision*, 1, 2001.

[34] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *European Conference on Computer Vision*, 2002.

[35] Y. Kunii and H. Chikatsu. Efficient line matching by image sequential analysis for urban area modeling. *International Society for Photogrammetry and Remote Sensing*, page 221, 2004.

[36] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Conference on Computer Vision*, 1999.

[37] R.I. Hartley L. de Agapito and E. Hayman. Linear calibration of rotating and zooming camera. *IEEE conference on Computer Vision and Pattern Recognition*, 1999.

[38] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. *International Conference on Computer Vision & Pattern Recognition*, 2, 2003.

[39] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface reconstruction algorithm. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 21, 1987.

[40] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[41] D.G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, 2, 1999.

[42] Q.T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Int. Journal of Computer Vision*, 17-1, 1996.

[43] Q.T. Luong and T. Vieville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2), 1996.

[44] S. Mahamud, M. Hebert, Y. Omori, , and J. Ponce. Provably-convergent iterative methods for projective structure from motion. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[45] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. Submitted to PAMI, 2004.

[46] K. Murakami and T. Naruse. High speed line detection by hough transform in local area. *IEEE International Conference on Pattern Recognition*, 3, 2000.

[47] S. Maybank O.D. Faugeras, Q. Luong. Camera self-calibration: Theory and experiment. *European Conference on Computer Vision*, 1992.

[48] O.Faugeras, Q.T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images : The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications.* MIT Press, 2001.

[49] D. Oram. Rectification for any epipolar geometry. *British Machine Vision Conference*, 2001.

[50] R.L. Pires, P. de Smet, and I. Bruyland. Line extraction with the use of an automatic gradient threshold technique and the hough transform. *IEEE International Conference on Image Processing*, 3, 2000.

[51] M. Pollefeys. Self-calibration and metric 3d reconstruction from uncalibrated image sequences. phd thesis, 1999.

[52] M. Pollefeys. Tutorial on 3d modeling from images, 2000. http://www.esat.kuleuven.ac.be/~pollefey/tutorial/.

[53] M. Pollefeys. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59:207–232, 2004.

[54] M. Pollefeys, R. Koch, and L.V. Gool. A simple and efficient rectification method for general motion. *International Conference on Computer Vision*, 1999.

[55] M. Pollefeys, R.Koch, and L.V. Gool. Selfcalibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. *IEEE International Conference on Computer Vision*, 1998.

[56] L. de Agapito R. Hartley, E. Hayman and I. Reid. Camera calibration and the search for infinity. *International Conference on Computer Vision*, 1999.

[57] Y.M. Ro, M. Kim, H.K. Kang, B.S. Manjunath, and J. Kim. Mpeg-7 homogeneous texture descriptor. *ETRI Journal*, 32(2):41–51, Jun 2001.

[58] S. Roy, J. Meunier, and I. J. Cox. Cylindrical rectification to minimize epipolar distortion. *Conference on Computer Vision and Pattern Recognition*, 1997.

[59] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets. *7th European Conference on Computer Vision*, 2002.

[60] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.

[61] E. Schechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2005.

[62] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[63] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.

[64] E. Shaffer and M. Garland. Mesh reduction with error control. *Proceedings of Conference on Visualization*, 2001.

[65] M.C. Shin, D. Goldgof, and K.W. Bowyer. An objective comparison methodology of edge detection algorithms using a structure from motion task. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.

[66] M.C. Shin, D. Goldgof, and K.W. Bowyer. An objective comparison methodology of edge detection algorithms using a structure from motion task. *Computer Vision and Pattern Recognition*, 1, 1999.

[67] C.G.M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, and F.J. Seinstra. The mediamill TRECVID 2004 semantic video search engine. *Proceedings of the 2th TRECVID Workshop*, 2004.

[68] J. Stauder. Point light source estimation from two images and its limits. *International Journal of Computer Vision*, 2000.

[69] C. Steger. An unbiased detector of curvilinear structure. *IEEE Pattern Analysis and Machine Intelligence*, 20(2), year = 1998,).

[70] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. *Conference on Computer Vision and Pattern Recognition*, 1997.

[71] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. *4th European Conference on Computer Vision*, 1996.

[72] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. *International Conference on Computer Vision and Pattern Recognition*, 2005.

[73] Y. Sun, C. Dumont, and M.A. Abidi. Mesh-based integration of range and color images. *SPIE's 14th International Symposium on Aerospace/Defense Sensing,Simulation, and Controls*, 2000.

[74] Z. Sun, V. Ramesh, , and A.M. Tekalp. Error characterization of the factorization method. *Computer Vision and Image Understanding*, 82, 2001.

[75] C. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. *Pattern Analysis and Machine Intelligence*, 17, 1995. no tensor, 2 views.

[76] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 1992.

[77] Y. Turk and M. Levoy. Zippered polygon meshes from range images. *Computer Graphics Proceedings SIGGRAPH*, 1994.

[78] T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. *British Machine Vision Conference*, 2000.

[79] V. Venkateswar and R. Chellappa R. Hierarchical stereo and motion correspondence using feature groupings. *International Journal of Computer Vision*, 15(3), 1995.

[80] Y. Wei and L. Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[81] M. Worring. *Proposal: Crime Scene Investigation using hand-held cameras*, 2004.

[82] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. *Third European Conference on Computer Vision*, 1994.

ISIS reports

This report is in the series of ISIS technical reports. The series editor is Rein van den Boomgaard (reports-isis@science.uva.nl). Within this series the following titles are available:

The file with technical reports “ISISreport–.bbl” is missing

You may order copies of the ISIS technical reports from the corresponding author or the series editor. Most of the reports can also be found on the web pages of the ISIS group (http://www.science.uva.nl/research/isis).

Intelligent Sensory Information Systems
University of Amsterdam
The Netherlands