Kragujevac J. Math. 28 (2005) 19–40.

A SOFTWARE SYSTEM FOR CREATING 3D COMPUTER MODELS FROM PHOTOGRAPHIC IMAGES

Ignace Van de Woestyne and Theo Moons

Katholieke Universiteit Brussel, Faculteit ETEW,
Stormstraat 2, B-1000 Brussel, Belgium

Abstract. Computer vision becomes increasingly important in several scientific, social and economical domains. Not seldom decisions in these areas are based amongst others on accurate computer models in general and 3D computer models in particular. The area of 3D reconstruction is the subarea of computer vision which involves the creation of accurate 3D models from all sorts of input, like photographic images, laser scans, altimetry,...

In this paper we will present a novel, interactive and user-friendly software environment, called ReconLab, together with its mathematical background. The system is amongst others capable of creating a 3D model of an object from images taken from that object.

1. INTRODUCTION

Computer models of existing 3-dimensional (3D) environments (also called *virtual reality* models) play an increasing role in the decision making process at all levels of society. Examples include the use of 3D city models for urban planning: e.g. to demonstrate the influence of a planned building on the surrounding townscape, or to provide the basis for simulation studies such as to predict the impact of noise to

the surrounding buildings while planning new traffic routes or to determine the optimal positions for the implant of antennas for mobile telephony. Another important application domain is surgery planning in medicine where operations involving risk of life are first tried out on a 3D model of the patient which is constructed by integrating volumetric data obtained by CT, MR, PET, ... scanners. But the occasional computer user certainly encountered virtual worlds when playing computer games or visiting a museum or city center via the internet. And, last but not least, nowadays capacity of combining real people and real environments with computer generated objects in film footage offers unlimited possibilities for all sorts of visual effects in movie productions.

In section 2 we will give some basic definitions and concepts and formulate what we mean by the reconstruction problem. Then, in section 3, the reconstruction problem is tackled from a mathematical point of view. We will discuss the distinct steps that can lead to the solution of the problem. In section 4 we shift focus from the mathematical point of view towards the practice of building a software environment for doing 3D reconstruction. More specifically, we will describe the capabilities (and limitations) of ReconLab, its features, point out some future developments and also give some examples.

2. PRELIMINARIES

2.1. DEFINITIONS AND CONCEPTS

A *3D object* is defined as a solid object in a 3-dimensional Euclidean space. A *3D model* is a surface in a 3-dimensional Euclidean space and a *visualization* of a 3D object/model is a graphical representation by means of a computer on a computer screen (which is considered to be a plane). Clearly visualization involves *discretization (sampling)* of the 3D object/model, *projection* onto the screen and *computer graphics techniques* such as triangulation, hidden surface removal,... in order to obtain a "nice" representation. In figure 1 one can see different visualisations of a 3D object.

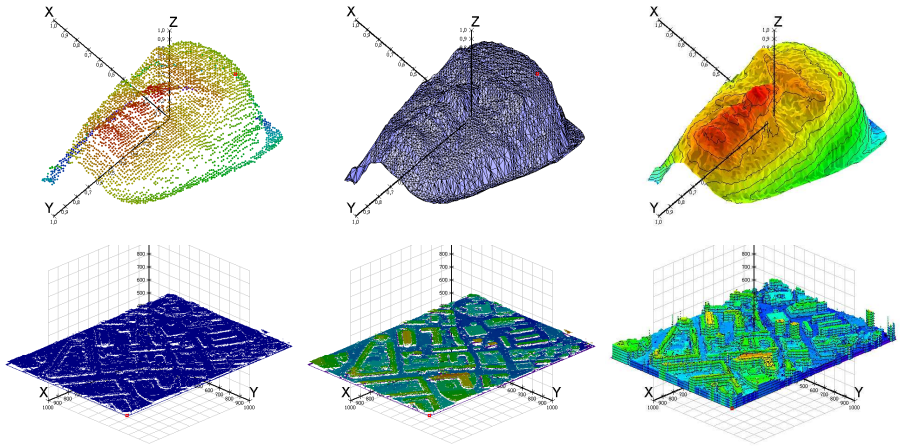

Figure 1. The images on top are different visualizations of the same 3D object (a part of a statue of Joseph Carner, a wellknown poet from Catalonia in Spain). Similarly, the images at the bottom are visualizations of a dense urban area in the city of Amiens in France. The objects are visualized as point model or as solid model. In the latter, a triangulation between the distinct 3D-points is performed. The 3D model can be colored using some fixed color or according to the altitude. Level curves can be included to improve visibility.

By *3D reconstruction* we mean the construction of a 3D model from "flat" information (photographical images for example) of a 3D object. In order to create this model we will have to deal with finding corresponding *image features* (points for instance) and determine the *geometrical relations* based on projection, since we assume that the photographic images were taken with what is called a *pinhole camera*. This type of camera is discussed in more detail in the following subsection.

2.2. A (DIGITAL) PINHOLE CAMERA

A commonly used (digital) camera can be modeled as a so-called *pinhole camera* or *camera obscura*[1]. In this type of camera the image of a 3D object in the environment (this is the *scene*) is formed by the rays of light that are reflected by the 3D object and

[1]Although this camera model gives an over-simplified representation of the image formation process in a real camera, it should be observed that the better lens systems which are commercially available today closely approximate a perspective projection. For our purposes it is therefore not necessary to use a more sophisticated camera model.

fall through the center of the lens onto the *image plane*. (see Figure 2). The image plane may be the surface of a light-sensitive film (as in the case of an analog photo camera for instance) or of a CCD (as in the case of a digital photo camera). Since the projection takes place behind the camera center, the physical image is actually a photo-negative image of the scene. Instead of working with this image, we can imagine an equivalent photo-positive image situated in front of the camera at the same distance of the center of the lens as the image plane (see also Figure 2). In what follows, the term *image plane* will always refer to this hypothetical plane in front of the camera.

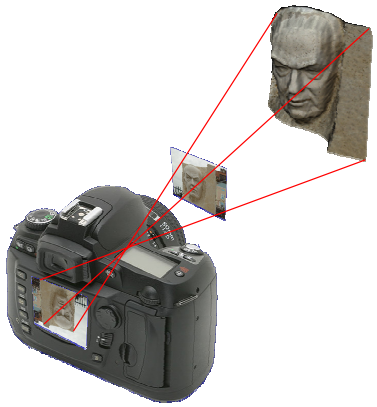

Figure 2. In a *pinhole camera* the image of a 3D object in the scene is formed by the rays of light that are reflected by the 3D object and fall through the center of the lens onto the image plane. The photo-negative image is situated behind the camera center in the plane determined by the light-sensitive film or the CCD. The photo-positive image is imagined in front of the camera at equal distance from the camera center as is the photo-negative image.

Furthermore, the photo-positive image obtained by a digital camera consists of colored *pixels*, which is short for *picture elements*. The number of pixels is determined by the resolution of the CCD used in the camera and also possibly by the software the camera uses for creating the image file. See Figure 3 for a close-up of pixels in a digital image.

Color is obtained by measuring for each pixel the amount of red, green and blue light that reaches this pixel. Usually one byte (this is an integer number from 0 to

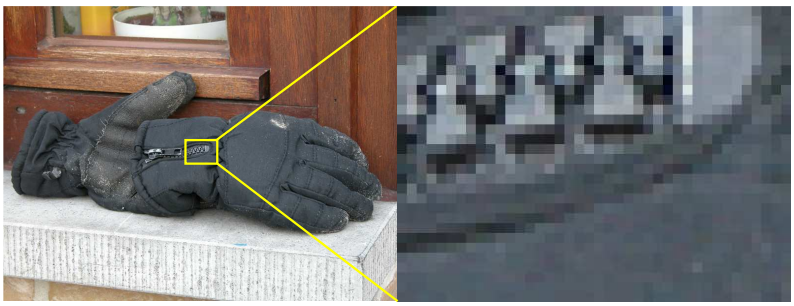

Figure 3. Left is a photo-positive image taken with a digital camera. On the right one sees a close-up of a part of this photographic image. One clearly notices the distinct pixels.

255) is used per pixel for each color band. In Figure 4 one sees the image of a flower, together with its *grey scale image* and the three color bands as it is observed by a digital camera. The grey scale image is obtained by averaging the three color band images. Notice the difference in intensities in the distinct color bands. Although color information can be useful, it increases the complexity of the computations and therefore it is usually not used for reconstruction purposes. Instead, the grey scale image is used for computations.

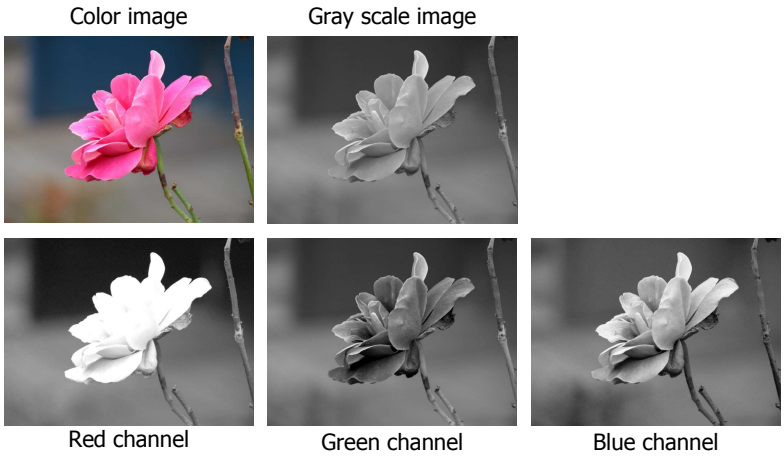

Figure 4. Here is the color image of a flower (a rose), together with its grey scale image and the three color band images.

2.3. THE 3D RECONSTRUCTION PROBLEM

The 3D reconstruction problem can now be formulated in the following way. How can we create a 3D model from a 3D object based on photographic images taken from it from different points of view? We remark that it is not known where the camera was located and how it was oriented at the moment of the recording nor that the camera settings (like zoom, aperture, lens type, ...) were known at that time. In other words, the 3D model must be constructed only from the information available in the photographic images.

3. TOWARDS A MATHEMATICAL SOLUTION

In this section we examine the 3D reconstruction problem from a mathematical point of view.

3.1. REFERENCE FRAMES AND PROJECTION EQUATIONS

In mathematical terms, a 3D object can be seen as a collection of points in Euclidean 3-space $\mathbb{R}^3$. The photographic image of a 3D object in the scene is the perspective projection of the object onto the image plane. By a *camera-centered reference frame* we mean a right-handed, orthonormal reference frame which is attached to the camera with the following properties. The origin coincides with the projection center of the lens of the camera, the Z-axis is the optical axis of the lens and the XY-plane is the plane through the center of the lens, perpendicular to the optical axis. Furthermore, the unity is chosen such that the image plane has equation $Z = 1$. See also Figure 5 (left).

The camera-centered reference frame induces an orthonormal reference frame in the image, as depicted in Figure 5 (right). This reference frame is called the *induced geometrical reference frame*. The image of a scene point P is the point of intersection p_u of the line through P and the origin of the camera-centered reference frame and

the plane with equation $Z = 1$. If P has coordinates $(X, Y, Z) \in \mathbb{R}^3$ with respect to the camera-centered reference frame and p_u has induced coordinates (u, v), then

$$\rho \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix},$$

(with ρ constant) which we shall denote as

$$\rho p_u = P. \tag{1}$$

Equation (1) is called the *projection equation* w.r.t. a camera-centered reference frame.

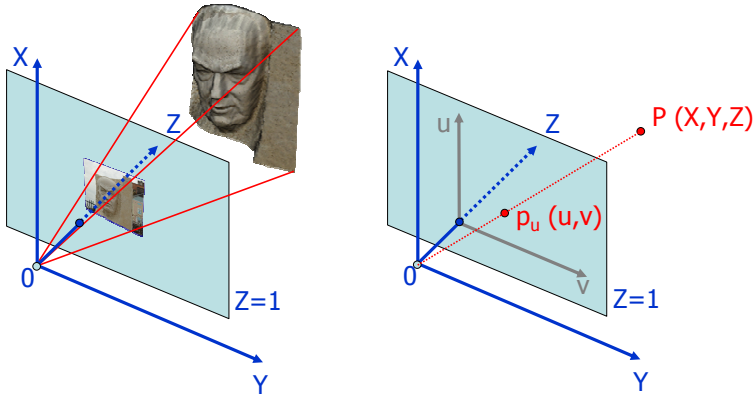

Figure 5. Left: The camera-centered reference frame. Right: The (u, v)-coordinate frame in the image induced by the camera-centered reference frame.

In general, the position and orientation of a camera can also be described w.r.t. some fixed reference frame, called the *world reference frame*. In Figure 6 this general situation is depicted. Clearly, the camera can be transformed to a camera-centered reference frame by first translating the camera such that its projection center coincides with the origin of the world reference frame and then by applying the rotation that aligns the axes. So, if the projection center of the camera has coordinates C w.r.t. the world reference frame and the rotation matrix of the camera is given by R, then

$$\rho p_u = R^t(P - C). \tag{2}$$

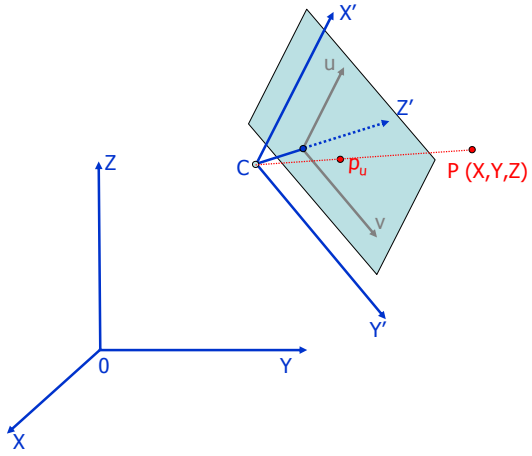

Figure 6. The general case in which the camera is positioned w.r.t. the world reference frame.

Equation (2) is called the *projection equation* w.r.t. the world reference frame.

When working with digital images, it is more natural to indicate the position of an image point in so-called *pixel coordinates* because of the pixels by which it is composed of. Usually, pixel coordinates are measured from the top-left corner and pixels possibly can have a skewed, non-rectangular shape. In general, the transition from geometrical (u, v)-coordinates to pixel coordinates, which we shall denote as (x, y), is modeled by a transformation of the form

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k & s & x_0 \\ 0 & l & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{3}$$

Here, (x_0, y_0) are the pixel coordinates of the origin of the uv-reference frame, called the *optical center* or the *principal point* of the image and k and l give the number of pixels per unit length in the horizontal and vertical direction respectively and thus implicitly describe the length and the width of a pixel. Furthermore, s is called the *skew* and measures how strong the shape of the pixels deviates from being rectangular ($s = 0$ corresponds to rectangular pixels).

We can abbreviate equation (3) to

$$p = K p_u. \tag{4}$$

The matrix K is called the *calibration matrix* of the camera and holds the *internal camera parameters* k, l, s, x_0 and y_0. Combined with equation (2), we obtain the following general projection equation

$$\rho p = K R^t (P - C). \tag{5}$$

The position C of the camera and its rotation matrix R are called the *external camera parameters*.

3.2. RECONSTRUCTION FROM IMAGES

The key issue is to illustrate how the geometric structure of a (static) scene can be recovered from a collection of images of it. Clearly, if there is only one image of an object, it is impossible to reconstruct the object, even if all internal and external camera parameters are known. However, if one had two images taken from different viewpoints, and both the internal and external camera parameters are known, then the world coordinates of a scene point P can be recovered from the pixel coordinates of its projections p_1 and p_2 in the two images as the intersection of the projecting rays in the scene defined by the given image points, as depicted in Figure 7.

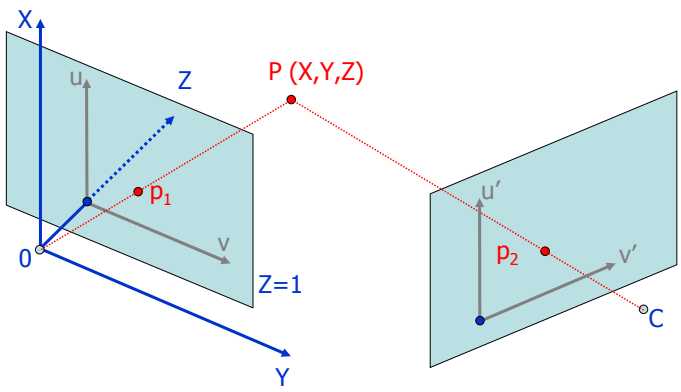

Figure 7. If the camera parameters are known, the world coordinates of a scene point P can be recovered from its projections p_1 and p_2 in the two images.

But when the camera parameters are not known, which was assumed in the general reconstruction problem, it is not immediately clear how to reconstruct the scene from

the images alone. On the other hand, one intuitively feels that every image of a static scene constrains in one way or another the shape and the relative positioning of the objects in the world, even if no information about the camera parameters is known. The key to the solution of this problem is found in the understanding of how the locations of (the projections of) an object in different views are related to each other. In what follows, we assume to have two images of the object taken from different viewpoints.

A point p_1 in one image is the projection of a scene point P that can be at any position along the projecting ray $\overline{OP}$ of the camera. Therefore, the corresponding point p_2 (i.e. the projection of P) in the second image of the same scene must lie on the straight line obtained as the intersection of the plane determined by O, C and P and the second image view, as depicted in Figure 8.

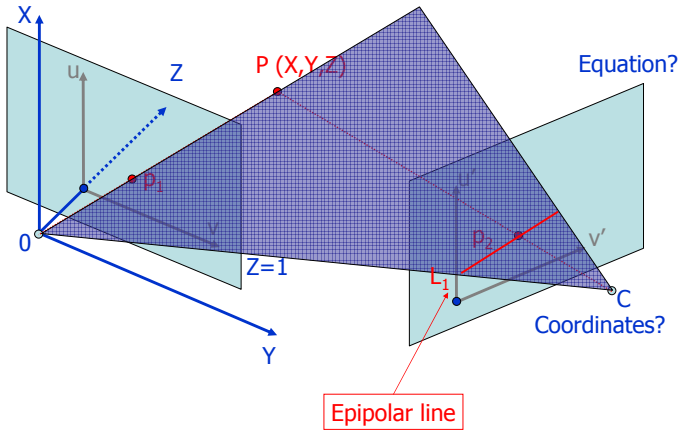

Figure 8. The point p_2 in the second image corresponding to a point p_1 in the first image lies on the epipolar line L_1 which is the intersection of the plane determined by O, C and P and the second image view.

We now have the following proposition.

Proposition 3.1. (Epipolar constraint) *For every two views I_1 and I_2 of a 3D object, there exists a 3×3-matrix F of rank 2, called the* fundamental matrix *of the image pair (I_1, I_2), with the property $p_2^t F p_1 = 0$ for each pair of corresponding points $p_1 \in I_1$ and $p_2 \in I_2$.*

Proof. Recall the general projection equation given by equation (5). Applied to

the first camera, we obtain

$$\rho_1 p_1 = K_1 P, \tag{6}$$

from which it follows that $P = \rho_1 K_1^{-1} p_1$. Combined with the general projection equation

$$\rho_2 p_2 = K_2 R^t (P - C) \tag{7}$$

for the second camera, we find that

$$\rho_2 p_2 = \rho_1 K_2 R^t K_1^{-1} p_1 + K_2 R^t (O - C). \tag{8}$$

Using the general projection equation, the second part of the right hand side of this equation can be seen as the projection of the projection center (O) of the first camera in the second view. This point is called the *epipole in the second view* and will be denoted by e_2. Therefore we have that

$$K_2 R^t (O - C) = \rho_e e_2 \tag{9}$$

for some constant ρ_e.

Let

$$A = K_2 R^t K_1^{-1}. \tag{10}$$

Then equation (8) becomes $\rho_2 p_2 = \rho_1 A p_1 + \rho_e e_2$. If we now take the cross product with e_2, we obtain $\rho_2 (e_2 \times p_2) = \rho_1 (e_2 \times A p_1)$ and from this it follows that $p_2^t (e_2 \times A p_1) = 0$. If F is defined as the product

$$F = [e_2]_\times A \tag{11}$$

of the skew-symmetric 3×3-matrix $[e_2]_\times$ representing[2] the cross product with e_2 and the matrix A, then the proposition follows. □

[2]For $p = (a, b, c) \in \mathbb{R}^3$, the skew-symmetric 3×3-matrix

$$[p]_\times = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}$$

is such that $[p]_\times v = p \times v$ for all $v \in \mathbb{R}^3$.

If the corresponding points p_1 and p_2 in the two images I_1 and I_2 are given with pixel coordinates (x_1, y_1) and (x_2, y_2) respectively, then the epipolar constraint can be written as

$$\begin{pmatrix} x_2 & y_2 & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = 0,$$

from which it follows immediately that the fundamental matrix can be computed (up to a scale factor) from the two images alone if one can identify at least 8 pairs of corresponding points in the images [2]. This conclusion yields a practical method for computing the fundamental matrix and the underlying epipolar geometry (epipolar lines and epipoles). A popular algorithm for computing the fundamental matrix is the so-called *normalized 8-point algorithm* which can be refined by iteratively minimizing some error function [4]. Automated ways of computing the fundamental matrix are developed based on a RANSAC method [3]. An overview of computational algorithms can be found in [5].

From the epipolar constraint it also follows that, if the fundamental matrix is known (by applying one of the mentioned methods for instance), the corresponding point p_2 in the second image of a point p_1 in the first image can be found along the epipolar line of p_1 in the second image. Consequently, the search for corresponding points can be simplified to a 1-dimensional search. Usually, such search algorithms maximize a similarity measure (like for instance *cross-correlation* on the gray scale images) along the epipolar line to find the "best" corresponding match (see Figure 9 for an illustration). If we denote the two gray scale images by I_1 and I_2 and $p_1 \in I_1$ and $p_2 \in I_2$ are two points given with resp. pixel coordinates (x_1, y_1) and (x_2, y_2), then the cross-correlation, based on a $((2r+1) \times (2c+1))$-window (with $r, c \in \mathbb{N}$), between these points is given by

$$\rho(p_1, p_2) = \frac{\sum_{i=-r}^{r} \sum_{j=-c}^{c} \left(I_1(x_1+i, y_1+j) - \overline{I_1}(x_1, y_1)\right)\left(I_2(x_2+i, y_2+j) - \overline{I_2}(x_2, y_2)\right)}{\sqrt{\sum_{i=-r}^{r} \sum_{j=-c}^{c} \left(I_1(x_1+i, y_1+j) - \overline{I_1}(x_1, y_1)\right)^2} \sqrt{\sum_{i=-r}^{r} \sum_{j=-c}^{c} \left(I_2(x_2+i, y_2+j) - \overline{I_2}(x_2, y_2)\right)^2}}.$$

From equation (11), combined with equations (9) and (10), it follows that the fundamental matrix F contains the internal (the calibration matrices K_1 and K_2)

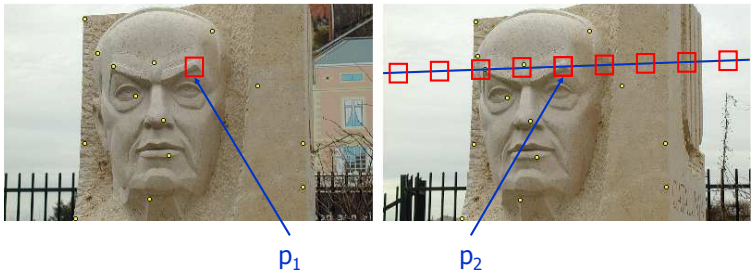

Figure 9. The search for corresponding points in two images can be done by maximizing a similarity measure, like for instance *cross-correlation* based on some window in the gray scale images, along the epipolar line. In this way the "best" corresponding point is found.

and the external camera parameters (the rotation matrix R and the position matrix C). In order to solve the reconstruction problem, these matrices need to be extracted from F. Since $e_2^t F p_1 = e_2^t [e_2]_\times A p_1 = e_2^t (e_2 \times A p_1) = 0$ for an arbitrary point p_1, $e_2^t F = 0$. Consequently, e_2 is the unique 3-vector in the left null-space of F with third coordinate equal to 1. The matrix A however can only be determined up to three parameters [7]. So obtaining a metrical reconstruction from two images is not possible. However, it is shown that a metrical reconstruction of a 3D object from three images, taken with the same camera, is possible [5].

On the other hand, suppose that the internal camera parameters (the calibration matrices K_1 and K_2) are known[3]. Then, using equation (4), the pixel coordinates p_1 and p_2 and the induced geometrical (u, v)-coordinates p_{1_u} and p_{2_u} for two points lying in the two images respectively, are related by

$$p_1 = K_1 p_{1_u} \text{ and } p_2 = K_2 p_{2_u}. \tag{12}$$

Consequently,

$$p_2^t F p_1 = {p_2}_u^t K_2^t F K_1 p_{1_u}, \tag{13}$$

and if we denote

$$E = K_2^t F K_1, \tag{14}$$

[3]Remark that this is not the original posed problem.

then the epipolar constraint given in propostion 3.1 can be rewritten as

$$p_{2_u}^t E p_{1_u} = 0. \tag{15}$$

Matrix E is called the *essential matrix* and was first introduced in [6].

We now have the following proposition.

Proposition 3.2. **[5]** *If $E = USV^t$ is the singular value decomposition of the essential matrix E, then*

$$R^t = U \begin{pmatrix} 0 & -\delta & 0 \\ \delta & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^t$$

with $\delta = \pm 1$. Furthermore, $C = \pm RU_3$, with U_3 the third column of the matrix U.

It follows from this proposition that, taking the different signs into account, four possible combinations of R and C can be considered. Combining these with the general projection equations (6) and (7), it is possible to determine up to a global scale factor the coordinates of the point P. Summarizing, if the internal parameters of the cameras are known, then a metrical reconstruction can be obtained from only two images. This conclusion is exploited by the software environment ReconLab that is presented in the next section.

4. THE SOFTWARE ENVIRONMENT ReconLab

4.1. THE AIMS SET FOR ReconLab

In computer vision the focus has mostly been set at designing *automated* procedures that can be executed "in batch" by a computer without user interaction. However, the complexity of the 3D reconstruction problem and of the distinct processes its solution consists of, make that it can (partially) fail quite easily on several levels of the "production process". And, in cases it doesn't fail, inevitabily errors and inaccuracies can occur making the end result useless. The conclusion is that even today, human supervision is still needed. With this philosophy in mind, ReconLab is

created at the K.U.Brussel (Catholic University of Brussels, Belgium) by the authors of this paper as an experimental tool for performing 3D reconstruction. Some parts of the reconstruction pipeline, as described above, are redesigned for ReconLab in order to allow user interaction at all stages of the reconstruction process. Briefly, ReconLab must be a *reconstruction laboratory* that:

- allows to experiment with different reconstruction algorithms,
- can handle multiple images and a 3D model *simultaneously*,
- *maintains relations* between image features and the 3D model,
- allows processing modes ranging *from interactive to automatic*,
- combines machine accuracy with *high-level* user control,
- allows *editing and enhancing* of image features and the 3D model,
- can run on both the MS Windows and on the Linux platform,
- is intuitive and user friendly to handle.

Since ReconLab is work in progress, not all of these aims are fully reached at this moment. Nevertheless, the current version 1.1 is capable of performing most of these tasks quite well based on only two digital images.

4.2. THE MODELING CAPABILITIES OF ReconLab

The software environment ReconLab consists of the 3 main parts: the *2D environment*, the *3D environment* and the *model environment*, which represent the usual phases in the "production process" of a 3D model starting from two images of the object. In the 2D environment, the goal is to construct a dense grid on one image and its corresponding points in the second image. In the 3D environment, the 3D points are computed from the corresponding points. Finally in the model environment, a VRML-model (this is a Virtual Reality Modeling Language - model) is constructed from the 3D points. The three parts in ReconLab are all connected to each other by

means of a central internal data structure. In this way it is guaranteed that changes on one level automatically influence all other levels as it should be.

In the 2D environment the epipolar geometry is constructed from two images. This is done by first estimating the fundamental matrix based on the 8-point algorithm. The choice of this algorithm is not essential. In fact it can be altered or combined with other algorithms very easily. We have choosen this algorithm because of its simplicity. In order to increase the accuracy, normalization and rank 2 correction [4] can be switched on and additional corrections are possible to compensate for possible inaccuracies. As a consequence of the epipolar constraint (see section 3), 8 corresponding points suffice for the computation of the fundamental matrix. However, a more accurate and stable result is obtained by using more points. We usually select 15 corresponding points and this is done manually. In this way, the selection of keypoints can be done very efficiently. Tools are available for the user to select points with subpixel-accuracy.

Once the fundamental matrix is estimated, the epipolar geometry can be used to match points from a dense grid (both regular (i.e. fixed distance) grids, and irregular grids can be used) on top of the first image to the corresponding points in the second image. This is done by scanning the corresponding epipolar line in the second image in search of the point with the largest cross-correlation with the original point (as explained in section 3). Moreover, this scanning can be followed by a *back-correspondence* algorithm in order to decrease the number of false matches.

In the 3D environment, the 3D points are reconstructed from the corresponding points found in the 2D environment. Since it is impossible to obtain a metric reconstruction from only two images, additional information about the camera is used. More precisely, we assume rectangular pixels (i.e. $skew = 0$) and known principal point and aspect ratio. The focal length of the lens can be estimated automatically, but this is very sensible to noise, which makes it not very useful when only two images are used. Therefore, more accurate reconstructions are obtained if the focal length is read directly from the EXIF-header of the digital images.

In the 3D environment the reconstructed 3D points can be visualized as a *point*

model or as a *solid model* (for which a Delaunay triangulation on the grid is extruded in the direction of the z-axis). The images in Figure 1 were all obtained using ReconLab. Options for both the point and the solid model are amongst others:

- the choice between central projection or parallel projection,
- the change in viewpoint and view direction,
- zooming in and out,
- the use of different coloring schemes (slope of normal, height, RGB, original image),
- rescaling the z-axis,
- autocentering of the image (w.r.t. x and y or w.r.t. x, y and z),
- changing the thickness of lines and/or points,
- include level curves (only for solid model).

The final stage in the production of a 3D model is the actual construction of a VRML-model. This is done in the model environment of ReconLab and can result in a *point model*, a *solid model* or a *texture model*. In the latter, one of the images from which the reconstruction is made, is mapped as texture on top of the solid model.

4.3. THE EDITING/ENHANCING CAPABILITIES OF ReconLab

As it is mentioned before, one of the goals of ReconLab is that the user should have the possibility to interact with the system at all stages of the reconstruction process. Consequently, a number of editing and enhancing capabilities are built into ReconLab. More precisely, at the moment the editing/enhancing capabilities consist of

1. in the 2D environment:
    - selecting regions of interest in the images,
    - low-level editing (moving, deleting and inserting) of individual matched points,

- inserting straight lines in the 2D environment,

2. in the 3D environment:
   - low-level editing (moving and deleting) of individual 3D points,
   - several techniques for interactively selecting and unselecting parts of the 3D reconstruction (e.g. the z-filter or polygon selection),
   - filtering (e.g. isolated point removal) and smoothing techniques (e.g. inverse distance weighting),
   - segmentation into connected regions,
3. in the model environment:
   - low-level editing of the triangulation (selecting/unselecting individual triangles),
   - selecting/unselecting regions of triangles.

4.4. EXAMPLES

In Figures 10 and 11 two 3D reconstructions are shown. The two original digital images are shown from which the 3D model is constructed and the 3D model is shown

Figure 10. 3D reconstruction of a totempole. On the left side are the two original digital images.

from different viewpoints. Also a non-textured view is shown on which the details in the model are visible.

Figure 11. 3D reconstruction of a gnome. On the left side are the two original digital images.

4.5 DEM-HANDLING IN ReconLab

We end by briefly mentioning the DEM-capabilities of ReconLab. Many building reconstruction and scene interpretation systems use a *Digital Elevation Model* (DEM) as an essential information source. Building reconstruction schemes usually use cadastre maps or manual procedures of some sort for building region delineation (see e.g. [8]). Mapping and scene classification schemes, on the other hand, often rely on a ground – above ground separation of the DEM points (see e.g. [1]). For dense urban areas complicating factors for this separation task are the relatively low number of ground points in comparison to above ground structures and – for a great number of towns in Europe – significant variations in terrain slope, in which case altitude is no longer an absolute indication for ground or above ground structures.

Moreover, the identification of erroneous 3D data points in the DEM always remains an important point of attention.

Therefore, a semi-automatic procedure, to efficiently extract a Digital Terrain Model (DTM) from a DEM of urban areas with significant variations in terrain slope and altitude, is built into ReconLab. The line of reasoning consists of segmenting the DEM into connected surface regions, identifying the regions with the largest extent, verifying whether they belong to the ground level and robustly fitting a parametric surface model to the ground points. Popular segmentation methods are based on watershed types of algorithm. Such an approach, however, may yield poor results when considerable variation in terrain slope and altitude is present in the scene. The method we use maximally exploits the proximity of DEM points to perform the segmentation task, thus being less sensitive to surface slope or altitude. In Figure 12 the distinct steps in the segmentation process are applied to a region in Amiens (France) and visualized with ReconLab. For more details, we refer to [9, 10].

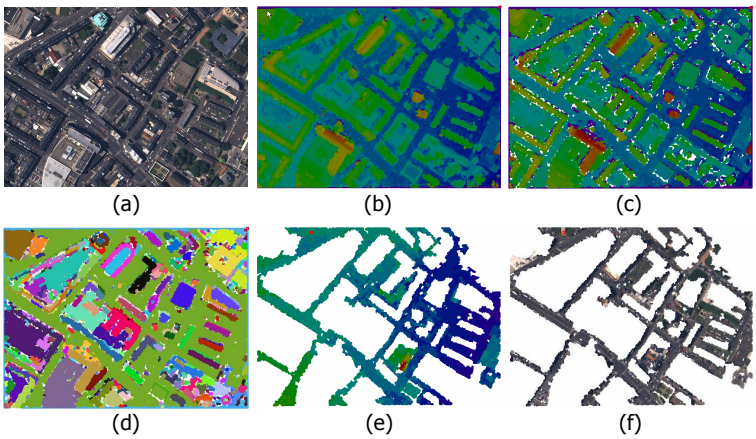

Figure 12. Results of ground level extraction for a sub-area of the airborne laser DEM of the Amiens region. **(a)** Part of an aerial image showing the sub-area of the Amiens region. **(b)** Part of the original DEM with altitude coloring. **(c)** The DEM part after smoothing and isolated point removal. **(d)** The result of automatic segmentation. **(e)** The automatically extracted ground level. **(f)** The extracted ground level with texture mapping for visual verification.

References

[1] M. Cord, M. Jordan, J. P. Cocquerez, *Accurate building structure recovery from high resolution aerial images*, Computer Vision and Image Understanding **82 (2)** (2001), pp. 138–173.

[2] O. Faugeras, *What can be seen in three dimensions with an uncalibrated stereo rig*, in: G. Sandini (ed.), Computer Vision – ECCV'92, LNCS **588**, Springer-Verlag, Berlin (1992), pp. 563–578.

[3] M. A. Fischler, R. C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Comm. of the ACM Vol. **24** (1981), no. 6, pp. 381–395.

[4] R. Hartley, *On defence of the 8-point algorithm*, Proc. of the 5th International Conference on Computer Vision (ICCV 1995), Cambridge, MA, IEEE Computer Society Press, Los Alamitos, CA (1995), pp. 1064–1070.

[5] R. Hartley, A. Zisserman, *Multiple view geometry in computer vision*, Cambridge Univerity Press, Cambridge (UK) (2001).

[6] H. C. Longuet-Higgins, *A Computer algorithm for reconstructing a scene from two projections*, Nature Vol. **293** (1981), pp. 133–135.

[7] T. Moons, *A guided tour through multiview relations*, in: R. Koch, L. Van Gool (eds.), Proceedings of the SMILE Workshop — 3D Structure from Multiple Images of Large-scale Environments, LNCS **1506**, Springer, Berlin (1998), pp. 302–345.

[8] T. Moons, D. Frère, J. Vandekerckhove, L. Van Gool, *Automatic modelling and 3D reconstruction of urban house roofs from high resolution aerial imagery*, in: H. Burkhardt, B. Neumann (eds.), *Computer Vision — ECCV'98*, LNCS **1406**, Springer, Berlin (1998), pp. I.410–I.425.

[9] I. Van de Woestyne, M. Jordan, T. Moons, M. Cord, *A software system for efficient DEM segmentation and DTM estimation in complex urban areas*, Proc. of ISPRS Congress 2004, Istanbul, Vol. **XXXV**, Part B (2004), pp. 134–139.

[10] I. Van de Woestyne, T. Moons, *3D modeling and editing with* ReconLab, Technical report of the Tournesol Meeting in Cergy (France), (2003), preprint, 14 p.