UNIVERSITÀ DI PISA

Facoltà di Scienze
Matematiche Fisiche e Naturali

Corso di Laurea Specialistica
in Tecnologie Informatiche

Tesi di laurea

Image-to-geometry registration using mutual correspondences

Candidato:

Michele Sottile

Relatori:	*Controrelatore:*
Dott. Matteo Dellepiane	**Dott.ssa Gianna Del Corso**
Dott. Paolo Cignoni	

Anno Accademico 2008/09

A tutti coloro da cui ho appreso

All'alba del primo sole, quando uomini e bestie vivevano da pari e la terra e il cielo non avevano nome, gli dei crearono Sogno e Conoscenza e li inviarono sulla terra. Sogno per ispirare e Conoscenza per istruire. Per secoli vagarono tra gli uomini ma nessuno fu in grado di accoglierli: alcuni si perdevano nel sonno, altri perdevano il senno.
Delusi, Sogno e Conoscenza abbandonarono questo universo, decisi a non farvi mai più ritorno. Sulla terra avevano concepito due gemelli, Arte e Scienza, e a loro cedettero il compito di elevare il genere umano dal suo stato bestiale. Essi riuscirono a toccare tutte le genti e l'era dell'Uomo ebbe inizio.

Michele Sottile

Image-to-geometry registration using mutual correspondence

Michele Sottile

Dipartimento di Informatica

Università di Pisa

Abstract

The thesis proposes a new methodology to the alignment of photographic images to 3D models. It is based on a semi-automatic technique. The approach needs user defined correspondences between images and the 3D model and the similarity between a rendering of the model and the original photo, measured through mutual information techniques.

A deep analysis of the existing methodologies led to the choice and the combination of these two approaches. It solves their flaws, keeping their strengths. The thesis presents a tool, called TexAlign Suit, developed for Windows and Unix platforms. It implements the proposed methodology along with other existing approaches.

Specific tests have been prepared in order to prove the efficacy of the MC approach in a wide set of use cases. They show great flexibility and user-friendliness.

Image-to-geometry registration using mutual correspondence

Michele Sottile

Dipartimento di Informatica

Università di Pisa

Sommario

La tesi propone un metodo innovativo per l'allineamento di immagini fotografiche su modelli 3D basato su una tecnica semi automatica. L'approccio proposto sfrutta sia corrispondenze tra immagine e modello scelte da un utente che la similarità tra la vista del modello renderizzato e la foto originale misurata mediante tecniche di mutua informazione.
Una approfondita analisi delle tecniche esistenti ha portato alla scelta e alla combinazione di questi due approcci. Essa permette di risolvere i loro difetti, mantenendone i pregi. La tesi presenta un tool, chiamato TexAlign Suit, sviluppato per piattaforme Windows e Unix. Esso implementa il nuovo approccio, affiancandolo a quelli già esistenti.
Test sviluppati ad hoc mostrano la validità della metodologia proposta. Essi fanno riferimento a casi d'uso diversi tra loro evidenziando la flessibilità e facilità di utilizzo della Mutual Correspondences.

Contents

Chapter 1

Introduction

In Computer Graphics, great efforts have always been made to generate a believable rendering of a 3D model. Several methods have been proposed, but the ones that assure the best resemblance with an existing object require accurate 3D surface data and surface textures. Nevertheless, the acquisition of the appearance of an object presents problems of different nature.

3D geometry of existing objects is usually acquired through 3D scanning, a highly diffused and relatively inexpensive technology. 3D scanners are essentially measurement devices: they allow to obtain accurate measurements of the geometry of a scene or an object in a very short time. Through the years costs decreased while to performances improved a lot. Hardware and software developments shortened the time needed to complete a scanning campaign (from the acquisition of the range maps to the production of the final model) from months to a few days.

Unfortunately 3D scanning can't provide information about surface properties (color and material). The best way to obtain a complete acquisition of the visual appearance of an object is taking pictures of the object.

Digital photos provide detailed 2D textures. Nowadays, even cheap digital cameras can take high definition images including additional information as focal length or sensor size. Exif file became a standard for the description and organization of these data.

Once 3D model and textures are available, a mapping between the two data sources is necessary. Images need to be registered to the 3D model. This means that the virtual camera should be disposed according to the parameters of the digital cameras. Estimation of the (extrinsic and intrinsic) camera parameters associated to each image is the real problem to solve. Many approaches have been proposed. They differentiate for the technology used (cheap labs versus expensive equipment), the level of user control in the registration process or the formalization of the problem itself. These methods have found a lot of practical applications, in very different fields such as Cultural Heritage, Medical technology, environments testing.

In this thesis, a new methodology to image-to-geometry registration is proposed. It is called Mutual Correspondences and originates as a refinement and improvement of the Mutual Information based alignment proposed by Corsini et al. in [MCS09]. This one is combined with the correspondences error based alignment that uses some correspondences between points on the image and on the geometry. The joint usage of these approaches allows to overcome their individual limits.

The thesis presents an open source tool called TexAlign Suite, which allows to align images on 3D models. It implements different approaches to the problem, giving the user the possibility to choose which one best suits to its task. Each approach deals with the minimization of a quantity.
The main strong-points of this tool are: a simple and user friendly interface, the possibility to handle very different cases (from very small to architectural size models), the opportunity to use different methodologies, the implementation of the Mutual Correspondences based alignment that shows the best results on a wide set of samples.

The structure of the thesis is organized as follows: Chapter 2 will give a general overview about the state of the art in the image to geometry registration. The first section will be devoted to the description of the camera model and the mathematic process that transforms 3D coordinates in 2D images according to specific camera parameters. The following section will present the main approaches to the problem divided into three main classes: correspondences based alignment (that includes the Tsai method and the error evaluation methods), feature based alignment and Mutual Information based alignment. The last section will show some practical applications of registration algorithms in three areas: color projection, 3D geometry deformation and 3D geometry completion.

Chapter 3 will present the new approach. The first section will compare the features of the different techniques together with the description of some tools that use them. The next section will describe in detail the Mutual Correspondences function.

Chapter 4 will present the TextAlign Suite tool developed on purpose. The first section will cover the user interface, after an overview on the used libraries. The second section will describe the project structure while the third will introduce the three calibration modes (correspondences error based, Mutual Information based and Mutual Correspondences based). The last section will supply a simple user guide.

Chapter 5 will present the tests results. They will show the effectiveness of the new approach with the use of graphics and screen-shots. Several test cases will be carefully analyzed in terms of visual feedbacks and convergence to the best alignment.

Finally, Chapter 6 will present the conclusion and an overview of the possible future improvements of the Mutual Correspondences approach and the TextAlign Suite.

Chapter 2

Related Work

In this chapter several known 2D-3D registration approaches will be discussed. They are arranged in three classes: correspondences based methods (Section 2.2), feature based methods (Section 2.3) and Mutual Information based methods (Section 2.4). They essentially differ in the choice of what drives the registration: user defined correspondences, object features, Mutual Information (MI) value.

All these methods face the problem of finding the camera parameters that generate a rendering of a 3D model of an object which is aligned to a 2D image of the corresponding object(Section 2.1).

Some practical applications of camera calibration and some uses of calibrated sets of image will be shown in Section 2.5.

2.1 Camera calibration

The problem of the alignment of a single image to a 3D model is closely related to the problem of camera calibration, that deals with the computing of all the parameters of the virtual camera in order to find an optimal inverse projection of the image on the 3D model.

2.1.1 The camera model

Camera parameters can be categorized into two groups [Tsa87] :

- *Extrinsic (or external) parameters*: these values are related to the position and the orientation in the space of the camera. They can be described with:
    - nine components for the rotation matrix R;
    - three components for the translation vector T;
- *Intrinsic (or internal) parameters*: these values are related to the internal characteristics of the camera. Some of them are:
    - f: the effective focal length, or image plane to projective center distance;
    - κ_i: lens distortion coefficients;
    - s_x: uncertainty scale factor for x, due to TV camera scanning and acquisition timing error;
    - *(C_x, C_y)*: computer image origin coordinates in the image plane. Sometimes it is referred as distorted center, because it's related to the distorted (or true) image coordinates, calculated with a radial lens correction;

These values allow any point P in the space to be transformed in the corresponding point on the image plane of the camera (and viceversa).

2.1.2 Mathematic radial alignment approach

Mathematically, transforming 3D world coordinates to 2D image coordinates according to the camera parameters implies resolving a series of equations. The main steps are [Tsa87] :

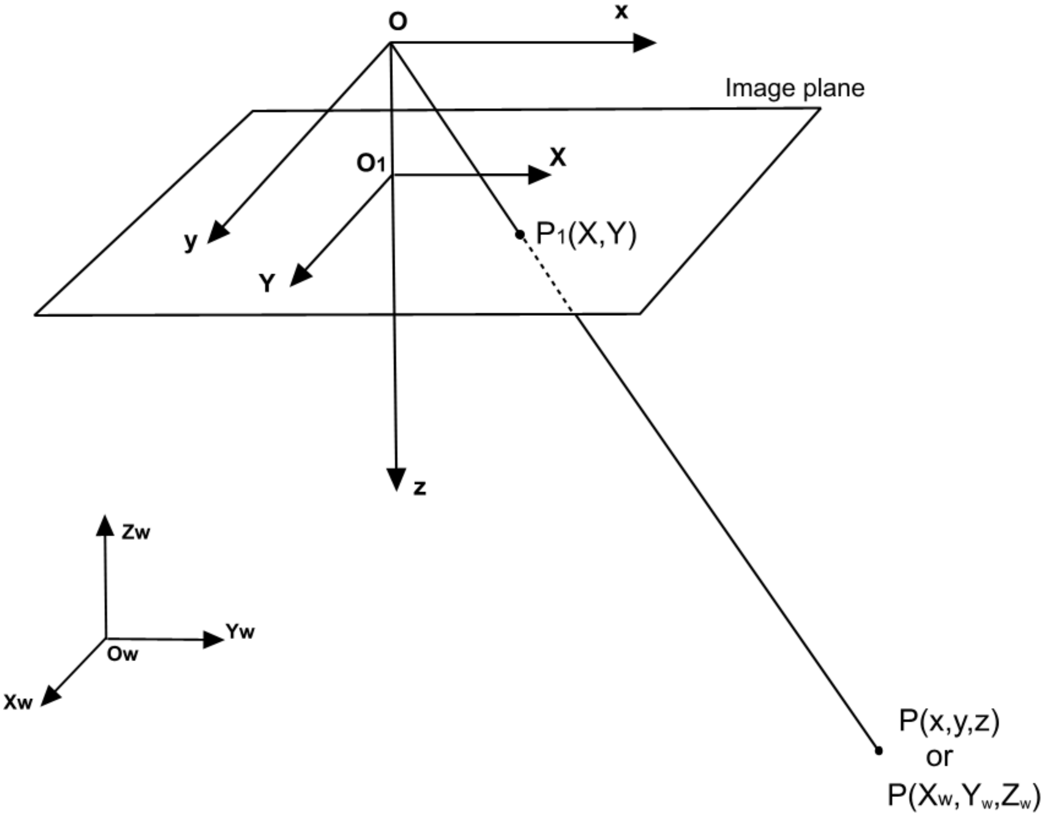


Figure 2.1: *A simple scheme of a camera perspective projection*

1. *Transformation of a point from its world coordinates (x_w, y_w, z_w) to the camera 3D coordinate system (x, y, z)*, using extrinsic parameters:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{2.1}$$

As shown in Figure 2.1 the camera is centered at point O, the optical center, and the z axis is the same as the optical axis. This is an imaginary line that defines the path along which light propagates to the system.

2. *Perspective projection with pinhole camera geometry*: transformation from 3D camera coordinate (x, y, z) to ideal (undistorted) image coordinate (X$_u$,Y$_u$) is obtained using f:

$$X_u = f\,\frac{x}{z} \quad , \quad Y_u = f\,\frac{y}{z} \tag{2.2}$$

   In Figure 2.1 O_1 is the center of projection in the image plane and f is the segment $\overline{OO_1}$.

3. *Radial lens distortion*: assuming a radial distortion produced by the lenses along the radial direction passing through the center of distortion, the undistorted (or true) image coordinates are:

$$X_d + D_x = X_u \quad , \quad Y_d + D_y = Y_u \tag{2.3}$$

   where

$$D_x = X_d(k_1 r^2 + k_2 r^4 + ...) \quad , \quad D_y = Y_d(k_1 r^2 + k_2 r^4 + ...) \tag{2.4}$$

   and

$$r = \sqrt{X_d^2 + Y_d^2} \tag{2.5}$$

   k_i are the distortion parameters. Tests show that it is possible to obtain a quite accurate model of distortion using only the first (k_1) coefficient. If the center of distortion doesn't correspond to the center of projection, the displacement p_1 and p_2 along the image axes can be estimated as well.

4. *Transformation to computer image coordinate (X$_f$, Y$_f$)*:

$$X_f = s_x d_x^{'-1} X_d + C_x \quad , \quad Y_f = d_y^{'-1} Y_d + C_y \tag{2.6}$$

   where

$(X_f,\, Y_f)$ row and column numbers of the image pixel in computer frame memory

$(C_x,\, C_y)$ row and column numbers of the center of computer frame memory (distorted center)

d_x center to center distance between adjacent CCD sensor in the X (scan line) direction,

d_y center to center distance between adjacent CCD sensor in the Y direction,

N_{cx} number of sensor elements in the X direction,

N_{fx} number of pixels in a line as sampled by a computer,

s_x image scale factor,

$$d_x^{'} = d_x \tfrac{N_{cx}}{N_{fx}}$$

s_x is an additional uncertainty parameter introduced due to a variety of factors (for example the slight hardware timing mismatch between image acquisition hardware and camera scanning hardware, or the imprecision of the timing). This step is necessary because the unit for $(X_f,\, Y_f)$ must be the number of pixels for the discrete image in the frame memory. Generally, manufacturers of the digital camera supply d_x, d_y, N_{cx} , N_{fx}.

In conclusion, in order to obtain a good registration of an image with a 3D model, it's necessary to estimate 12 extrinsic parameters (9 for rotation and 3 for translation), and from 1 (focal length) to 5 (focal, distortion coefficient, image scale factor and center of distortion) intrinsic parameters (step 2-4).

2.2 Correspondences based estimation of camera parameters

The most generic and robust way to align a set of images on a 3D model is to *provide a set of 2D-3D points correspondences*, and estimate the parameters by solving a non-linear equation system, or minimizing a pre-defined function. Part of the intrinsic parameters can be provided by the camera manufacturer or included in the Exif file of the digital photo, or it can be estimated preliminary and assumed as constant for every acquired image.

Several automatic and semi-automatic systems have been created to estimate intrinsics parameters. For example, the approach by Zhang [Zha00] provides very good results using a very simple calibration pattern. Once that intrinsics are known in advance, it is necessary to estimate only the position in the space of the camera: this problem is known as pose estimation, and one of the most comprehensive solutions was provided by Kumar [KH94].

2.2.1 The Tsai method

In the case of a more general problem, where all the parameters have to be estimated, several possible mathematic approaches have been proposed, but the most widely known and used is the **Tsai method** [Tsa87] that implements step by step the algorithm shown in Section 2.1.2. It uses a two-stage technique to compute, at first, the position and orientation and, then, the internal parameters of the camera.

The Tsai model describes the camera using 11 parameters: 6 for extrinsics (3 for translation components, 3 for rotation components) and 5 for intrinsics (1 for focal length and 4 for lens distortion, scale factor, and distorted center). The rotation matrix R is expressed with the Euler angles yaw θ , pitch ϕ and tilt ψ for rotation:

$$\begin{pmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\cos\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{pmatrix}$$

The method is implemented for both coplanar and non-coplanar points constellation, and it proves to be very simple and robust. With 5 or 7 correspondences respectively, an overdetermined system of linear equation can be established and solved, though 12 correspondences give the most reliable results. The best solution is found regardless of the initial position of the model. Another strong point is that an implementation of the method is freely available on the web [Ali04].

2.2.2 Error evaluation methods

Almost all the calibration procedures are led back to a minimization problem. They differentiate mainly for the function chosen for the estimation. Among these, **error minimizing functions** are very interesting. For example, the error can be computed as the mean distance in pixel between the 2D image point and the corresponding 3D point projected on the image plane for each correspondence.

$$E(Corr, C) = \frac{\sum_{\forall corresp_i \in Corr} \sqrt{(x_{p_i(C)} - x_i)^2 + (y_{p_i(C)} - y_i)^2}}{N} \tag{2.7}$$

$$C = (\theta, \phi, \psi, t_x, t_y, t_z, f)$$

where:

(x_{p_i}, y_{p_i})	projected 2D point of the original 3D point of the correspondence
(x_i, y_i)	original 2D point of the correspondences
N	total number of correspondences
θ, ϕ, ψ	Eulero angles
t_x, t_y, t_z	components of translation vector
f	focal

Although such an error could not be considered a measure of the quality of the alignment, tests demonstrate that minimizing this function gives good results.

Another approach [DG97] uses a linear and non linear method to the simultaneous estimation of both the intrinsic and extrinsic camera parameters using point or line correspondences.
Minimization problems are typically solved through algorithms involving iterative computations (Figure 2.2). Different software implementations to minimize functions have been proposed. For example, packages like LevMar [Lou04] and Newuoa [Pow04] are freely available on the web.

2.3 Feature based estimation of camera parameters

Many efforts have been made to automatically align a set of uncalibrated images to a 3D model. An automatic planning of a number of images and of the positions of the camera could lead to good results [MK99]. After choosing a number of virtual camera position that covers the entire surface of the 3D model, the user takes photos that resemble the renderings of the model from each camera positions (Figure 2.3). Though this approach reduces the importance of the registration step, it cannot be considered as a general

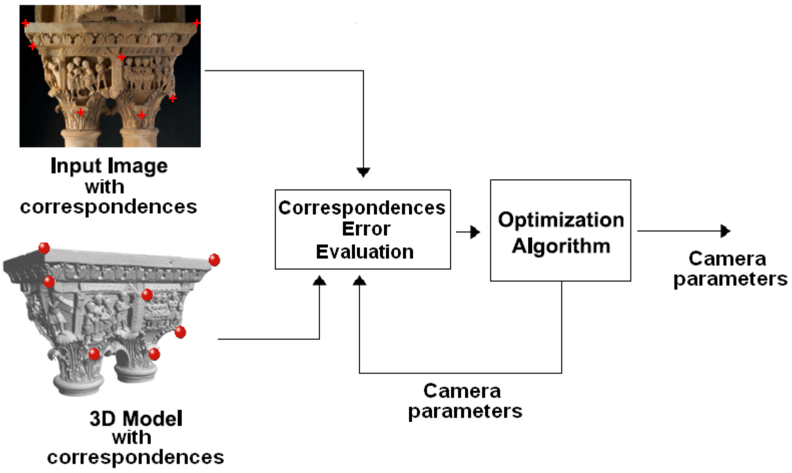


Figure 2.2: *A sketch of the Correspondences registration algorithm*

solution, since it would be needed to put the object to acquire inside a specific setup environment. This is usually impossible especially for the application in the Cultural Heritage context, where frequently the objects must not even be touched during acquisition.

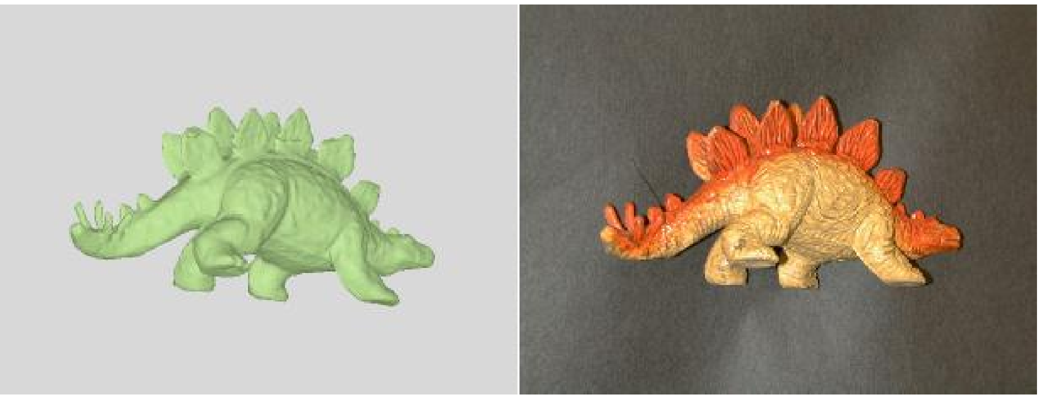


Figure 2.3: *A reference image from a selected virtual camera position (left) and the relative digital camera (right)*

In order to achieve a more robust camera calibration, several methods that exploit image features have been proposed. A feature can be defined as any significant piece of information extracted from an image which provides more detailed understanding of it. In other words, features are objects, which should be distinct, salient and preferable frequently spread over the image and easy to detect. Feature based image registration relies on extracting and matching similar features from pairs of images, specifically, the rendering of the 3D model of an object and a photo of the object itself.
For example, Neugebauer et al [NK99] present an hybrid approach where the estimation based on correspondences is combined with a registration based on the analysis of the image features (Figure 2.4). This semi-automatic approach needs a preliminary calibration of the intrinsics of the camera. Moreover, one of the hypotheses is that the illumination must be the same for all the images: this brings to lower quality color information, because there will be no possibility to remove illumination artifacts during the color projection phase.

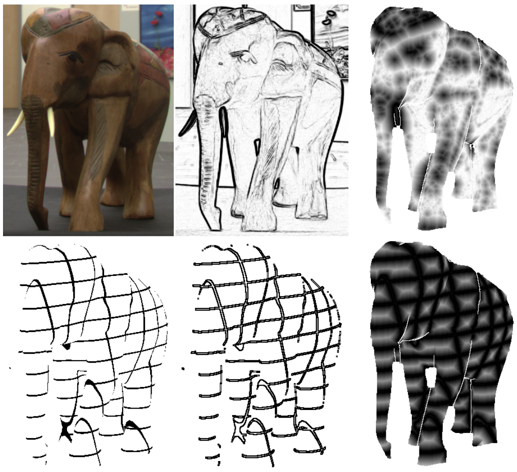


Figure 2.4: *Image analysis of an object from [NK99]: image features are explored to enhance registration to 3D model.*

Ikeuchi [IOT$^+$07] presents an automated 2D-to-3D registration method that relies on the reflectance range image. However, also in this case the algorithm requires an initial estimation of the image-to-range alignment in order to converge.

Several other algorithms try to find the camera transformation by minimizing the error between the contour found in the image and the contour of the projected 3D model [BLS92, IY96, Low91, MK99]. The error is typically computed as the sum of distances between a number of sample points on one contour and the nearest points on the other [Low91]. Another approach computes the sum of minimal distances of rays from the eye point through the image contour to the model's surface, which are computed using 3D distance maps [BLS92]. Lensch [LHS00] proposes a robust implementation of previous silhouette based techniques, introducing a similarity measure to compare them. Moreover, the whole pipeline from registration to texturing is covered with very robust and almost automatic solutions (Figure 2.5). Unfortunately, the use of silhouette matching has two important limitations: the first one is that it must be easy to distinguish the object with respect to the background: this needs controlled setup acquisition, or a time-consuming "block out" manual session. The second one is that the object must be entirely present inside each image: this can be a very big drawback when a big object must be acquired, and the aim is to preserve the detail of color. In this case, it could be necessary to frame only portions of the object, and this prevents silhouette methods to work properly.

A recent work for 3D-3D and 2D-3D automatic registration [LSY$^+$06] proposes an algorithm for a more general case, but under the assumption that the 3D scene contains clusters of vertical and horizontal lines. Analogously, other previous approaches like [LS05] need to exploit orthogonality constraints. The main application for this group of works stand in the field of architectural models (see Figure 2.6).

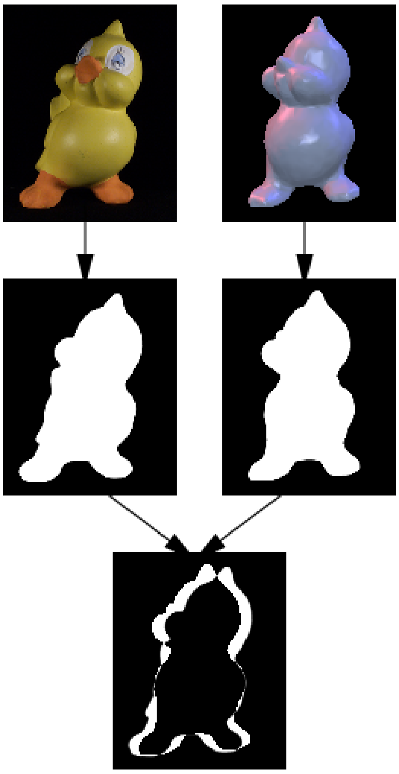


Figure 2.5: *Silhouette comparison from [LHS00]: the silhouette of image and model are compared to calculate a similarity measure.*

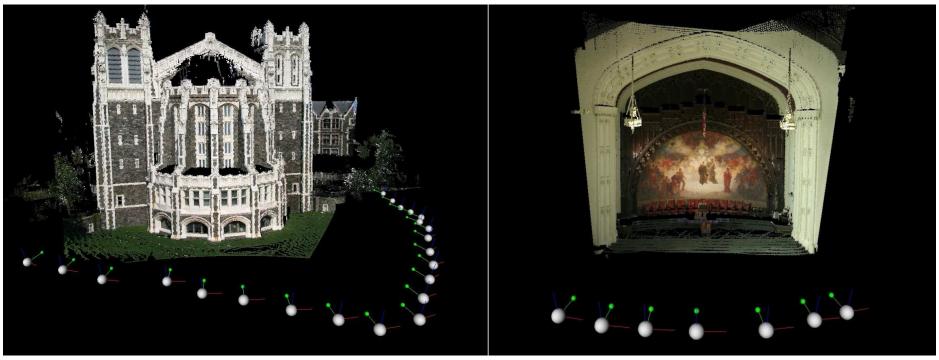


Figure 2.6: *Two results for the approach presented in [LSY+06].*

2.4 Mutual Information based estimation of camera parameters

While the above approaches try to extract geometric features (silhouette, reflectance, etc..) from both the image and the 3d model, it is possible to explore different directions, dealing with object features in another way. For example the Mutual Information (MI) can be used. MI is a statistical measure of non-linear correlation between two data sources. In this approach it is used as a similarity measure between the image to be registered and a rendering of the model geometry. It is inserted into an iterative optimization framework in order to drive the registration.

Two of the first methods implementing MI are the one by Viola and Wells [VWMW97] and the almost contemporary work by Maes et al. [MCV$^+$97]. In particular, both papers showed a surprising correlation between the images and the normal field rendering of the corresponding 3D model. After these, several registration methods based on MI have been proposed. A comprehensive overview is presented in [PMV03].

In general, most of these works regard simple geometric transformations like 2D roto-translation or affine transformation, so that some of the issues related to the camera model registration are not present. In the use of MI, the choice of the optimization strategy to achieve the maximization is critic; the comparative evaluation in [MVS99] presents the pros and cons of several methods.

Significant improvements in this field have been made by [MCS09].The base idea is rendering the 3D model according to different illumination-related geometric properties (normals, ambient occlusion and reflection direction) or others (silhouette). Actually, due to the texture and the unknown illumination conditions, the input image could appear very different from a rendering of the geometry. Therefore, using several types of renderings such as ambient occlusion, normal map, reflection map, silhouette map, and a combination of them gives more meaningful images for the registration process. Since MI evaluates a generic dependency between two random variables and the images are visually very different, MI represents a good similarity tool. It could be shown that the MI strictly relates these maps and a grey-scale version of the input image.

MI can also be seen as the amount of information about B that A contains. Mathematically MI can be expressed as:

$$MI_{(I_A, I_B)} = -\sum_{a,b} p(a,b) \lg \frac{p(a,b)}{p(a)p(b)} \tag{2.8}$$

where

I_A, I_B	images
$p(a,b)$	joint probability of the event (a,b)
$p(a)$	probability that a pixel of I_A gets value a
$p(b)$	probability that a pixel of I_B gets value b

The joint probability distribution can be estimated easily by evaluating the joint histogram (H) of the two images and then dividing the number of occurrences of each entry by the total number of pixels $N_A \times N_B$. A joint histogram is a bi-dimensional histogram made up of $N_A \times N_B$ bins; the occurrence (a,b) is associated with the bin (i,j) where $i = \lfloor a/N_A \rfloor$ and $j = \lfloor b/N_B \rfloor$ (see Figure 2.7). Each entry (a,b) is the number of occurrences an intensity a in one image corresponds to an intensity b in the other. The higher is the dispersion of joint events distribution, the lower is the value of MI. The more the images are similar the more the histogram is compact.

The registration algorithm is an iterative optimizing algorithm that tries to maximize the MI between the grey-scale input image an the the renderings

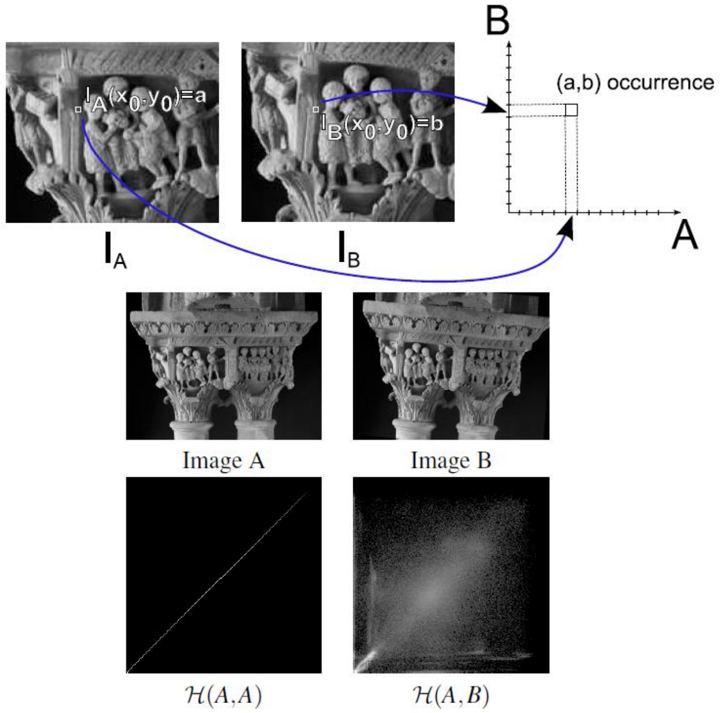


Image A

Image B

$\mathcal{H}(A,A)$

$\mathcal{H}(A,B)$

Figure 2.7: *Joint histogram. (Top) Construction. (Bottom) An example. The joint histogram of the image A with itself and the joint histogram of the same image with a rotated version of it.*
In the example $I(A,A)$ = 6.307 and $I(A,B)$ = 1.109.

of the model, estimating 7 camera parameters (6 for camera position and orientation and 1 for focal length). The normalization of the parameters plays a key role in the context of the minimization. The other intrinsic camera parameters are assumed to be pre-determined.
The problem can be formalized as the optimization of a 7D space function:

$$C^* = \arg\max_{C\in\mathbb{R}} MI(I_A, I_B(C)) \tag{2.9}$$

$$C = (\theta, \phi, \psi, t_x, t_y, t_z, f)$$

where:

I_A	preprocessed image to align
I_B	the rendering of the 3D model that depends on the camera parameters C
θ, ϕ, ψ	Eulero angles
t_x, t_y, t_z	components of translation vector
f	focal

The optimization problem (Figure 2.8) is divided into two sub-problems (the paper proposes a solution only for the latter):

- **finding an approximate solution, the *rough alignment***;
- **refining this solution into an optimal one, *the fine alignment***: the input image is converted into a grey-scale version, and, according to the actual camera parameters, a rendering of the 3D model is generated. Then the MI is evaluated.

  An iterative optimization algorithm updates the camera parameters until the best registration is achieved, that is when the the maximum value of the function MI is found.

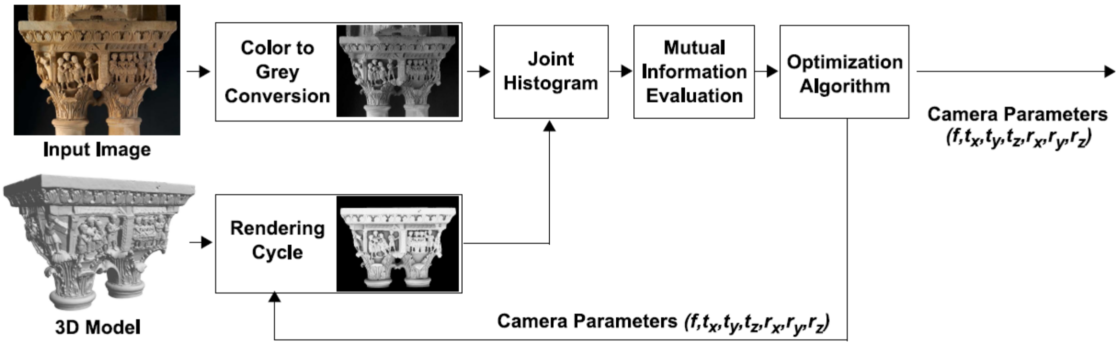

Input Image
Color to Grey Conversion
Joint Histogram
Mutual Information Evaluation
Optimization Algorithm
Camera Parameters (f,tx,ty,tz,rx,ry,rz)
3D Model
Rendering Cycle
Camera Parameters (f,tx,ty,tz,rx,ry,rz)

Figure 2.8: *A sketch of the MI registration algorithm*

The Illumination-related geometric chosen properties are:

- *Background and Silhouette*:
  the joint histogram is evaluated using the whole rendering viewport, not only the part of the image that corresponds to the 3D model rendered pixels. In doing so, is it possible to implicitly infer information about the silhouette of the object, especially when the statistics of the image background is very different form the one of the object itself. The explicit contribution of the silhouette could be considered as well, though sometimes ineffective (Figure 2.9).

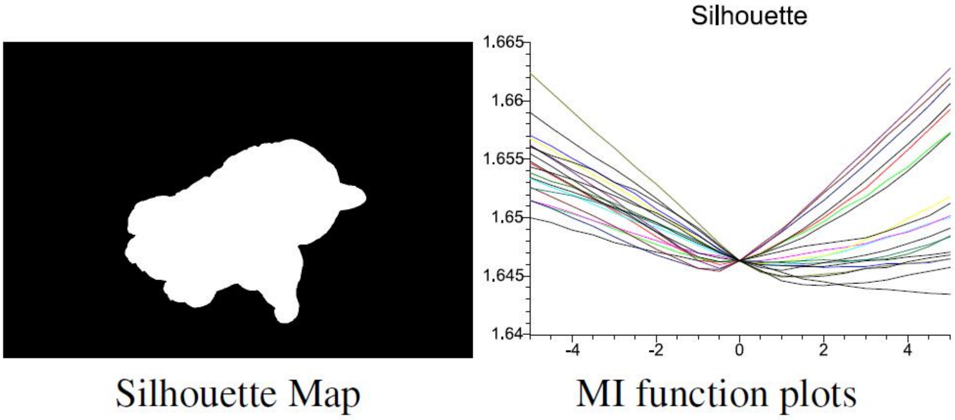


Figure 2.9: *Silhouette map performances. The silhouette information fails to register the model. The use of the internal visual information is decisive in this case.*

- *Surface Normals*:
  in the computation of the Lambertian (diffuse) lighting of a surface, normal maps achieve good results. Viola and Wells [VWMW97] demonstrate a good correlation between surface normals and the variation of

shading generated by directional light source, regardless of light direction (Figure 2.10).

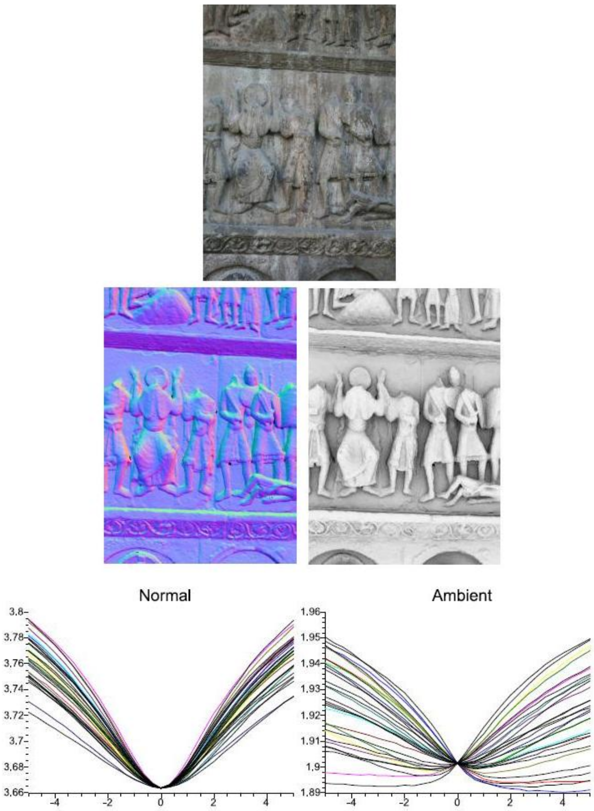


Figure 2.10: *Normal vs Ambient.(Top) Original image. (Middle)Normal and ambient occlusion maps of the aligned model. (Bottom) Corresponding MI functions plots. Dealing with a stone bas-relief, ambient occlusion doesn't provide enough lighting information to get a fine alignment. So the plot doesn't exhibit a minimum at the aligned position.*

- *Ambient Occlusion*:

  occluded part of the model will receive little light independently by the lighting environment of the scene. In case of uniform illumination or complex geometry, self-shadowing defines the object volume (Figure 2.11).

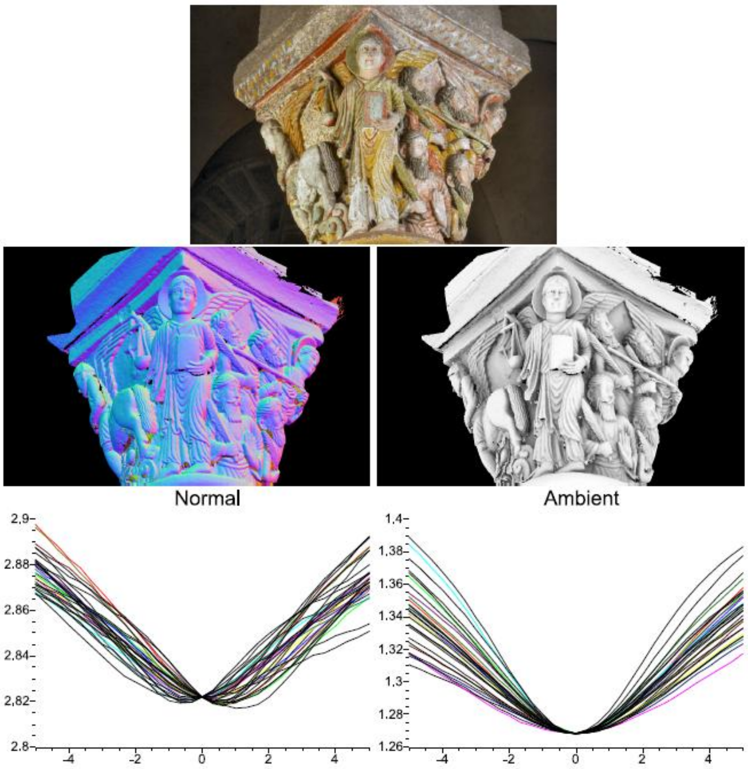


Figure 2.11: *Normal vs Ambient . (Top) Original image. (Middle)Normal and ambient occlusion maps of the aligned model. (Bottom) Corresponding MI functions plots. Dealing with the complex geometry of the stone capital and uniform illumination, the ambient occlusion let the MI function have a strong minimum.*

- *Reflection Map*:
  the rendering stores the direction of the mirror reflection. This value is simple to compute since the viewpoint, the orientation and position of the model are known in advance. For a specular material, this rendering

is statistically related to the highlights.(Figure 2.12).

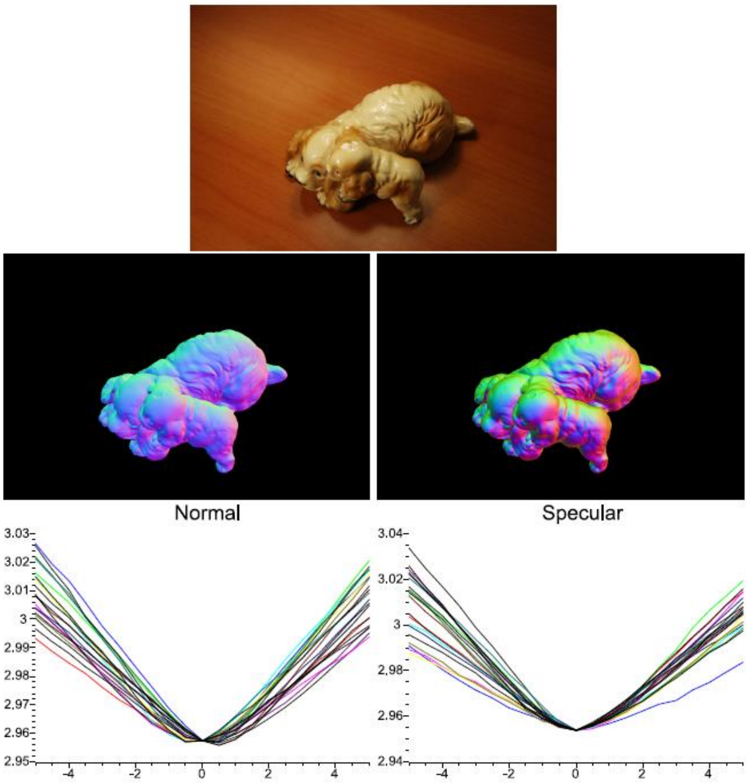


Figure 2.12: *Reflections vs Normals. (Top) Original image.(Middle) Normal and reflection maps of the aligned model.(Bottom) Corresponding MI functions plots. Dealing with an object made of specular material, the reflection map and the original image share a clear minimum of the MI plot around the aligned position.*

M. Corsini et al. [MCS09] propose a combination of normal map and ambient occlusion that ensures a robust fine alignment, regardless of the color and material of the object and the lighting environment under which the image was taken.

According to the material, the specularity, the illumination, the geometry and

the color of the object, different renderings could achieve different results. In order to compare test results, it could be helpful to graphically show the MI plot and check if there is a minimum at the aligned position (obtained with a careful semiautomatic alignment [FDG$^+$05]). Since the MI function around the aligned position is a function of seven camera parameters, the overall shape is explored around the aligned position with some 1D sections, 30 in our case, calculated in random directions in the 7D space; where the MI has a global minimum every section should exhibit the same minimum.
Each camera parameter needs to be scaled so that the increment of 1 unit produces an alignment error of 1 pixel.The X axis represents the distance in the scaled parameter space, while the Y axis represents the values of the mutual information. It is important to underline that these values are not normalized, and they depend on the specific image and 3D model. The quality of the MI function is defined by its shape: the important factors are the existence of a well defined minimum and a smooth shape, which permits wider range of convergence.

Figures 2.9, 2.10, 2.11, 2.12 show MI function comparisons between different illumination-related geometric properties renderings and the original image.

Among the applications of the MI based methods, Viola and Wells [VWMW97] apply their methodology to medical data (Figure 2.13), which are often quite poor in quality.

Recently, two exploitations of MI have been proposed for non-medical applications: 3D object tracking for simple template-based objects [PK08], and image registration improvement [CS07].

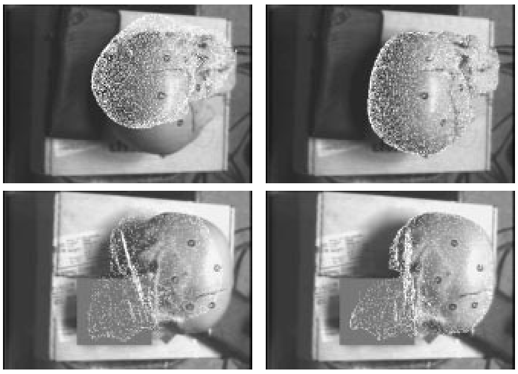


Figure 2.13: *Registration via MI presented in [VWMW97]. This is a Skull Alignment Experiments.In order: Initial Alignment, Final Alignment, Initial Alignment with Occlusion, Final Alignment with Occlusion*

2.5 Practical applications

Image registration is the preliminary step in different fields of application. One of these is *geometry enrichment*. Its basic idea is to find ways to interpret the information given by images to enrich the quality of 3D models. Given a set of images of a scene and some kind of geometric representation of it, it is possible to use geometric data to help the analysis of the content of images and the extraction of information from them.

Dellepiane [Del09] presents some methodologies to enrich digital 3D information using a set of uncalibrated images. He analyzes three main fields of application:

- projection of color data on 3D models (color mapping);
- 3D geometry deformation (morphing a starting geometry using the information extracted from a small set of unregistered images) ;
- 3D geometry completion (hole filling).

Practical applications (mainly in the field of Cultural Heritage) of the methods above mentioned are shown as well. All of them start from a set of uncalibrated image and a 3D geometry. They show that it is possible to enrich 3D geometry in a really robust way, adaptable to very different cases, from the acquisition of small objects to the visualization of very large and complex architectures.

2.5.1 Color projection: The Double David project

3D scanning and camera calibration procedures have been used for the documentation of the the status of the statue of Michelangelo's David before and after a restoration. This was mostly focused on the removal of dust, spots and other deposits accumulated in the years on its surface, and on the replacement of the plaster fillings of some fractures (e.g. the ones filling the small gaps of the fragments of the arm broken in XV cent.). The digital documentation relies on a detailed 3D model of Michelangelo's David and a set of photos.

The Dataset: high quality images and dense 3D model

The 3D model of Michelangelo's David was provided by the Stanford's Digital Michelangelo project [LPC$^+$00] (56 million triangles, reconstructed from 4000 range images using a distance field with 1mm. cell size). Since the restoration was going to bring changes mostly in terms of different appearance of the surface, a high quality photographic essay of the pre and post -restoration condition was taken(a graphic representation of the planned photo survey is shown in Figure 2.14).

The amount of 2D data collected (61 images, res. 1920x2560 to document the pre-restoration status; 68 images, res. 2336x3504 for post-restoration status) was about 800 Mega pixels. Some sample images from both sets are

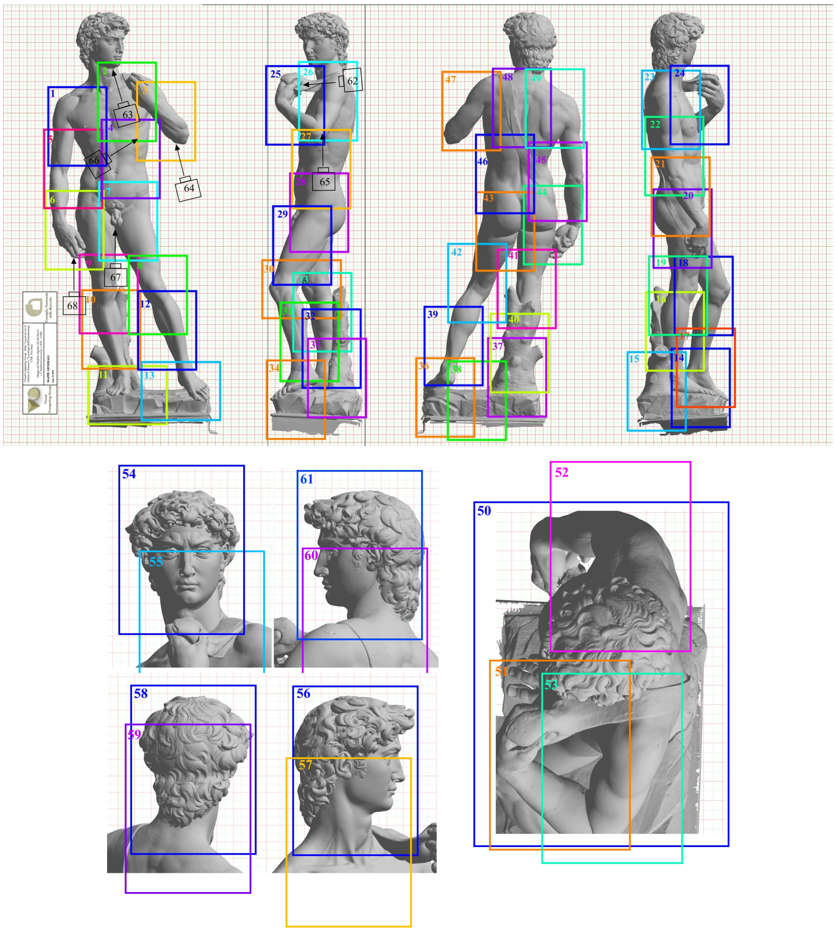


Figure 2.14: *Examples schema of the photographic campaign, describing the coverage of the set of photos.*

shown in Figure 2.15.

The restoration approach was based on a manual relief drafting, followed by a digitization phase and a final mapping. The restorers performed a precise graphic survey on the status of the David's surface. These annotations were drawn by the restorers on transparent acetate layers positioned onto each printed images, using different color to indicate the same phenomena in the different sheets. Therefore, 4 different graphic layers for each one of

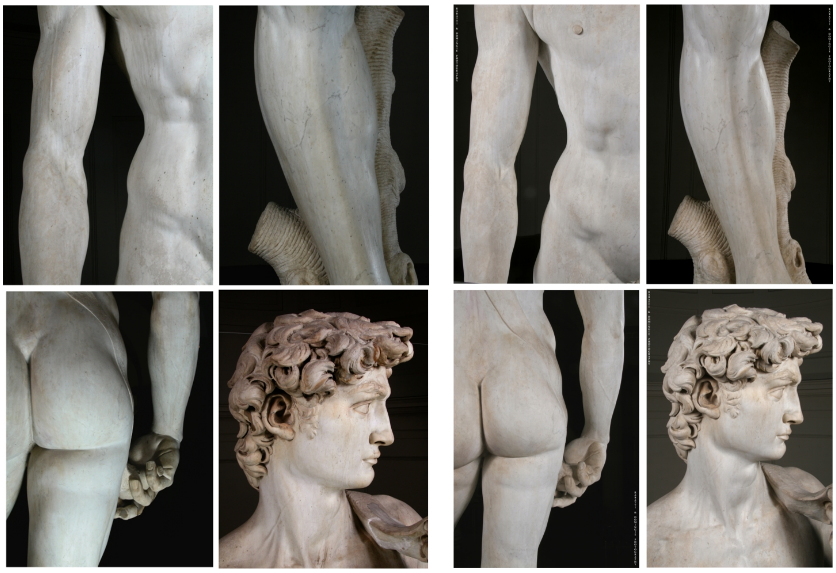


Figure 2.15: *Examples of images from the pre- and post-restoration campaign.*

the 61 high-resolution photos (documenting the pre-restoration status) were created.

Mapping and visualizing RGB data on the 3D model

In order to visualize, analyze and compare data with a paper-less mode, two approaches were chosen: (a) 'classic' 2D mapping of the scanned reliefs on the corresponding 2D RGB images, and (b) mapping of the RGB data (pre- and post-restoration images) on the digital 3D model aligning the two photo sets (129 images registered) on the high detail 3D model and projecting the color. The 2D mapping was chosen as an easy way for experienced and unexperienced users to access the photographic and reliefs archive; the user can choose any of the provided views, and then visualize the corresponding photo and, selectively upon his/her choice, the superimposed reliefs related to imperfections, deposits and deteriorations.

The 2D-3D registration and color projection make possible an interactive visualization of the colored 3D model, giving the user the possibility to analyze the appearance of the statue before and after restoration from any arbitrary point of view.

The pre and post-restoration models are shown on the left and right side, respectively (Figure 2.16) . The power of camera calibration allows to obtain an image of the restored model perfectly aligned to the starting pre-restoration photo. It is possible to use the camera parameters estimated for a photo that belongs to the pre-restoration set and render the post-restoration model from the corresponding camera position.

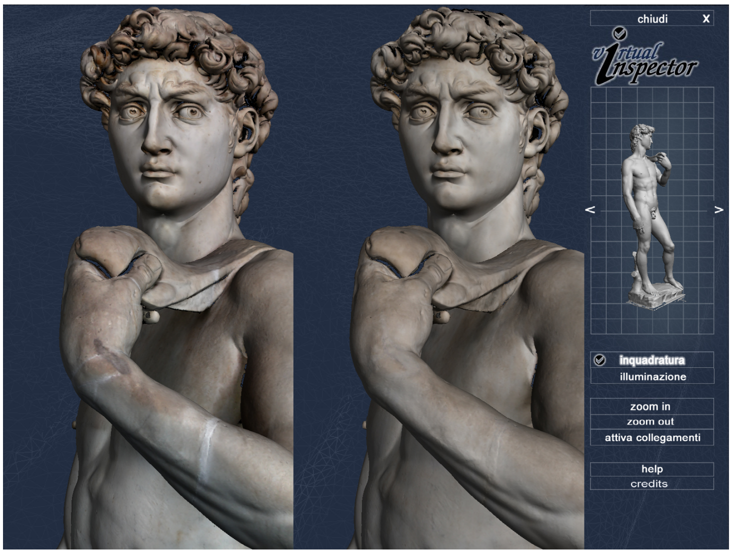


Figure 2.16: *Screenshot of the Virtual Inspector visualization tool used for inspection and virtual manipulation of a complex and highly detailed 3D model*

An example of this particular mapping is shown in Figure 2.17, where the left-most image is taken from the pre-restoration set and the central image is the corresponding one from the post restoration set. This could be very useful not only for superimposition of restoration reliefs but also to try to reproduce a similar illumination condition, which was slightly different between the two sets.

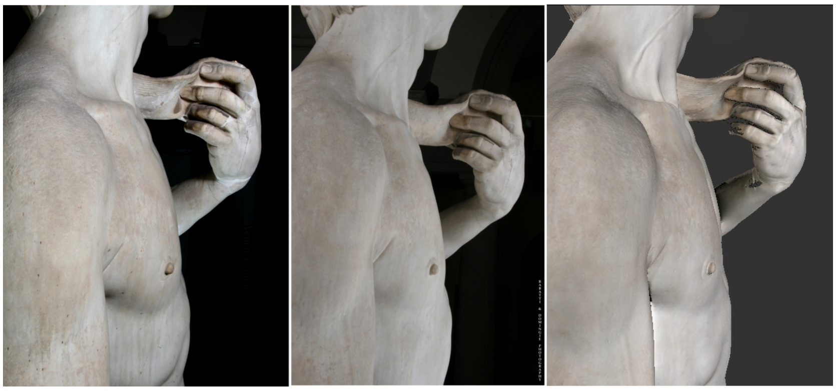


Figure 2.17: *one of the images of the pre restoration set (Left); the corresponding image in the post restoration set (Center); rendering of post restoration 3D model from camera position of pre restoration image (Right)*

In conclusion, the mapping of the David was an interesting application of color projection in Cultural Heritage. In particular, 3D navigation adds significant possibilities to compare the visual appearance of David before and after restoration. Moreover, highly detailed geometric information and highly detailed color information were put together in a real-time navigable system.

Another interesting and similar application of color projection techniques is the modeling of 3D colored scannings of the capitals in the cloister of Monreale. It belongs to the much wider CENOBIUM (Cultural Electronic Network Online: Binding up Interoperably Usable Multimedia) [BCC$^+$07] project. It was born to provide a web-based, openly accessible work environment, which includes 3D models created by scanning, CAD-representations, digitized historical photographs and digital photography of the highest pro-

fessional quality.
All the capitals of the Cloister were acquired with a set of 8 images. A set of photos, documenting a capital, is shown in Figure 2.18.

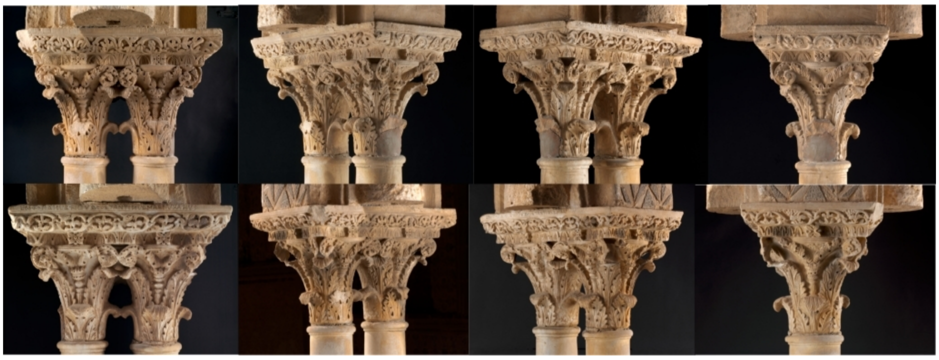


Figure 2.18: *An example of the set of photos acquired for sampling the color of each capital.*

High quality 3D models of the capitals were produced by using a Konica Minolta VI 910 Laser Scanner. After the reconstruction of an accurate 3D model, the images have been calibrated and their color projected onto it using a specific tool called TexAlign [FDG+05, FCG+05]. Twenty highly detailed 3D models of the most artistically interesting capitals of the cloister were reconstructed. The screenshots of two models are shown in Figure 2.19.

2.5.2 3D geometry deformation: 3D head model reconstruction from images for HRTF

In the context of Virtual Reality applications, in order to obtain a realistic three dimensional audio, binaural rendering is used. This technique provides an exact reproduction of the acoustic field over headphones. Realistic results can be achieved only by using individualized HRTF filters. The head-related transfer function (HRTF) describes how a given sound wave input

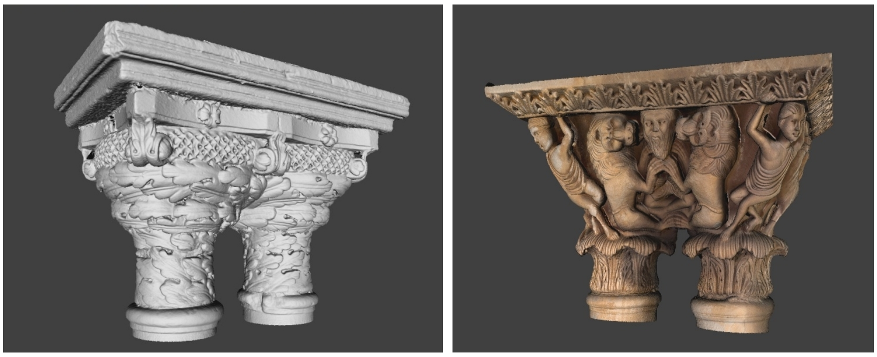


Figure 2.19: *the "Sh10" capital, ornamental leaves (Left); the model of "Sh37" capital with color information (Right).*

(parameterized as frequency and source location) is filtered by the diffraction and reflection properties of the head and pinna, before the sound reaches the eardrum and inner ear. Its shape can noticeably vary between subjects and it's closely related to the features of the head (primarily the ear, and secondarily nose, chin, head size). Recently, some methods have been introduced to simulate virtual measurements of HRTFs filters by running finite element simulations [Kat01, KN06] or ray-casting [ARM06] on a 3D head model. A collaboration between Visual Computing Lab and Reves Lab at INRIA brought to the creation of an alternative method for sound scattering calculation [TDLD07]. Instant sound scattering is based on the use of the Kirchoff approximation, with which it is possible to calculate the first bounce of sound scattering in a very short time (minutes instead of hours). In order to apply this method to HRTF calculation, it is necessary to provide an accurate model of a 3D head. 3D scanning could be expensive so, in order to reduce the cost, a new method have been proposed: an automatic method to produce an accurate model of head starting from a set of uncalibrated photos. Final model would have been obtained via the morphing of a starting 3D model from a large database of of a dummy heads (Figure 2.20) After

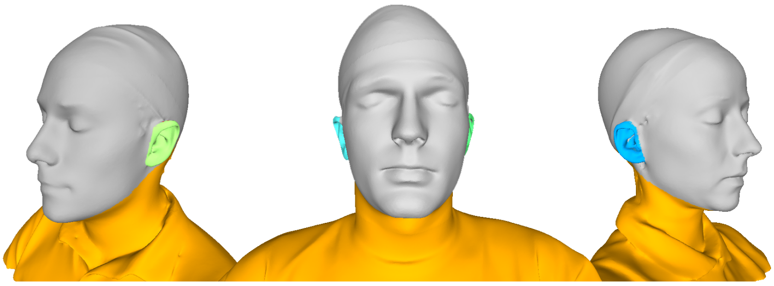


Figure 2.20: *Three elements of the 3D dummy library*

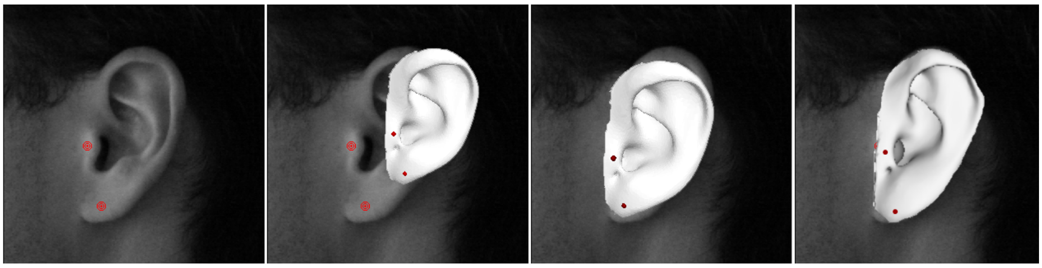


Figure 2.21: *An example of ear selection: starting image, dummy ear camera position before and after alignment, ear shape after low accuracy morphing*

extracting the external borders of both ears from images (extracted mask), the dummy head that best matches the ear features is selected for morphing. The user needs to set some key-points (6 or 7) on each photo. A model of perspective camera is used and a rigid alignment is performed, by modifying extrinsic camera parameters so that the external borders of the 3D ear are best aligned to the extracted mask (Figure 2.21). A similar approach to the one used for ears is applied to profile and frontal photos to provide best camera data for head deformation (Figure 2.22). Using the set of cameras which defines the alignment of the dummy model with respect to each image, a set of viewport-dependent 2D-to-3D model deformation is calculated. The set of deformations is then combined to morph the dummy model to its final shape.

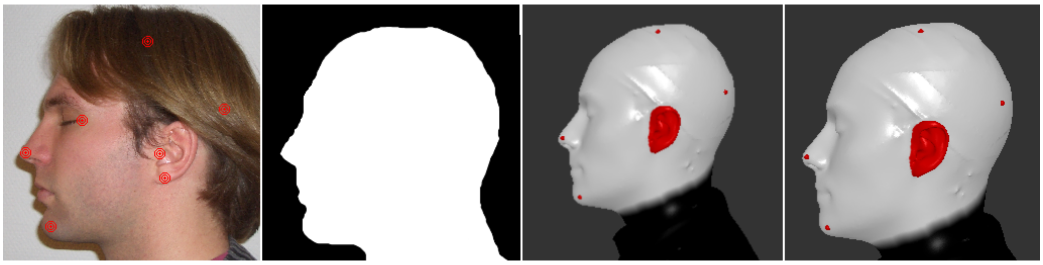


Figure 2.22: *An example of head alignment: starting image, extracted mask, dummy head camera position before and after alignment*

2.5.3 3D geometry completion: filling holes starting from calibrated images

Starting from the same previous input data (a 3D model and a set of images), some methods can recover from some errors or problems happened during 3D data acquisition. In fact, it is not unusual to have parts of the model which can be taken by a photo, but cannot be acquired by the scanner due to the angle between the camera and the laser emitter, or because it is impossible to find the position of the scanner which permits to "close the hole". So, an open model results. This time the images of the object need to be calibrated. The input dataset is made by a 3D open model of the object, a image depicting the pattern on the object (Figure 2.23) and the camera data which define the alignment of the image (obtained with TexAlign [FDG$^+$05]).

The hole reconstruction framework [DVS09] can be divided in three main parts:

1. **Image processing.** The first operation is the extraction of the information about the pattern from the image. It is thus necessary to apply image filters in order to extract the red lines from the image (Figure 2.23).

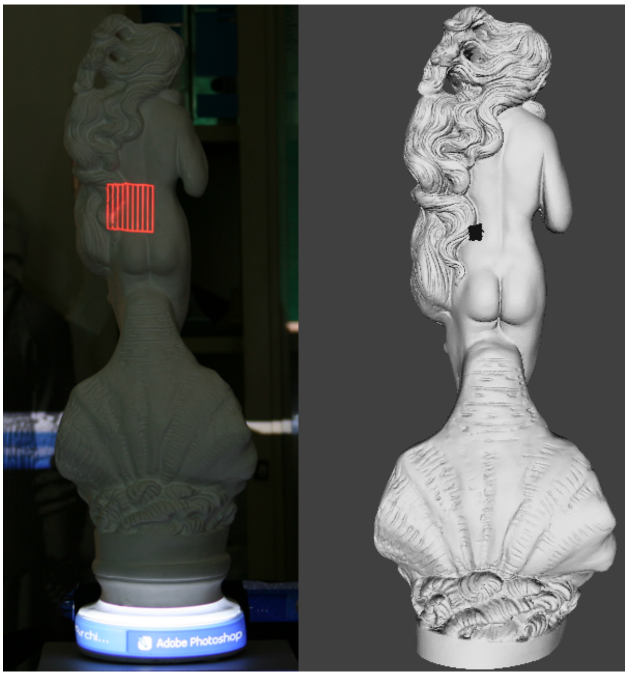


Figure 2.23: *An example of model with hole and image with projected pattern*

The user is asked to manually indicate the position of the four corners of the pattern in order to have a very accurate projector estimation.

2. **Estimation of pattern projector position.** During this important phase (which would not be needed in the case of a projector which is fixed to the digital camera) the position of the pattern projector is estimated starting from the knowledge of the shape of the projected pattern and the information taken from the extracted pattern on the registered image.

3. **Geometry reconstruction.** Once that camera and projector positions are known, it is possible to calculate the corresponding 3D position of each pixel in the image. This is obtained with the usual

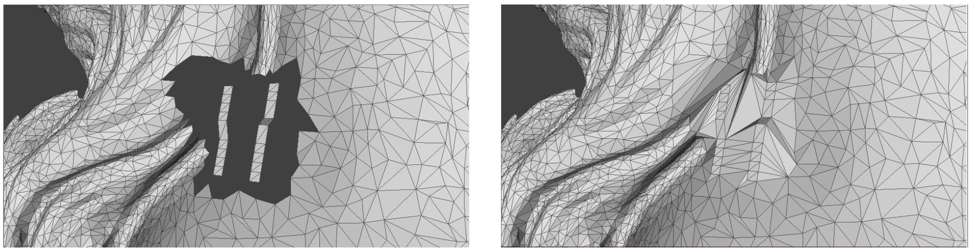


Figure 2.24: *the triangle strips generated using the image (Left); the rough hole filling (Right)*

technique of the triangulation. A 3D point is generated for each pixel of the extracted pattern which does not already map on the geometry. The produced points are used to create a series of strips (Figure 2.24 left) which split the original hole in a series of smaller holes. The information used for creating the strips is extracted from the images, and it is a quite exact representation of the real geometry of the object. The remaining steps of the hole filling procedure are the usual ones: a rough hole filling is applied to the mesh, then a refinement step brings the size of the newly created triangles to the medium size of the triangles of the original mesh.

Finally, a smoothing step (which involves also some of the geometry of the border of the original mesh) produces a final surface which is nearly indistinguishable respect to the rest of the model.

This hole filling technique can be applied in Cultural Heritage where, in most cases, a complete scanning of objects is practically impossible.

Chapter 3

Mutual Correspondences

This Chapter will summarize the main features of the three classic approaches to the alignment problem: correspondences based methods (Section 3.2), feature based methods (Section 3.3) and MI based methods (Section 3.4). Pro and cons will be emphasized along with a deep analysis of tools that use them (section 3.2 and 3.4). Starting from the classic approach, a new one will be introduced (section 3.5). It is called *Mutual Correspondences based method*.

3.1 General issues in image registration

Though the previously mentioned 2D-3D registration methods (Chapter 2) can achieve good results, they all present some weak points. Moreover, the image registration problem has some general issues. Hence the creation of robust, reliable and comparable methods is quite difficult.

The general issues of image registration are shown below. They will be considered both in the tool creation (Chapter 4) and in the tests results (Chapter 5) sections:

- *Finding a reliable quality measure of a 2D-3D alignment*

  Image registration problem handles two different types of data: 2D

and 3D points. Their intrinsically different nature makes difficult any comparison between them. Though suitable for practical cases, all the measures proposed for 2D-3D alignment lack in something.
Minimizing the distance between the 2D point of the correspondence and the 2D projection of the related 3D point could be somewhat misleading. In this case, we are dealing with a value which is only implicitly related to 2D and 3D points. So, it cannot be considered a proper comparison measure. Moreover, it strongly depends on how accurately the user sets and arranges the correspondences.
On the other side, MI is a statistical measure connected to the geometry only because illumination related geometric properties maps are considered. It is essentially a similarity measure between two images.

Nevertheless it could be possible to quantify the goodness of a camera calibration. For example, setting up a camera in a lab with all the parameters known in advance, and comparing the estimated camera with the real one. Though reliable, this procedure cannot be always accomplished and requires a suited environment. Practically speaking, only the *visual feedback* could be considered a *good quality measure*, which represents a problem for a more automatic approach and the comparison between methods.

- *Focal estimation*
  An incorrect starting approximation of the focal length of a camera could affect the entire optimization algorithm: finding an appropriate initial focal is still an open problem. The convergence of the focal is hard because its values fall in a range which is usually smaller (20-70 mm) than the other parameters range. For example, the image is almost always taken at a distance of 500-1500 mm from the object (if not more). To help convergence, the optimization algorithm proposed in

[MCS09] estimates the camera parameters (extrinsics and focal) adding offsets to the current values. Nevertheless the focal still affects the function to optimize possibly producing some local minima, where the algorithm could stop. Another way to solve this issue is the use of Exif metadata, which are nowadays provided by all digital cameras. The focal length value that can be found in the Exif is usually accurate enough to be a good starting point for accurate final alignments.

Aside from these general problems, each method of the classes mentioned in Chapter 2 has its pros and cons. In order to find an alternative approach, a careful analysis of them has been made.

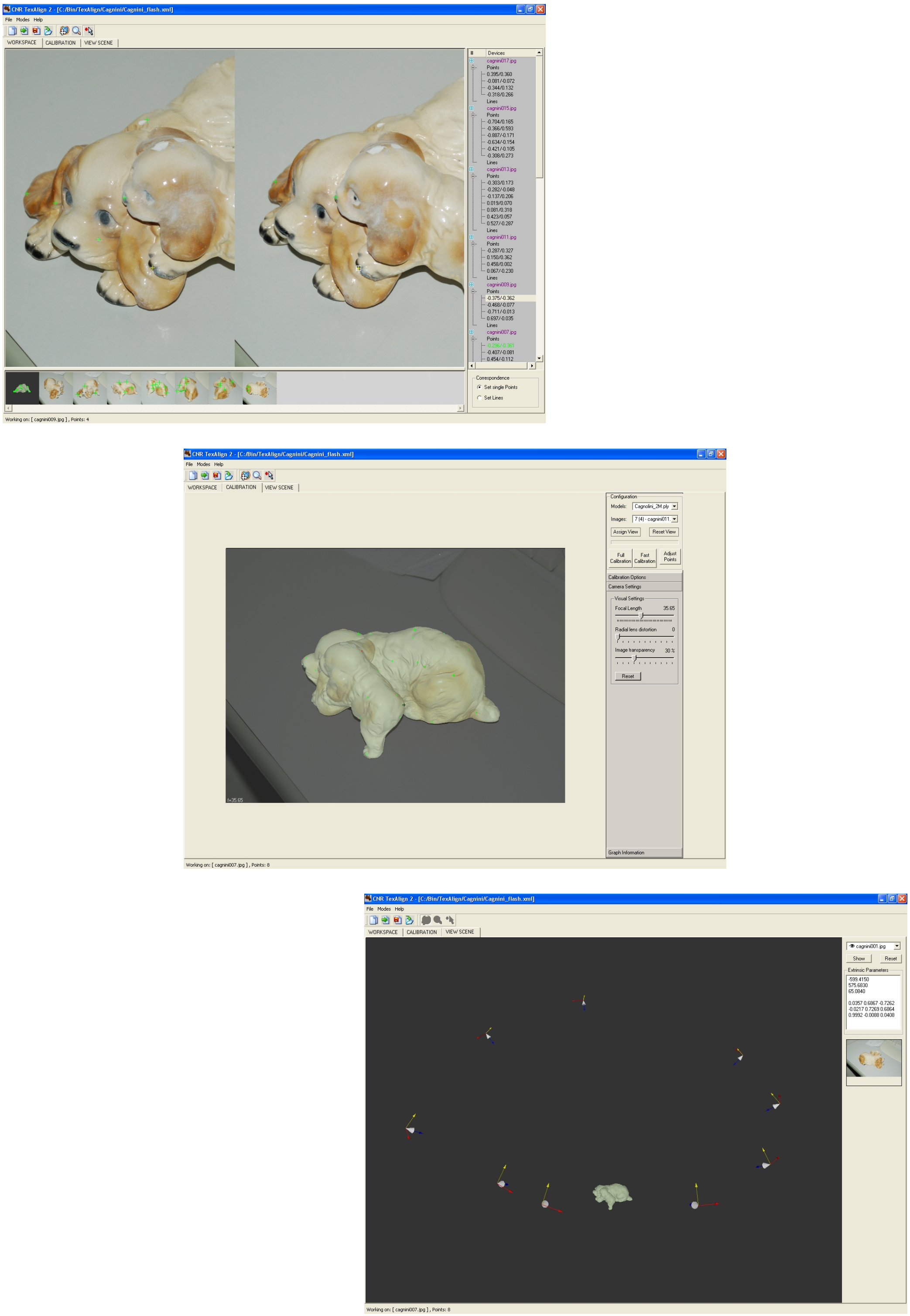
3.2 Correspondences based methods analysis: The TexAlign Tool

Correspondences based methods (Section 2.2) require users to find a lot of correspondences between the 3D model and the 2D image. Some tools have been proposed (for example TexAlign [FDG+05, FCG+05]) but the usability depends tightly from the application case.

TexAlign, a tool provided by Visual Computing Lab of ISTI-CNR, has been used in a wide variety of possible objects (from very small to very large ones). The registration process is organized as a work project, so that tens of images can be handled contemporaneously, and the alignment process can be saved and resumed at any time.

The interface of TexAlign is divided in three spaces:

- the *WorkSpace Tab* (Figure 3.1, top) contains all the elements of the registration project (visualized as thumbnails in the lower part of the screen).



cagnini017.jpg
Points
0.395/0.360
-0.081/-0.072
-0.344/0.132
-0.318/0.266
Lines
cagnini015.jpg
Points
-0.704/0.165
-0.366/0.593
-0.887/-0.171
-0.634/-0.154
-0.421/-0.105
-0.308/0.273
Lines
cagnini013.jpg
Points
-0.303/0.173
-0.282/-0.048
-0.137/0.206
0.019/0.070
0.081/0.318
0.423/0.057
0.527/-0.287
Lines
cagnini011.jpg
Points
-0.287/0.327
0.150/0.362
0.458/0.002
0.067/-0.230
Lines
cagnini009.jpg
Points
-0.375/-0.362
-0.468/-0.077
-0.711/-0.013
0.697/-0.035
Lines
cagnini007.jpg
Points
-0.296/-0.361
-0.407/-0.081
0.454/-0.112

● Set single Points
○ Set Lines

Working on: [cagnini009.jpg], Points: 4

Configuration
Models: Cagnolini_2M.ply
Images: 7 (4) - cagnini011.
Assign View | Reset View
Full Calibration | Fast Calibration | Adjust Points

Visual Settings
Focal Length 35.65
Radial lens distortion 0
Image transparency 30 %
Reset

Working on: [cagnini007.jpg], Points: 8

cagnini001.jpg
Show | Reset
Extinsic Parameters
-599.4150
575.6830
65.0840

0.0357 0.6867 -0.7262
-0.0217 0.7269 0.6864
0.9992 -0.0088 0.0408

Working on: [cagnini007.jpg], Points: 8

Figure 3.1: *The three spaces of TexAlign: top, Workspace Tab; middle, Calibration Tab; bottom, Scene visualization Tab.*

In this space, the user can set correspondences between all the elements of the project: not only between the 3D model and any image, but also between images. The data structure used to keep track of all the relations between the elements of the registration process is the *correspondence graph* (Figure 3.2), developed specifically for the tool.

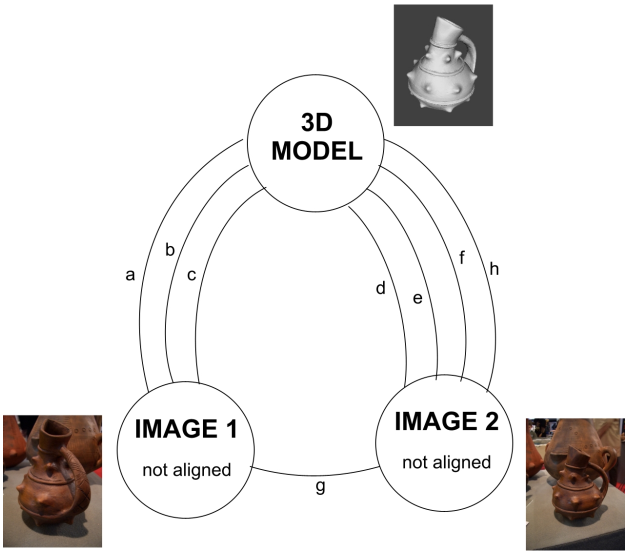


Figure 3.2: *A simple example of correspondence graph.*

- the *Calibration Tab* (Figure 3.1, middle) allows to launch the alignment for any image which has enough correspondences set to the 3D model. The estimation of camera parameters can be calculated using two methods: the classic Tsai one [Tsa87], which needs at least 12 correspondences, and a non-linear method [DG97] derived from the approach of Faugeras and Toscani [FT86], which needs at least 6 correspondences and performs optimization on extrinsic parameters and the focal length value. The second approach needs less correspondences, but the final results is influenced by the initial position of the model respect to the image: hence, a preliminary rough alignment is needed.

Once that the user is satisfied with the alignment, the associated view can be assigned to the image.

- the *Scene View Tab* (Figure 3.1,bottom) shows the model and the camera positions associated to each registered image.

TexAlign is a user-friendly tool that offers a very reliable and robust procedure. If the model presents some remarkable geometrical or color/-lighting related features, it could be very simple for the user to set the correspondences. Moreover, the initial position of the model is only slightly influential.Nevertheless, correspondences setting can be a tricky process that requires time and accuracy. For example, Tsai method needs at least 12 and they could be very difficult to detect.
In order to obtain accurate correspondences, it is necessary to:

- Be meticulous on setting the correspondences. The more the correspondences are precise, the more the registration is accurate.
- Distribute the correspondences selectively. They should not be concentrated in the same place, and should cover the entire model.
- Prefer sharp features.

Like TexAlign, all the tools of this group consider only the correspondences, without being aware of the complete geometry of the model. This could have some advantages when the mesh is not so accurate (for example due to some errors during the 3D scanning) because it forces the algorithm to be stuck on some exact points. Any other information is not considered. But in general correspondences setting can be hard, and one wrong correspondence can influence the final result. Hence the alignment process is usually iterative. The user has to:

1. set some correspondences;

2. launch the calibrator;

3. add/correct the correspondences until the calibration is satisfying

As already stated, the analysis of the result is mainly based on visual comparison.

3.3 Feature based methods analysis

Feature based methods (Section 2.3) have been introduced to contain user intervention as much as possible. Correspondences based methods strongly rely on the precision and distribution of 2D-3D points pairs making them user dependent.

Feature based methods essentially use image processing algorithms that compare two images according to a similarity measure. Typically, they are implemented in two steps:

1. searching for specific features in both images;

2. finding a matching between them according to these features.

This makes feature based methods completely automatic. Image processing algorithms are very fast, and scalable, which is desirable and advantageous. The accuracy is not user dependent, and a fine alignment needs only the starting point, precise geometry and clear features. Unfortunately not every feature based methods are general-purpose. There are constraints that, often, make impossible their use in some applications (i.e. Cultural Heritage). Some methods need pre-processing steps (reflectance estimation, or de-contouring) that could affect performances. Other methods starts from the assumption of having the, same illumination conditions, an uniform background or the presence of the entire object inside each photos. At these conditions, a well-equipped (and often expensive) environment is mandatory.

3.4 Mutual Information based methods analysis

Mutual Information based methods (Section 2.4) need very little interaction with the user and introduced a lot of advantages.
First of all, as stated in Section 2.4, by exploiting some illumination-related geometric properties, it is possible to create much more meaningful renderings for the MI than the photorealistic ones. GPU implementation of the algorithm proposed could be developed as well.

Compared to the error evaluation methods (section 2.2.2), MI is a less intuitive and immediate alignment measure (though this term can be a little misleading). It is only a statistical quantity to be maximized that has no tangible correspondence. To have a concrete idea of the sense of this measure, it is necessary to draw some graphs as in Figures 2.9, 2.10, 2.11, 2.12.

For test validation, the paper proposed by [MCS09] uses an user-friendly tool for the alignment through MI . Its interface presents a window where a 3D rendering is placed on top of a 2D image. A slider controls the level of transparency so that the user can better check the state of the alignment. User can also choose if inserting the focal among the parameters to estimate. Different kinds of renderings are proposed (Figure 3.3):

- Normal + Ambient
- Normal
- Ambient
- Specular
- Silhouette
- Specular + Ambient

Ambient Occlusion is pre-calculated and stored as color, the others implemented with shaders.

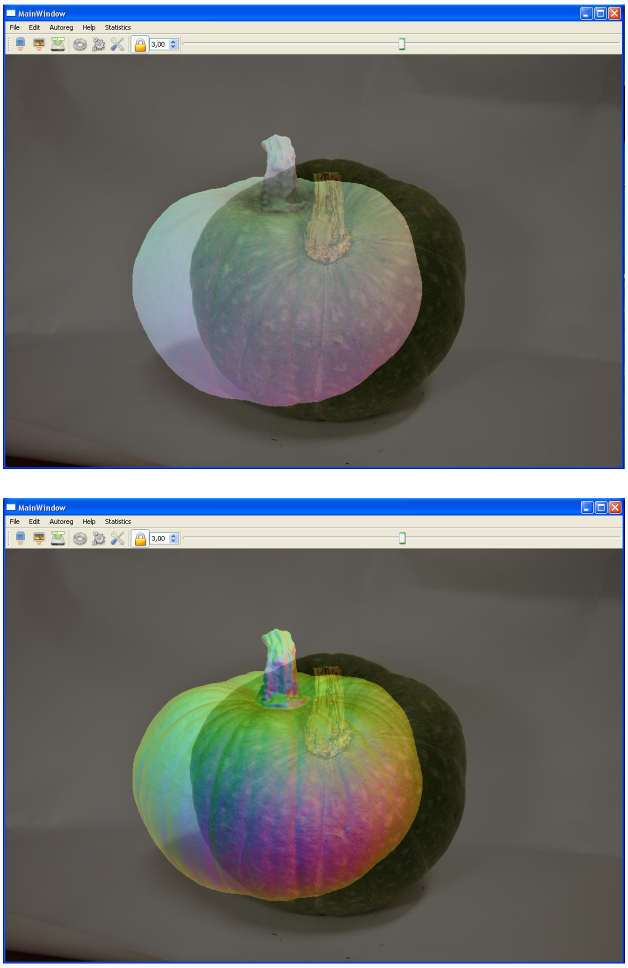

MainWindow
File Edit Autoreg Help Statistics
3,00
MainWindow
File Edit Autoreg Help Statistics
3,00

Figure 3.3: *2 renderings of the 3D model in the tool used in paper [MCS09]: top, Normal + Ambient; bottom, Specular.*

The optimization problem formalized by the Formula 2.9 needs to be solved in order to to estimate the camera parameters. The tool provides two implementations of the fine alignment described in section 2.4. The *Optimize* mode uses the open source library NEWUOA [Pow04]. For each iteration it computes, first, the offsets to be added to the current estimated parameters, and then the MI between the two images.

NEWUOA is a software developed by M.J.D. Powell for unconstrained optimization without derivatives [?]. The NEWUOA seeks the least value of a function F(x) (x is a vector of dimension n) when F(x) can be calculated for any vector of variables x . The algorithm is iterative; a quadratic model is required at the beginning of each iteration, which is used in a trust region procedure for adjusting the variables. When the quadratic model is revised, the new model interpolates F at m points, the value m=2n+1 being recommended.

In the tool referred in paper [MCS09] user can set the following variables:

- *Tolerance*: it states the accuracy of the final alignment;
- *Expected Variance*: it states how much the initial alignment is incorrect;
- *Background weight*: it weights the relevance of the silhouette.

The *Optimize Full* mode calls the *Optimize* function n times (with n set by the user) changing the value of tolerance, variance, n and other variables, to let NEWUOA explore a much wider trust region. No information has to be supplied other than the initial position and (if available) the focal length, that should be as much accurate as possible. If both are very far from the correct values, Mutual Information could achieve incorrect results in terms of the quality of the alignment and efficiency. Nevertheless, having reasonably precise initial position or focal length values reduces the possibility to stop in a local minimum and speeds up convergence time. The performance are extremely interesting, and the alignment of a single image can be found in less then a minute.

The typical usage of the tool expects the user to:

1. find a good initial position;

2. call the *Optimize Full* function without estimating the focal;
3. call the *Optimize* function to refine the alignment, this time estimating the focal too.

Due to the essentially statistical nature of Mutual Information, its correlation to the geometry is only implicit through the illumination-related geometric properties used in the renderings. So, in case of inaccurate 3D models, it shows some restrictions.

This tool doesn't offer enough control to the user. If the algorithm falls in a local minimum, there is no way to prevent the algorithm from stopping. Moving the initial position or changing the focal could not change the final result as well.

Tables 3.1, 3.2 and 3.3 summarize pros and cons of the different approaches.

Correspondences based methods

Pros	**Cons**
robust, reliable and efficient	takes time and accuracy
gives control to the user	semiautomatic
independent from the initial position	needs lots of precise correspondences, sometimes difficult to find
strongly and partially geometrical related and so useful with incorrect 3D model	independent from the whole 3D geometry
error gives an intuitive idea of the alignment	

Table 3.1: *A summary of the pro/cons of the correspondences based methods*

Feature based methods

Pros	**Cons**
automatic	no control by the user on the final result
precise	requires accurate geometry and evident features
fast	sometimes requires a pre-processing step (for example, de-contouring)
	strongly dependent from the initial position

Table 3.2: *A summary of the pro/cons of the correspondences based methods*

MI based methods

Pros	**Cons**
automatic	no control by the user on the final result
implementable in GPU	strongly dependent from the focal length and the initial position
implicit whole geometric correlation	implicit geometric correlation and errors with incorrect 3D model
fast and precise	sometimes difficult repeatability of the alignment
intuitive for the user	MI value doesn't give an intuitive idea of the alignment

Table 3.3: *A summary of the pro/cons of the MI based methods*

3.5 Mutual Correspondences: a new approach

Every discussed method has its pros and cons. The basic idea is to find a new approach that, keeping the reached goals, can deal with the still unresolved issues. It is important that the new approach could be used in different scenarios and should not carry overhead costs. Refining the previous techniques cannot meet this target, because each one presents limits related to its own nature. A much more interesting approach could be to try to combine some approaches taking their best. In Tables 3.1, 3.2 and 3.3 it is evident that each pros and cons are somehow complementary, and this boosts the idea of a combination of some of those.

Mutual Correspondences (MC) tries to merge the Correspondences based methods and the MI based methods. It starts from the observation that these two classes of methods face up to the alignment problem in two opposite ways. The first is a semiautomatic, geometric approach, the second an automatic, statistical one. The idea is to find a third equally reliable and robust approach that combines the degrees of freedom of the Correspondence based methods, the automatism and the whole geometric correlation of the MI based methods. Feature based methods haven't been considered because they are not so robust. Moreover, MI based methods clearly accomplish the same goals of feature based methods without requiring pre-processing phases or suited environment. Some features could be explored even with MI base methods (silhouette), but this time they are not extracted from an image. Feature information are directly encoded in an image, and used implicitly by the MI function to compare the original photo and the 3D rendering ([MCS09])(see Section 2.4).

The Mutual Correspondences error function is expressed by the weighted function 3.1:

$$MC_{(I_A,I_B,Corr,C)} = k(-MI(I_A, I_B(C)) + (1-k)E(Corr, C) \tag{3.1}$$

$$C = (\theta, \phi, \psi, t_x, t_y, t_z, f)$$

where

I_A	preprocessed image to align
I_B	the rendering of the 3D model that depends on the camera parameters C
$MI(I_A, I_B(C)$	MI correlation function 2.8
$E(Corr, C)$	correspondences function 2.7
θ, ϕ, ψ	Eulero angles
t_x, t_y, t_z	components of translation vector
f	focal
k	weight constant
$Corr$	correspondence set

The alignment problem is led back to a minimization problem of the function 3.1:

$$C^* = \arg\min_{C \in \mathbb{R}} MC_{(I_A, I_B, Corr, C)}) \tag{3.2}$$

$$C = (\theta, \phi, \psi, t_x, t_y, t_z, f)$$

This function could seem a quite odd because two different quantities are involved. On the one hand there is a statistical, dimensionless quantity; on the other hand a measure of length (in pixel). Their nature is very different and adding them up is mathematically weird. Nevertheless, they indirectly share the same target: quantifying the similarity between two images. Tests results will show that this intuition works (Chapter 5). Maybe, the absence of a reliable quality measure in the image to geometry alignment process can explain this apparent mismatch.

The optimization problem presented in Section 2.4 is refined with the addition of the error estimation. In parallel with the computation of the

Mutual Information, the correspondences error between the 2D points and the projections of the correspondent 3D points is calculated. The function that drives the alignment now is the 3.1. k, a constant in the range 0-1, allows the user to weight each sub-function. The k value makes possible to customize the technique according to different scenarios (for an analysis of the effects of different values of k, see Chapter 5). A synergistic and smart use of both correspondences and MI allow to cover different cases.

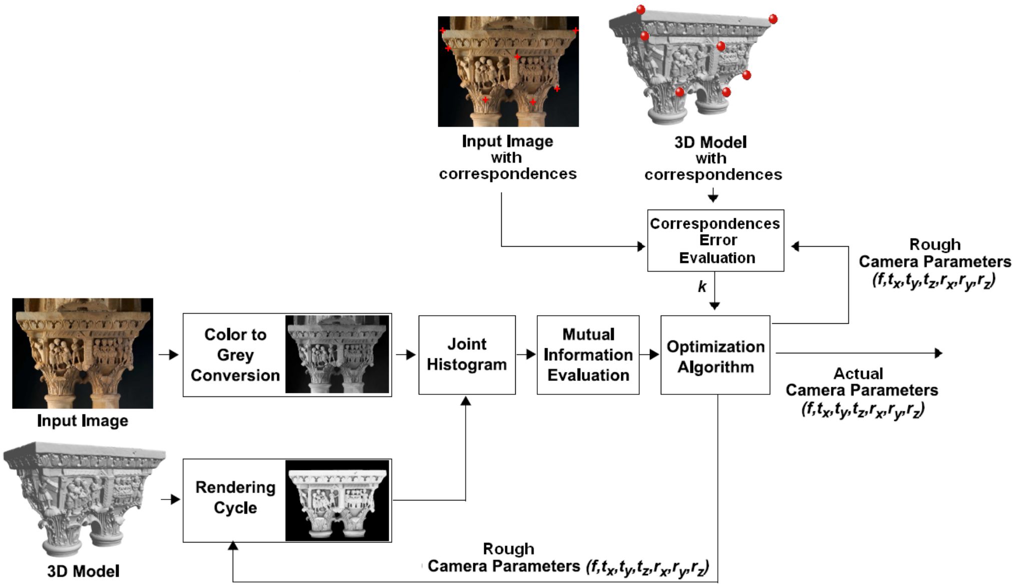

Input Image with correspondences
3D Model with correspondences
Correspondences Error Evaluation
Rough Camera Parameters $(f,t_x,t_y,t_z,r_x,r_y,r_z)$
k
Input Image
Color to Grey Conversion
Joint Histogram
Mutual Information Evaluation
Optimization Algorithm
Actual Camera Parameters $(f,t_x,t_y,t_z,r_x,r_y,r_z)$
3D Model
Rendering Cycle
Rough Camera Parameters $(f,t_x,t_y,t_z,r_x,r_y,r_z)$

Figure 3.4: *A sketch of the MC registration algorithm*

Though very different, their usage is complementary. While correspondences based methods are flexible and give the user a significant control over the final alignment, MI based method are automatic and fast. The combination of the error and MI functions lets the user have a robust, flexible and automatic method adding, at the same time, a further degree of control through k.

Along with the usage, their flaws are complementary too. A joint use of the two techniques, automatically solves some of their weak points. Correspondences clearly drive and force convergence, avoiding local minima that MI can't skip; at the same time, MI reduces considerably the number of correspondences the user must set. So, the amount of time and accuracy required to the user is smaller. Moreover, through MI, it is possible to connect, though implicitly, with the whole 3D model.

Moreover, correspondences based methods can converge even with initial incorrect values of the focal and starting position. This feature has been preserved in the MC function due to its own nature. If the model is very far from the alignment position, the error calculated through the correspondences is much bigger than the Mutual Information. So, apart from the value of k, it will always weight more in the overall MC function. Only when the correspondences error and the MI reach comparable values, k becomes meaningful. Generally, it is much easier to find a good initial position manually than finding some very precise correspondences.

In conclusion, it is possible to find two distinct moments in the convergence process::

- *a rough alignment*: the error is much bigger than the MI, so the latter is negligible; MC will soon converge towards the neighbourhood of the minimum.
- *a fine alignment*: when the value of the error and the MI are approximately the same, k weights the two sub-functions.

The typical usage of the method expects the user to:

1. manually find a satisfying initial position and call the function with k=1 (that corresponds to a fully MI based method) for an early alignment;
2. if necessary, refine the alignment by setting an appropriate number of correspondences around the non correctly aligned areas; k must be set

as well.

A robust implementation of the MC based method is provided by the **TexAlign Suite** (Chapter 4).

Chapter 4

TexAlign Suite

This Section will present the new **TexAlign Suite**, a framework developed with the aim to put together different approaches to the image to geometry alignment problem. It is written in C++ and uses only open source libraries as Vcg Lib, OpenGl, Qt, Tsai, Levmar e NEWUOA. It is developed for Windows and Unix platforms.

The Section will describe the libraries used (Section 4.1.1), the user interface (Section 4.2.1), the project structure (Section 4.2.2), the calibration modes (Section 4.2.3). The last Section (4.3) contains a practical user guide.

4.1 Overview

Each of the methods discussed so far depends on a variety of factor such as lighting conditions, goodness of the 3D scanning, geometry of the model, local minima, features, accuracy and background. In different scenarios it could be easier using one in place of another. So **TexAlign Suite** has been developed on purpose. It reunites all the three approaches (involving error function, MI and MC) in only one robust tool allowing to use the best for every specific case.

TexAlign Suite has been widely influenced by the TexAlign [FDG+05] and the tool for MI registration developed both by Visual Computing Lab of ISTI-

CNR [MCS09]. TexAlign Suite takes a 3D model and one image as an input. The visual output is a fine 2D-3D alignment according to proper camera parameters (extrinsics and intrisics). The registration process is organized as a work project, so that the alignment can be saved and resumed at any time.

4.1.1 Libraries

VcgLib

Vcg Lib is a C++ templatated library for the manipulation, processing of triangle and tetrahedral meshes. The library, released under the GPL license, is the result of the collaborative efforts of the Visual Computing Lab of the ISTI - an institute of the Italian National Research Council (CNR).
Vcg Lib provides support for creation and managements of components, data structures algorithms. Renderings and GUI design are not supported. It is up to the application that uses Vcg Lib to deal with all the other aspects and the interaction with the user.

TexAlign Suite uses Vcg Lib for the creation and management of the mesh. The tracking and camera model is implemented and managed using Vcg Lib data structures and methods.

OpenGl

OpenGL (Open Graphics Library) is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. In TexAlign Suite it is used to rendering the mesh through shaders.

Qt

Qt is a cross-platform application development framework, widely used for the development of GUI programs (in which case it is known as a widget toolkit), and also used for developing non-GUI programs such as console tools and servers. Qt is object oriented and uses standard C++, but makes extensive use of the C pre-processor to enrich the language.

TextAlign uses Qt for the GUI design and development. It turns out to be very helpful for the possibility to graphically design some components and for the signal and slots construct. Controls (or widgets) can send signals containing event information which can be received by other controls using special functions known as slots.

Others

For the analysis of the Tsai library, Levmar and NEWUOA software packages see Section 4.2.3.

4.2 TexAlign Architecture

TexAlign Suite is a C++ framework that puts together different approaches to the image to geometry alignment problem. It is developed for Windows and Unix platforms using only open source libraries.
It has three main modules:

- user interface;
- projects management (in/out);
- the calibration modes.

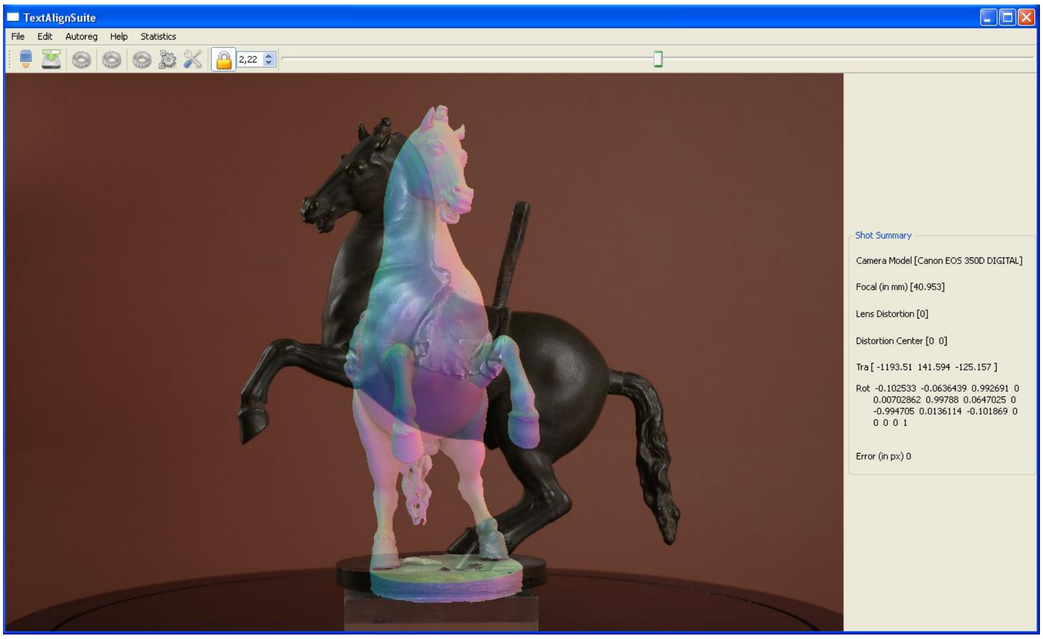
4.2.1 User Interface

TexAlign Suite presents one principal space, a working area, where a rendering of a 3D model of an object and a photo are overlapped. The photo shows the 3D model from a particular viewpoint. User can easily move the 3D model (a triangular mesh) with the help of a trackball. On the right side of the working area there is a window containing a camera summary. It visualizes the current parameters and other info:

- Camera model: if present, it is read from the Exif file;
- Focal length: in mm; initially read from the project file or from the Exif file, if present;
- Lens distortion: in case of radial distortion;
- Center of distortion: in case of radial distortion;
- Tra: translation vector;
- Rot: rotation matrix;
- Error: in px. Meaningful only if correspondences are set; it is the mean distance between the 2d image points and the correspondent 3D points projected on the image plane, according to the formula 2.7.

An essential menu and a toolbar offer the user all the functionalities he needs to use the three calibration modes:

- Correspondences modes (Tsai and Levmar);
- MI mode;
- MC mode.



Shot Summary

Camera Model [Canon EOS 350D DIGITAL]

Focal (in mm) [40.953]

Lens Distortion [0]

Distortion Center [0 0]

Tra [-1193.51 141.594 -125.157]

```
Rot  -0.102533  -0.0636439  0.992691  0
     0.00702862  0.99788  0.0647025  0
     -0.994705  0.0136114  -0.101869  0
     0  0  0  1
```

Error (in px) 0

Figure 4.1: *A screenshot of the TexAlignSuite*

The Toolbar

TexAlign Suite toolbar offers some useful shortcuts and functionalities. The shortcuts are related to input and output operations (save/load project), calibration modes (Tsai, Levmar, MI, MC) and preferences. An icon representing a padlock allow the user to *lock/unlock the focal*. This is fundamental because each method can include the focal length in the list of the parameters to be estimated. In the MI mode sometimes could be useful keeping the focal locked (for example if the starting point is very far from the aligned position) to prevent a bad estimation. This could affect the entire convergence process.

Sometimes the focal length is unknown and not present in the Exif file. With a spinbox the user can manually change it. In some cases, when a significant offset needs to be added to the current value, forcing the focal is

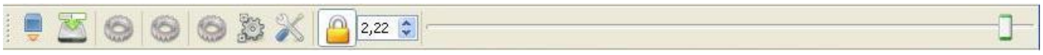


Figure 4.2: *A screenshot of the toolbar*

the only way to have significant improvements.

A slider has been introduced to control the transparency of the image. This is very helpful to check the alignment. As stated in Section 3.1 visual feedback is the only reliable quality measure. Hence, overlapping the model and the image and changing image alpha, helps to highlight some details in the alignment and to state where refine it (i.e. setting more correspondences).

4.2.2 The Project

The TexAlign Suite project is saved in an XML file. His structure is very simple and intuitive. It is arranged as follow:

- information about the 3D model (path, id, number of correspondences attached);
- information about the image (path, id, number of correspondences attached);
- camera parameters (rotation matrix, translation vector, viewport dimensions, focal length, scale correction and others);
- list of correspondences as a pair of 2D-3D points, with the id of their related device (model or image).

Menu items and shortcuts on the toolbar let the user save and load his projects.

4.2.3 The Calibration Modes

There are four different calibration methods: Tsai mode, Levmar mode, MI mode, MC mode. They are implementations of the correspondences, MI and MC based methods, respectively, using different libraries. They can be launched from items in the Calibration menu or from toolbar shortcuts.

In each mode there is the possibility to read the focal length from the Exif file attached to the image (if available). The Exchangeable image file format (Exif) is a specification for the image file format used by digital cameras. The specification uses the existing JPEG, TIFF Rev. 6.0, and RIFF WAV file formats, with the addition of specific metadata tags.
TextAlign Suite has a small but effective C++ library that returns each tag in the Exif file. If the focal length is not stored in the project file, the tool automatically tries to load it from the Exif file. This value is supplied by the camera manufacturer, hence it is reliable.

Moreover, in the Options window, user can also load the CCD width. It is useful to calculate the pixel width. Having the correct focal length and pixel width makes the entire registration process easier and faster.

The Tsai mode

The Tsai mode implements a correspondences based method (Section 2.2). It relies on the Tsai library [Tsa95] a software package in C that contains routines for calibrating Roger Tsai's perspective projection camera model (Section 2.1). It is open source and can be downloaded for free. Routines are provided for three levels of calibration. The first level is a direct implementation of Tsai's algorithm in which only the extrinsic parameters are estimate. In the second one, 3 (focal, T_z, and k_1) of the 11 calculated model parameters are numerically optimized. The third level is an extension of Tsai's algorithm in which the full 11 parameter model is numerically optimized.

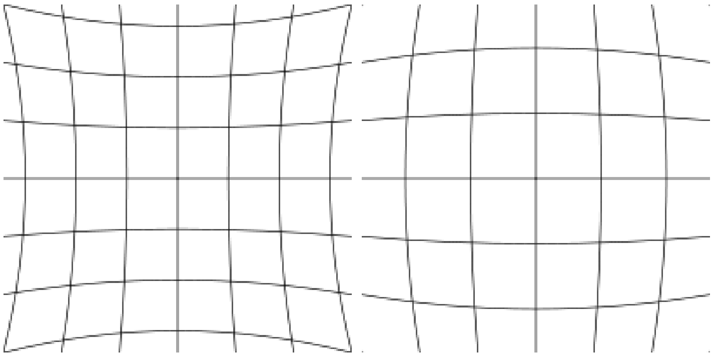


Figure 4.3: *An example of pincushion distortion (left) and barrel distortion (right).*

Full optimization is slower but provides the most accurate calibration. Tsai mode uses the first and the third level of calibration depending on the status of the focal length icon. If it is blocked, only the extrinsic parameters are calibrated; otherwise, all the 11 parameters are calculated and optimized. It needs at least 12 correspondences to work.
Among the values the Tsai library calibrates, there is the radial distortion parameter k_1. The sign of k_1 identifies two different types of radial distortion:

- $k_1 < 0 \Rightarrow$ **pincushion distortion**: image magnification increases with the distance from the optical axis. The visible effect is that lines that do not go through the center of the image are bowed inwards, towards the center of the image (Figure 4.3 left).

- $k_1 > 0 \Rightarrow$ **barrel distortion**: image magnification decreases with distance from the optical axis. The apparent effect is that of an image which has been mapped around a sphere (Figure 4.3 right).

- $k_1 = 0 \Rightarrow$ **no distortion**.

TexAlign Suite supports a functionality that undistort the original image, after estimating k_1. To properly interpolate the pixels, the Push-Pull algorithm [DCOY03] is used, avoiding discontinuity in the image.

The Levmar mode

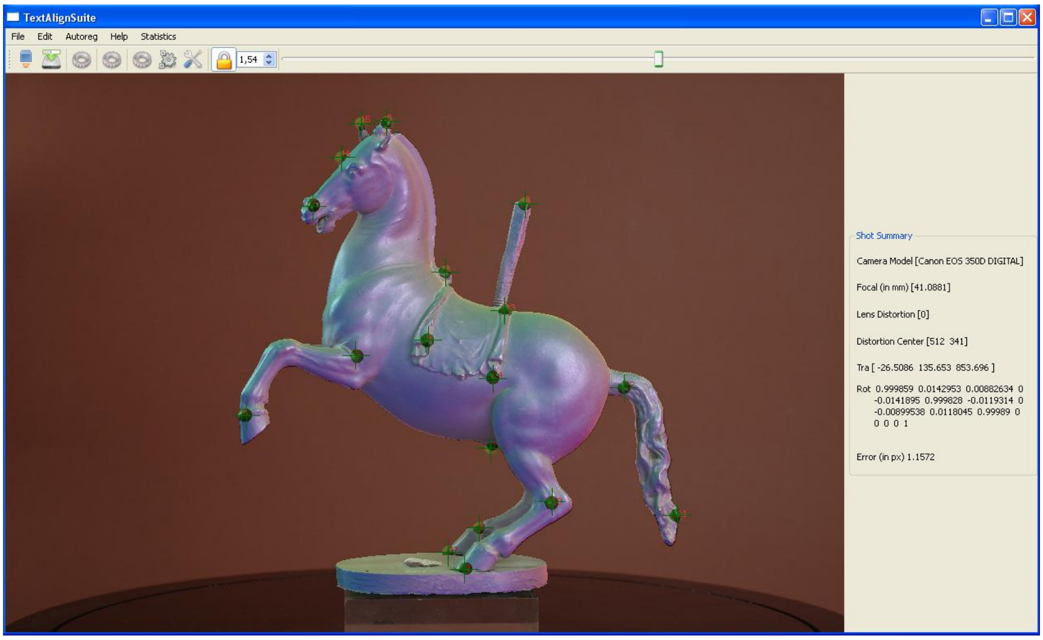
The Levmar mode is an implementation of a correspondences error based method (Section 2.2.2). Starting from a number of correspondences (it needs at least 3 or 4 to estimate focal length too) it computes the mean distance between the 2D image points and the correspondent 3D points projected on the image plane, according to the equation 2.7. This equation is formalized as a sum of squares of m nonlinear functions with $m = 2{*}\textit{total correspondences}$ (considering two distinct functions for x and y). Minimizing this function brings to a fine alignment.

The chosen minimization library is Levmar [Lou04], written in C/C++. The current version (2.5) is open source and available for a free download. The Levmar software package is an implementations of the Levenberg-Marquardt optimization algorithm (LM). It is an iterative technique that finds a local minimum of a function f: $R^n \to R^m$ that is expressed as the sum of squares of m nonlinear functions of n parameters with $m \geq n$. The formula 2.7 is expressed this way. Constraint $m \geq n$ imposes at least 3 (or 4) correspondences.

Levenberg-Marquardt has become a standard technique for nonlinear least-squares problems. It can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method which is slow, but guarantees to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method.

The Gauss-Newton method is based on a linear approximation to the components of a function $f(x)$ (a linear model of $f(x)$) in the neighbourhood of x. Function 2.7 can be linearly approximated. Due to the nature of the function, the Levenberg-Marquardt optimization algorithm (LM) appears as a sensible choice.

Focal length estimation is performed trough the padlock icon, just like in the Tsai mode.



Shot Summary

Camera Model [Canon EOS 350D DIGITAL]

Focal (in mm) [41.0881]

Lens Distortion [0]

Distortion Center [512 341]

Tra [-26.5086 135.653 853.696]

```
Rot 0.999859 0.0142953 0.00882634 0
    -0.0141895 0.999828 -0.0119314 0
    -0.00899538 0.0118045 0.99989 0
    0 0 0 1
```

Error (in px) 1.1572

Figure 4.4: *A screenshot of the Tsai/Levmar mode*

Levmar and Tsai mode share comparable performance, but the first one needs fewer correspondences. Like the Tsai mode, the convergence is guaranteed even for very far initial positions from the alignment position, although Levmar can be influenced by the initial position. Pre-processing (involving normal computation) could overcome this problem. However only the extrinsics and the focal length are estimated. Distortion parameters are discarded, making the undistortion of the original image impossible. If the final alignment is not satisfactory, user can set other correspondences, or use the current position as the initial position in the MI or MC mode.

The real issue of Levmar mode is estimating properly the focal length, if it starts from an incorrect value (Figure 4.5).

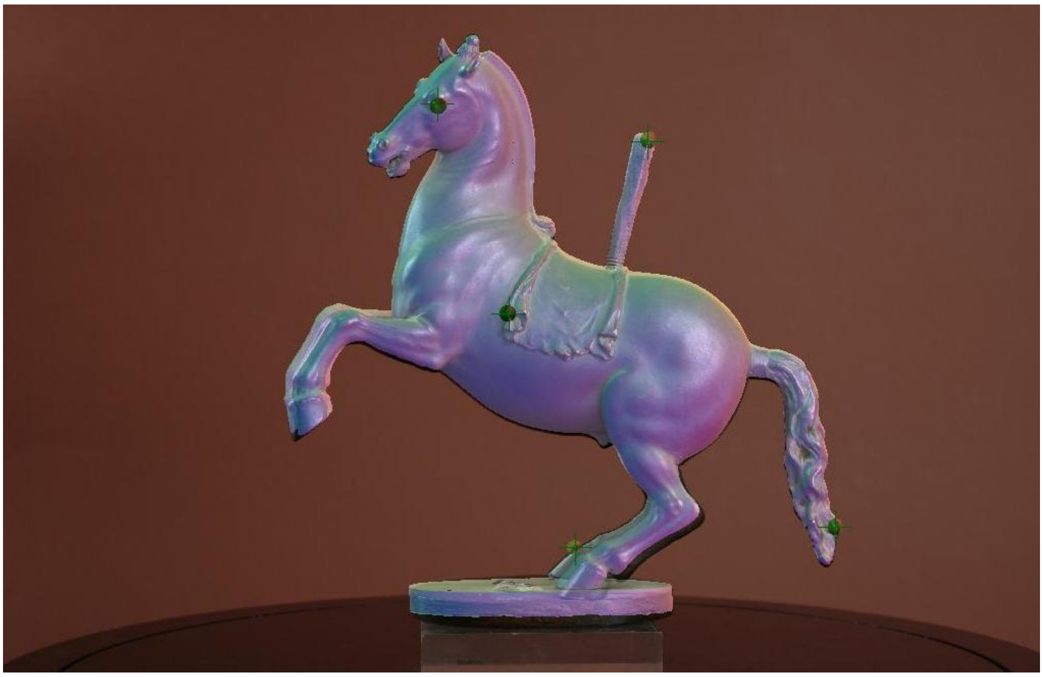
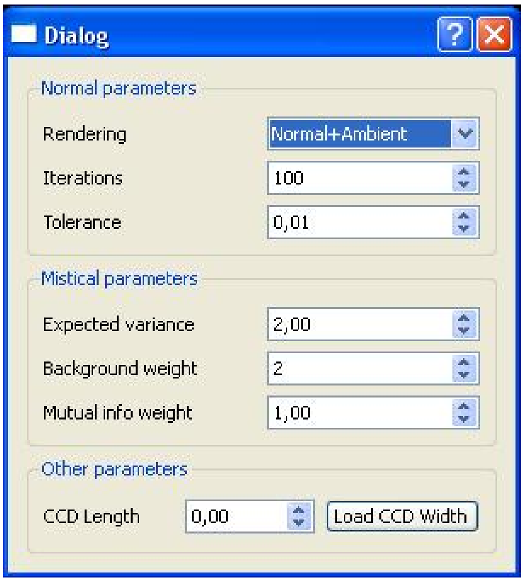


Figure 4.5: *An alignment in Levmar mode with an incorrect focal*

The MI mode

The MI mode is an implementation of a MI based method, using illumination-related geometric properties (Section 2.4). The available renderings use properties like ambient occlusion, surface normals, reflection, silhouette or a combination of some of them (normal and ambient, specular and ambient). Ambient Occlusion is pre-calculated and stored as color, the others are implemented with OpenGL shaders. The MI between one of the renderings and a grey-scale version of the original image is calculated. User can choose the rendering mode in the Options window (Figure 4.6).



Normal parameters

Rendering: Normal+Ambient

Iterations: 100

Tolerance: 0,01

Mistical parameters

Expected variance: 2,00

Background weight: 2

Mutual info weight: 1,00

Other parameters

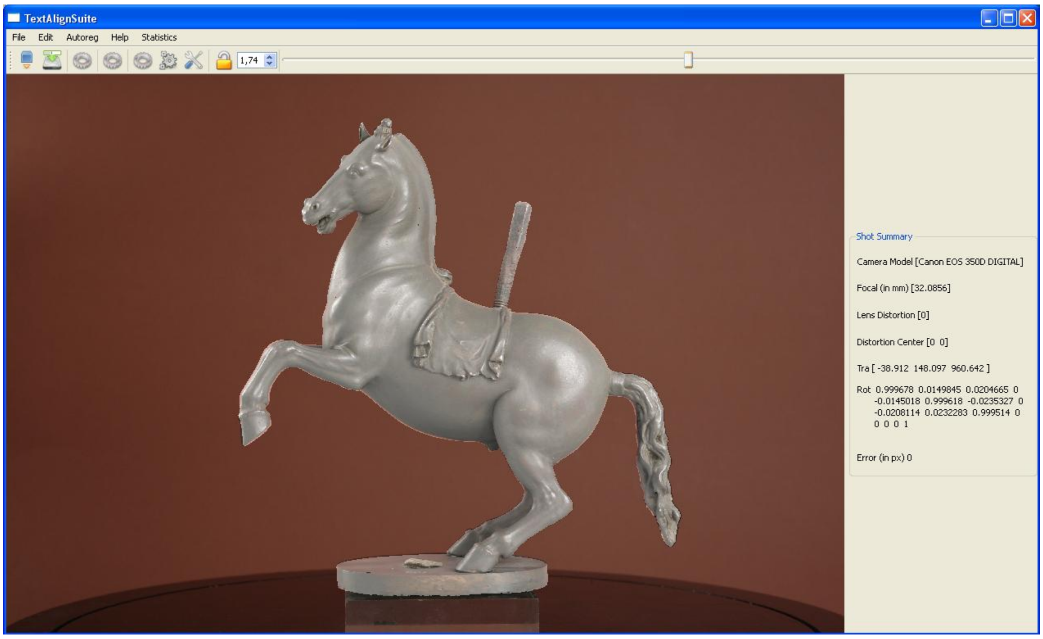
CCD Length: 0,00

Figure 4.6: *A screenshot of the Options window*

The camera estimation has been formalized as an optimization problem (Formula 2.9). The value calculated with Formula 2.8 needs to be minimized with an iterative algorithm.

The open source library NEWUOA([Pow04]) has been used. NEWUOA is a software developed by M.J.D. Powell for unconstrained optimization without derivatives. The NEWUOA seeks the least value of a function F(x) (x is a vector of dimension n) when F(x) can be calculated for any vector of variables x . The algorithm is iterative and at the beginning of each iteration F(x) is approximated with a quadratic model Q. It is used in a trust region procedure for adjusting the variables. When Q is revised, the new Q interpolates F(x) at m points, the value $m = 2n + 1$ being recommended, with n the number of parameters. Only one interpolation point is altered on each iteration.

The choice of NEWUOA is justified by the smoothness of the function to optimize. Algorithms based on estimating derivatives perform poorly on the



```
Shot Summary

Camera Model [Canon EOS 350D DIGITAL]

Focal (in mm) [32.0856]

Lens Distortion [0]

Distortion Center [0  0]

Tra [ -38.912  148.097  960.642 ]

Rot  0.999678  0.0149845  0.0204665  0
     -0.0145018  0.999618  -0.0235327  0
     -0.0208114  0.0232283  0.999514  0
     0  0  0  1

Error (in px) 0
```

Figure 4.7: *A screenshot of the MI mode (Ambient occlusion rendering used)*

non-smooth MI functions, especially near the minimum. Other algorithms, based on pattern search, require many iterations to converge, due to the number of dimensions of the problem. Most image registration techniques try to produce smooth MI function using techniques such as Parzen-window [Par62] or antialiasing rendering. Although MI minimization could be formalized as a nonlinear least-squares problem, NEWUOA is preferred to Levmar. Infact MI function cannot be linearly approximated and smoothing processes are very expensive.

In TexAlign Suite, the alignment can be performed in the *Optimize* mode, clicking on an icon on the toolbar or from the menu. Each iteration computes first the offsets to be added to the current estimated parameters, and then the MI between the two images. In the Option window user can set the maximum number of iterations (parameter *Iterations*), beyond which the

algorithm would stop even if it hasn't find a local minimum.
User can also set the following variables:

- *Tolerance*: it states the accuracy of the final alignment;
- *Expected Variance*: it states how much the initial alignment is incorrect;
- *Background weight*: it weights the relevance of the silhouette.

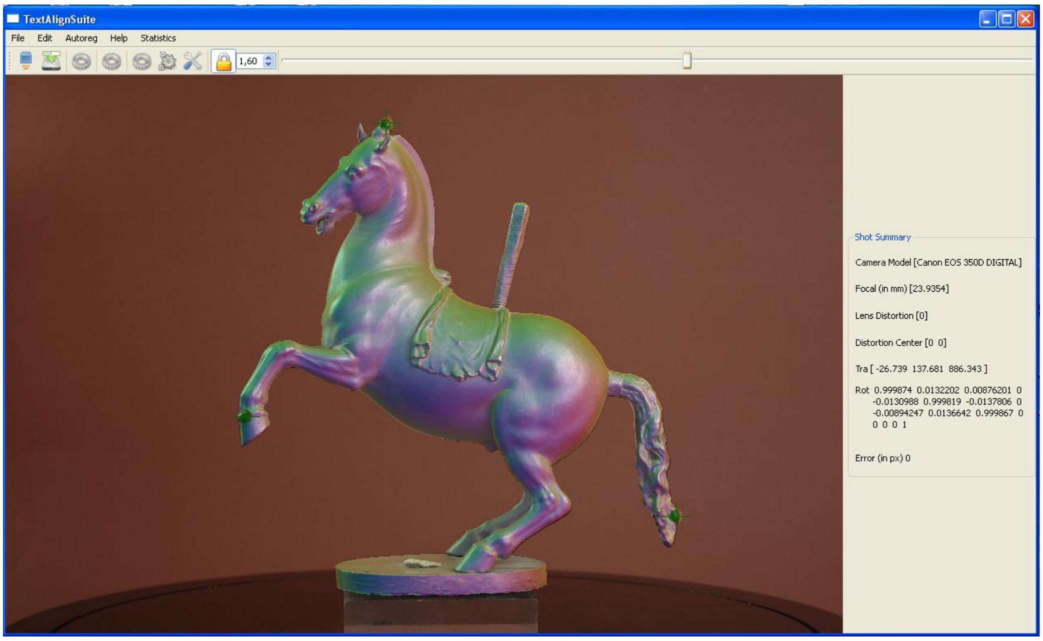
In order to perform MI based alignment, parameter Mutual Info weigth must be set to 1. The *Optimize Full* mode calls the *Optimize* function *Iterations* times changing the value of tolerance, variance, and other variables, to let NEWUOA explore a much wider trust region.

The MC mode

The MC mode is an implementation of the MC based method proposed in Section 3.5. Having properly designed the other two modes, its implementation is straightforward. The MC mode is perfectly integrated in the MI mode through the parameter *Mutual Info weigth* in the Options window. Its range is 0-1 and represents the k value in the related model (equation 3.1). MC offers the same functionalities of the MI mode plus the *Mutual Info weigth* that combines the computation of the MI and the correspondens error expressed by the formula 2.7. Setting *Mutual Info weigth* to 1 means switching to the MI based alignment and no correspondences are required; setting 0 means switching to the correspondences error based alignment, this time using the NEWUOA library. In this case at least one correspondence needs to be set. In-between values weight the two approaches.

NEWUOA library has been preferred to the Levmar library for a number of reasons. Setting *Mutual Info weigth* to 0 allow to have another implementation of the correspondences error alignment and could work with only one correspondence. Nevertheless, using NEWUOA to solve this non-linear least squares problem, even if possible, is not advantageous. NEWUOA is based on the interpolation of a quadratic model around m sampled points, that, though flexible, is quite expensive. Levmar, instead, is specifically developed for non-linear least squares problems; so it shows a better performance. As to minimizing MC function (with $k > 0$), it retains all the features of the MI function, including non being linearly approximated; so, using NEWUOA library is an immediate choice.

Since in TexAlign Suite MC mode is built upon the MI mode, it could be launched in the *Optimize mode* or *Optmize Full mode*.



Shot Summary

Camera Model [Canon EOS 350D DIGITAL]

Focal (in mm) [23.9354]

Lens Distortion [0]

Distortion Center [0 0]

Tra [-26.739 137.681 886.343]

```
Rot  0.999874  0.0132202  0.00876201  0
     -0.0130988  0.999819  -0.0137806  0
     -0.00894247  0.0136642  0.999867  0
     0  0  0  1
```

Error (in px) 0

Figure 4.8: *A screenshot of the MC mode (A combination of specular and ambient occlusion rendering used)*

4.3 User guide

TexAlign Suite is a powerful and flexible tool that works well in different scenarios. Every mode should be used properly according to the task. User may want to use the mode he feels more comfortable with. But for most cases, some are better than others, and the MC mode is the more general.

- **Tsai mode** should be used when correspondences are very easy to set and focal length unknown. User must be very accurate and meticulous. It is necessary when distortion parameters need to be calibrated.
- **Levmar mode** should be used when correspondences are very difficult to set. In fact it is possible that only few correspondences are clear and evident. User must be very accurate and meticulous. Focal should be known in advance.
- **MI mode** should be used when a more automatic process is needed and MI optimization algorithm doesn't stop in local minima. Typical usage expects that user:
    1. sets an acceptable initial position;
    2. calls the *Optimize Full* function without estimating the focal;
    3. calls the *Optimize* function to refine the alignment, this time estimating the focal (if not present in the Exif file) too.
- **MC mode**, the most flexible and generic, should be used whenever the other modes fail. Typical usage expects that user:
    1. sets an acceptable initial position or sets few accurate correspondences and, through Levmar mode, finds an acceptable initial position;

2. launches MI mode with *Optimize* function, estimating the focal (if not present in the Exif file);
3. sets (or changes) correspondences around the non-aligned areas.
4. changes $MutualInfoweight$ to balance correctly the two subfunctions. Tests results show that $k = 0.9$ is an acceptable value (see Chapter 5).

Chapter 5

Tests Results

This Chapter will discuss tests results that prove the effectiveness of the proposed methodology. Section 5.1 will show convergence tests of the MC function and a visual comparison of the results achieved with the three approaches. Section 5.2 will present some particular cases where the use of MC is necessary to reach a fine alignment.

5.1 Validation of Mutual Correspondences

5.1.1 Convergence tests

In Section 3, Mutual Correspondences (MC) have been introduced as a sort of quality measure obtained merging the correspondences error and the Mutual Information (Section 2). The combined use of the two techniques gives a fine alignment even in problematic scenarios. What could have been mathematically unclear (dealing with such different measures), turned out to work as expected during tests. They all have been performed considering 800x600 pixels images, to keep uniformity with tests made by Corsini et al. in [MCS09].

One of the first test deals with a 3D model of a horse statue. The mesh is made of about 1 million triangles and is aligned with a 2496x1664 pixels photo (Figure 5.1). Five correspondences have been set.

This test shows that minimizing MC means approaching a fine alignment, just like minimizing correspondences error or MI.

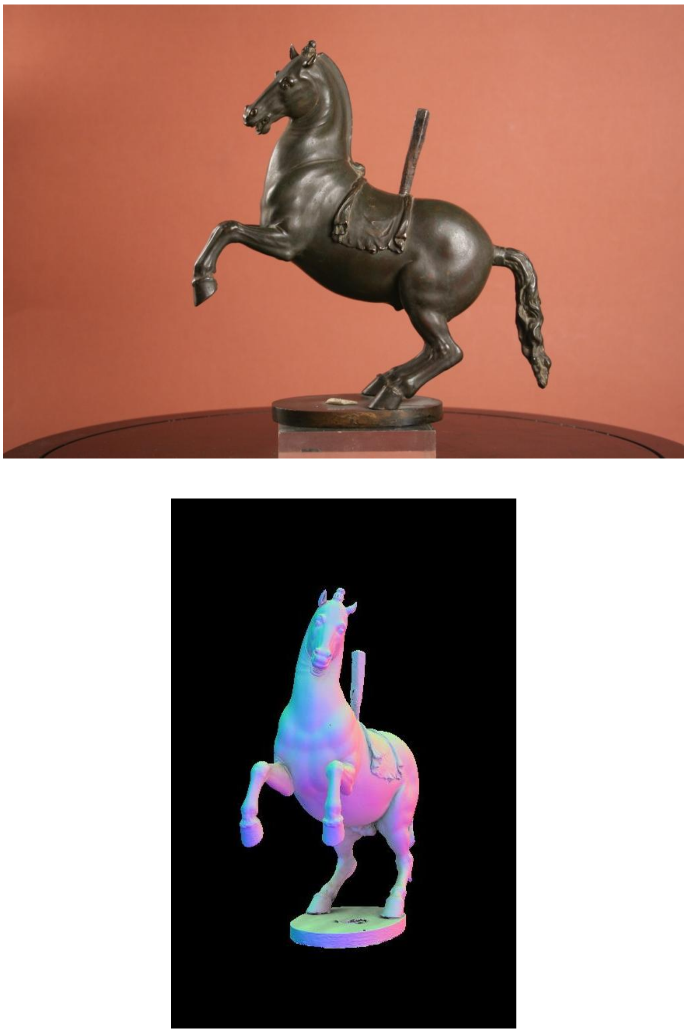


Figure 5.1: *Photo (left) and model (right)of the horse statue*

The plots in Figure 5.2 display the MC function (equation 3.1) and its weighted sub-functions at each iteration of the algorithm proposed in Section 3.5. MC function shares the slope of the correspondences error sub-function. Even starting from a very far initial position from the best alignment, MC has a very steep convergence,while the MI sub-plot appears very flat. Although the weight is very high, for the most part of iterations, correspondences error is bigger than the MI values. This is clear in the Table 5.1 where around the iteration number 140, the two values shift their roles. From that point, till the end, MI begins to be relevant, and the whole MC plot approaches the MI sub-plot. Visually speaking this means that, for the most part, the registration is driven by the correspondences; when a rough alignment is reached, MI (less and less refined by the correspondences error) steps into as MC function guide.

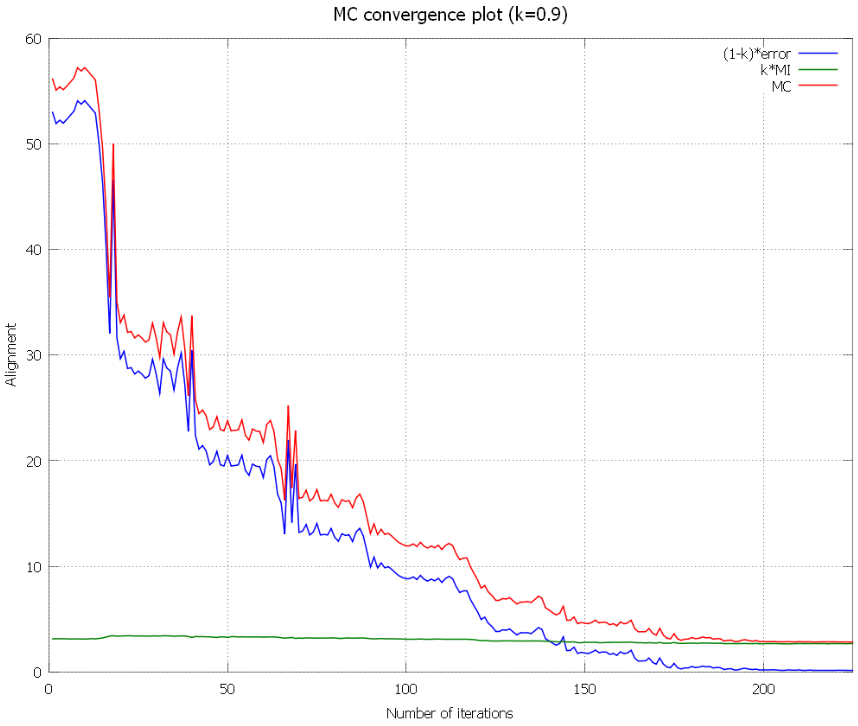


Figure 5.2: *Convergence by iterations with MC function and the two sub-functions*

n	(1-k)*error	k*MI	MC
1	52.9731	3.14399	56.1171
2	51.9025	3.15874	55.0613
3	52.2288	3.14834	55.3772
4	51.9446	3.16748	55.112
5	52.3153	3.14386	55.4592
6	52.7031	3.13107	55.8342
7	53.0951	3.14442	56.2396
8	54.0594	3.13641	57.1958
9	53.744	3.1414	56.8854
10	54.0761	3.1212	57.1973
11	53.6892	3.14253	56.8317
12	53.2815	3.16616	56.4477
13	52.8889	3.1415	56.0304
14	50.0593	3.18579	53.2451
15	46.4017	3.22623	49.6279
16	40.1358	3.31964	43.4555
17	32.0654	3.40113	35.4665
18	46.5906	3.43389	50.0245
19	31.6482	3.40344	35.0517
20	29.6527	3.41751	33.0702
21	30.3635	3.42312	33.7866
22	28.7255	3.41824	32.1437
23	28.8149	3.42005	32.235
24	28.2054	3.41477	31.6202
25	28.4877	3.42062	31.9084
26	28.2054	3.41477	31.6202
27	27.8085	3.41121	31.2197
28	28.0519	3.40314	31.4551
29	29.5952	3.42559	33.0207
30	28.2129	3.41747	31.6304
31	26.3912	3.41079	29.802
32	29.6335	3.42129	33.0548
33	28.7692	3.42072	32.1899
34	28.4908	3.4239	31.9147
35	26.6993	3.39468	30.094
36	28.7815	3.41056	32.192
37	30.1807	3.42172	33.6024
38	27.3127	3.41245	30.7251
39	22.769	3.37125	26.1402
40	30.4452	3.27486	33.72
41	22.3758	3.36475	25.7405
42	21.1011	3.35441	24.4555
43	21.439	3.35896	24.7979
44	20.9197	3.33992	24.2596
45	19.6114	3.3379	22.9493
46	19.907	3.30354	23.2105
47	20.8881	3.28103	24.1692
48	19.6173	3.33236	22.9497
49	19.5029	3.32838	22.8313
50	20.4979	3.28448	23.7824
51	19.5029	3.32838	22.8313
52	19.551	3.32847	22.8795
53	19.603	3.31992	22.9229
54	20.5248	3.33955	23.8644
55	19.0807	3.33045	22.4111
56	18.627	3.32447	21.9514
57	19.6886	3.32874	23.0174
58	19.4852	3.32567	22.8109
59	19.4443	3.33177	22.7761
60	18.4135	3.30843	21.722

n	(1-k)*error	k*MI	MC
61	20.1212	3.32461	23.4458
62	20.4794	3.32904	23.8084
63	19.4284	3.32446	22.7528
64	16.8445	3.29966	20.1441
65	16.0143	3.2506	19.2649
66	13.06	3.21357	16.2736
67	21.9695	3.24523	25.2147
68	14.1536	3.27708	17.4306
69	19.676	3.18936	22.8653
70	13.2119	3.22127	16.4332
71	13.3459	3.19697	16.5429
72	13.9647	3.23831	17.203
73	12.9759	3.22547	16.2014
74	13.2496	3.2232	16.4728
75	14.0666	3.21145	17.278
76	12.9759	3.22547	16.2014
77	13.0319	3.225	16.2569
78	12.9794	3.21941	16.1988
79	13.5872	3.23892	16.8261
80	12.7625	3.23042	15.9929
81	12.3802	3.22602	15.6062
82	13.0947	3.22718	16.3219
83	12.9335	3.22478	16.1583
84	12.9969	3.23094	16.2278
85	12.349	3.21136	15.5604
86	13.2647	3.2208	16.4855
87	13.6131	3.22577	16.8389
88	12.9023	3.22215	16.1245
89	11.4522	3.19961	14.6518
90	9.94548	3.16481	13.1103
91	10.8769	3.1459	14.0228
92	9.86266	3.15763	13.0203
93	10.3357	3.17926	13.515
94	9.87039	3.15876	13.0291
95	9.97834	3.15007	13.1284
96	9.70945	3.15523	12.8647
97	9.39344	3.14511	12.5385
98	9.11435	3.13964	12.254
99	8.95119	3.13238	12.0836
100	8.81513	3.12746	11.9426
101	8.81513	3.12746	11.9426
102	9.00685	3.13039	12.1372
103	8.7633	3.12382	11.8871
104	9.14838	3.13527	12.2836
105	8.79015	3.12944	11.9196
106	8.60949	3.12877	11.7383
107	8.80032	3.12894	11.9293
108	8.64342	3.12493	11.7683
109	8.87395	3.13076	12.0047
110	8.48433	3.11866	11.603
111	8.86141	3.12461	11.986
112	9.05576	3.12607	12.1818
113	8.85078	3.12511	11.9759
114	8.10147	3.11369	11.2152
115	7.53279	3.10455	10.6373
116	7.68067	3.1027	10.7834
117	7.69624	3.11159	10.8078
118	6.86602	3.08914	9.95515
119	6.27425	3.07093	9.34518
120	5.70546	3.01723	8.72269

n	(1-k)*error	k*MI	MC
121	4.97973	2.96945	7.94918
122	5.19799	2.99798	8.19597
123	4.62259	2.95928	7.58187
124	4.32458	2.94398	7.26856
125	3.81695	2.9482	6.76515
126	3.81695	2.9482	6.76515
127	4.00462	2.95647	6.96109
128	3.9614	2.95305	6.91445
129	4.08864	2.9595	7.04814
130	3.78433	2.94793	6.73226
131	3.5148	2.93752	6.45232
132	3.72577	2.93852	6.66429
133	3.72368	2.94123	6.66491
134	3.72925	2.94508	6.67432
135	3.64157	2.94362	6.58519
136	3.90433	2.95184	6.85617
137	4.21631	2.96097	7.17728
138	4.04924	2.96115	7.01038
139	3.15863	2.92211	6.08074
140	2.96729	2.91052	5.87781
141	2.73452	2.89513	5.62965
142	2.54051	2.86583	5.40634
143	2.70864	2.85903	5.56766
144	3.33873	2.89284	6.23158
145	2.06725	2.84106	4.90831
146	2.0662	2.84812	4.91432
147	2.35697	2.85758	5.21455
148	1.76415	2.81574	4.57988
149	1.87921	2.81455	4.69376
150	1.80064	2.82005	4.6207
151	1.76415	2.81574	4.57988
152	1.88949	2.82069	4.71018
153	2.08602	2.832	4.91802
154	1.83578	2.81697	4.65274
155	1.90783	2.81676	4.72459
156	1.89553	2.8064	4.70193
157	1.67857	2.79994	4.47852
158	1.76282	2.82064	4.58346
159	1.57913	2.80653	4.38566
160	1.92164	2.82856	4.7502
161	1.74464	2.81909	4.56373
162	1.82347	2.8323	4.65578
163	2.0689	2.83882	4.90772
164	1.31708	2.78862	4.1057
165	1.04225	2.76989	3.81214
166	1.05646	2.76942	3.82588
167	1.0658	2.75401	3.8198
168	1.34209	2.78611	4.12821
169	0.918978	2.7625	3.68148
170	0.754526	2.74778	3.50231
171	1.33126	2.80716	4.13842
172	0.762562	2.73808	3.50064
173	0.485826	2.72702	3.21284
174	0.407048	2.7324	3.13944
175	0.825115	2.80724	3.63235
176	0.407048	2.7324	3.13944
177	0.294685	2.71099	3.00568
178	0.396166	2.71967	3.11583
179	0.391555	2.73286	3.12442
180	0.531856	2.74035	3.27221

n	(1-k)*error	k*MI	MC
181	0.429102	2.73307	3.16217
182	0.4762	2.72983	3.20603
183	0.567103	2.75616	3.32326
184	0.493833	2.74809	3.24192
185	0.530268	2.7397	3.26997
186	0.383301	2.72855	3.11185
187	0.443816	2.73093	3.17474
188	0.414747	2.73631	3.15105
189	0.240949	2.69413	2.93508
190	0.307649	2.69821	3.00586
191	0.324614	2.69638	3.02099
192	0.193399	2.69265	2.88605
193	0.217185	2.6917	2.90889
194	0.326706	2.69323	3.01993
195	0.358365	2.70765	3.06601
196	0.277598	2.69669	2.97429
197	0.239895	2.69103	2.93093
198	0.277796	2.71088	2.98868
199	0.195246	2.67518	2.87042
200	0.216558	2.68671	2.90327
201	0.195246	2.67518	2.87042
202	0.20981	2.67368	2.88349
203	0.220431	2.67544	2.89587
204	0.18837	2.67679	2.86516
205	0.199704	2.67753	2.87723
206	0.199641	2.67566	2.8753
207	0.208452	2.67085	2.8793
208	0.188205	2.67864	2.86685
209	0.175859	2.67686	2.85272
210	0.21001	2.67662	2.88663
211	0.201446	2.67462	2.87607
212	0.200584	2.67626	2.87684
213	0.191609	2.68042	2.87203
214	0.163214	2.68071	2.84393
215	0.165655	2.68138	2.84704
216	0.167507	2.68337	2.85088
217	0.168101	2.68473	2.85283
218	0.172878	2.67632	2.8492
219	0.173093	2.68123	2.85433
220	0.17238	2.68295	2.85533
221	0.174346	2.67799	2.85234
222	0.167775	2.67918	2.84696
223	0.171052	2.68373	2.85478
224	0.160219	2.6794	2.83962
225	0.165609	2.67956	2.84517

Table 5.1: *Sub-functions values in MC optimization algorithm*

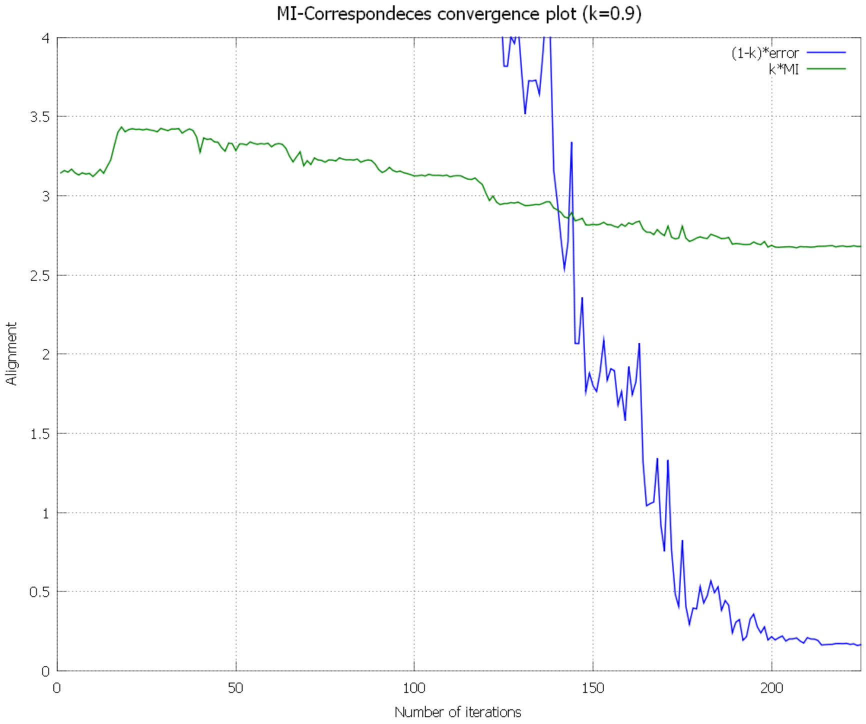


Figure 5.3: *Convergence by iterations with the MC two sub-functions*

Figure 5.3 is even more helpful to understand the relationship between the two sub-functions and how they affect each other. MI weighted plot falls off only when the the sub-functions show comparable values. This happens around a rough alignment. As a result, the MC based optimization algorithm solves the problem (that Corsini et al. left open in [MCS09]) of automatically find a rough alignment.

This behavior is clearly visible in Figure 5.4, where screen-shots show the visual feedbacks of the alignment process at some succeeding iterations. As stated in Section 3.1 visual feedback is the only practical quality measure, and, in this case, it displays the expected results: an accurate fine alignment in a decent number of iterations (225).

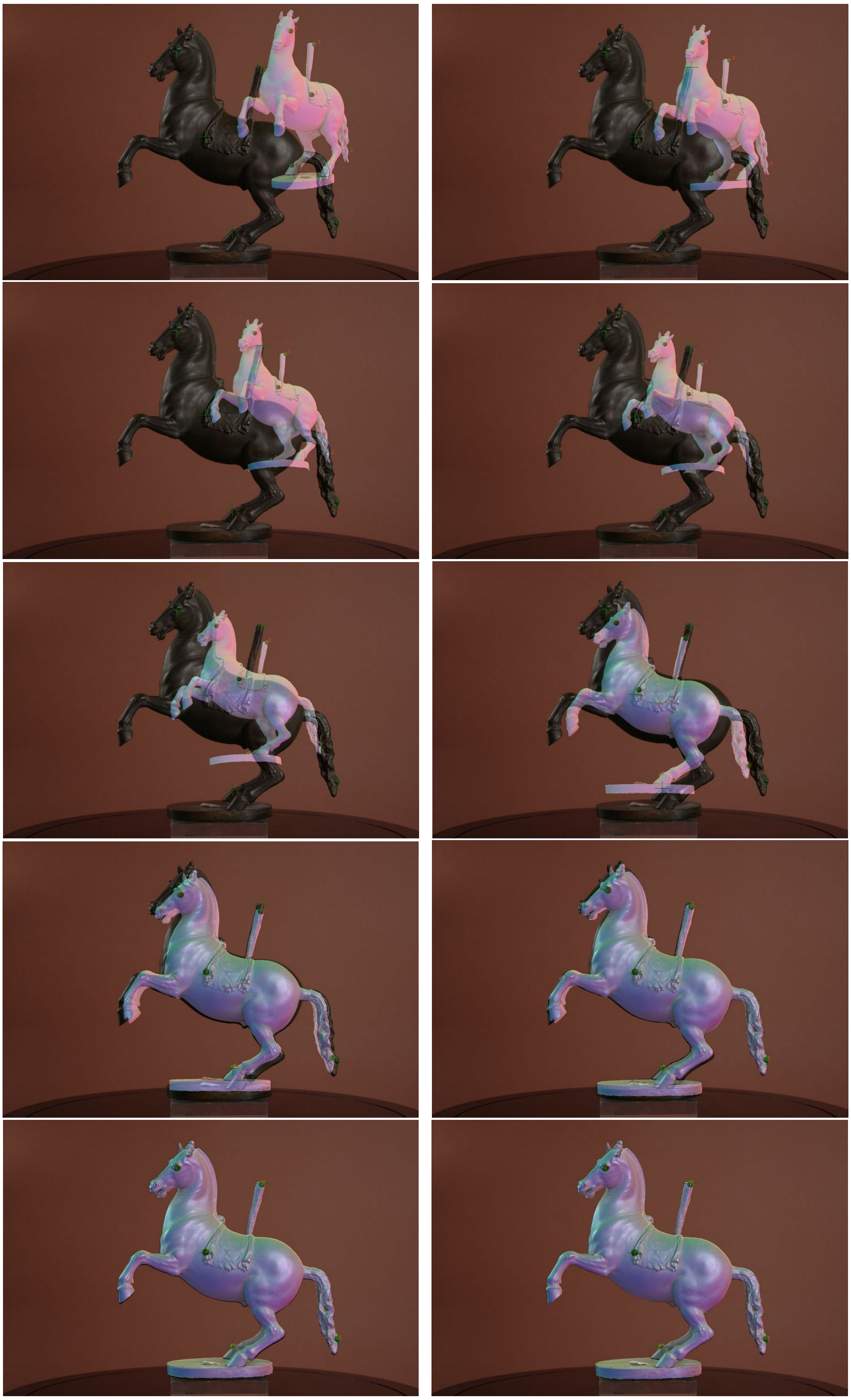


Figure 5.4: *Visual feedback of the MC alignment algorithm.* $MCWeight = 0.9$

5.1.2 Comparing the three approaches

It could be interesting to compare the different results that the three methodologies (correspondences error, MI, MC) achieve. Screen-shots are essential to have an idea of their validity. Model in Figure 5.1 represents a classic and exemplary scenario. It have been aligned in the MC mode of the TextAlign Suite on a 800x600 pixels view-port, changing $MCWeigth$ value. The final alignments are shown in Figure 5.6. They all start from the same initial position. MC based alignment($MCWeight \neq 0, 1$) achieves the best results, with $MCWeigth = 0.9$. MI based alignment ($MCWeigth = 1$) is the worst of the three, probably due to a minimum that MI cannot overpass. Focal length prevents correspondences error based alignment to be barely decent. It cannot be correctly estimated by setting only five correspondences and starting from an distant initial position. Moreover it takes twice the amount of iterations. In Levmar mode, better results are achieved with better performance only starting with correct focal length.(Sections 4.2.3).

In order to prove what visual feedbacks display, the shape of the MI and MC functions have been evaluated in the neighborhood of the optimal solution. This analysis provides useful information about the convergence obtained with the two approaches. The initial accurate registration was obtained using MC based alignment with $MutualWeight = 0.9$. Camera parameters related to the optimal position have been perturbated in different directions. An offset for a maximum of 30 pixels from the optimal position results. Considering that the viewport used is 800x600 pixels, the perturbation is meaningful. The plots in Figure 5.5 have been obtained just like the ones shown in Section 2.4. The X axis represents the distance in pixels, while the Y axis represents, respectively, the values of MI and MC.

MI plot looks very noisy and values on the Y axis move along a very small range (2.5-4). A minimum on the aligned position is evident, as well lots of local minima where the algorithm could stop. This is coherent with the visual feedback of Figure 5.6 where the optimal alignment is not achieved. MC plot shows, on a much wider range on the Y axis, a steep slope with a clear minimum on the aligned position. This allows the minimization algorithm to have a good convergence.

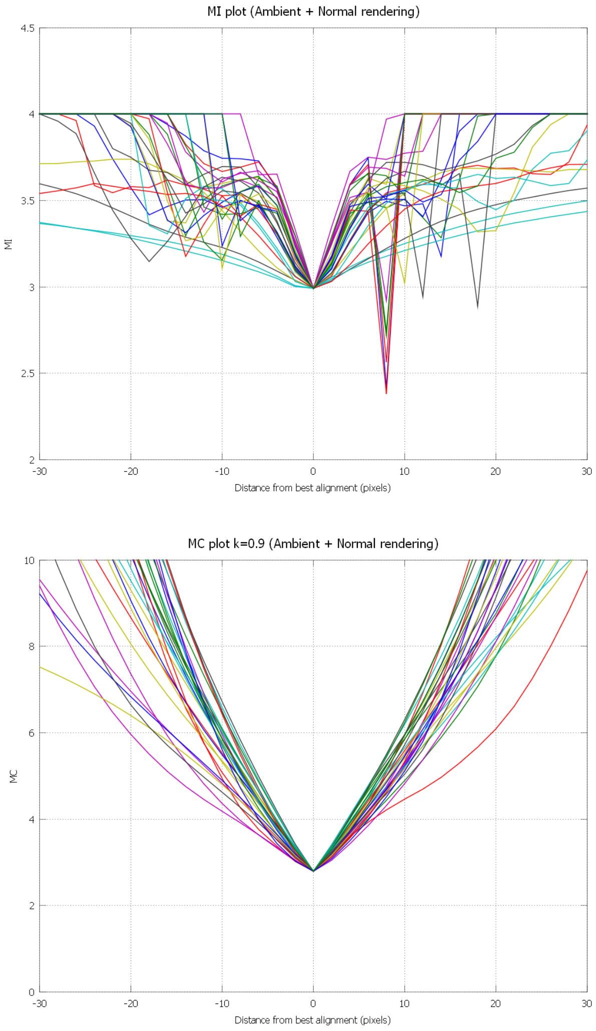
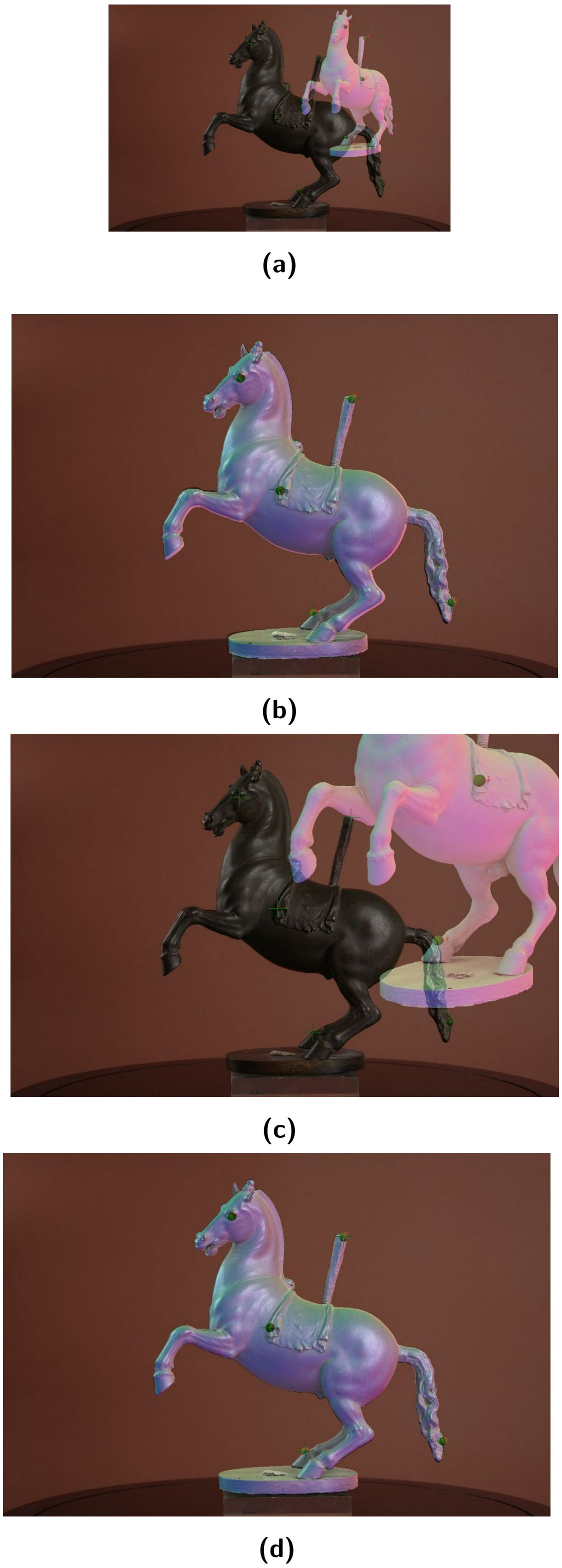


Figure 5.5: *Horse statue: MI function plot (top), MC function plot (bottom)*



(a)

(b)

(c)

(d)

Figure 5.6: *Comparison between the final alignment got with the three approaches: (a) starting point, (b) correspondences error based method ($k = 0$), (c) MI based method ($k = 1$), (d) MC based method ($k = 0,9$)*

5.2 Test cases

Specific tests have been prepared in order to prove the efficacy of the MC approach in a wider set of samples. Typically, the main reasons that prevent the classic approaches from an accurate alignment are related to:

- mismatch in the resolutions of data (images and models);
- possible distortion in the photos;
- inaccurate or incomplete geometry;
- lack of meaningful features.

Visual feedbacks and graphics of these tests prove that the use of Mutual Correspondences can overcome these problems. They have been used to compare the approaches.

The tests run on TextAlign Suite in a 800x600 pixels view-port with a limited number of correspondences (4-5). It is important to clarify that all the examples shown below can be registered using correspondences only. But, in order to have a good alignment, the number of correspondences should be at least 20. Only a high number of correspondences makes the estimation of the focal length possible. Nevertheless, the goal of the MC based alignment is to minimizing the intervention of the user, keeping a certain degree of control over the process through few correspondences. After all, if the user is able to find 20 accurate correspondences, Tsai method is the best choice because it estimates focal distortion too.

Graphics have been obtained perturbing camera parameters around the optimal position and plotting the three functions along some random directions, just like the previous graphics. Again test results show that $MCWeight = 0.9$ achieves the best alignment. Graphics of the shape of the correspondence error function are show to better understand the relationship between the three functions.

5.2.1 Shepherd nativity statue

This test tries to align a 1936x1296 pixels photo on a 3D model of 2 millions triangles with 4 correspondences (Figure 5.7). Each data source show high level of detail. Figure 5.8 displays the results of MI and MC registration.

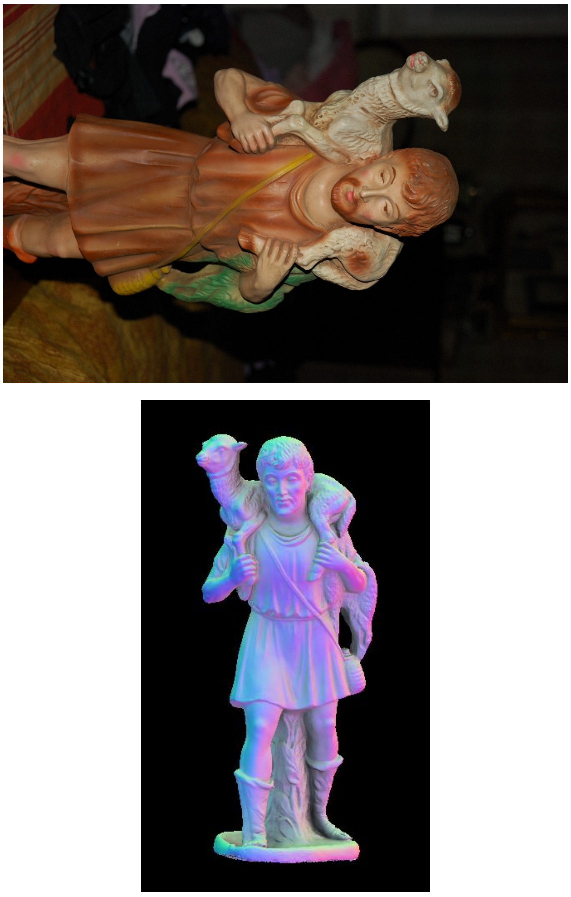


Figure 5.7: *Photo (left) and model (right)of the shepherd figurine*

Even if the result is valid, a whole accurate alignment can't be obtained just using MI; around the pelvis and the legs, the shepherd model shows an error of some pixels. This is probably due to a radial distortion of the original photo that can't be estimated with MI. The correspondences, adequately distributed, force the calibration, obtaining a better alignment. The analysis of the graphics in Figure 5.9 validates this hypothesis. MC plot keeps the slope of the convergences error plot producing a strong minimum. MI plot, instead, presents two shallower and shifted minima on a much smaller range on the Y-axis.

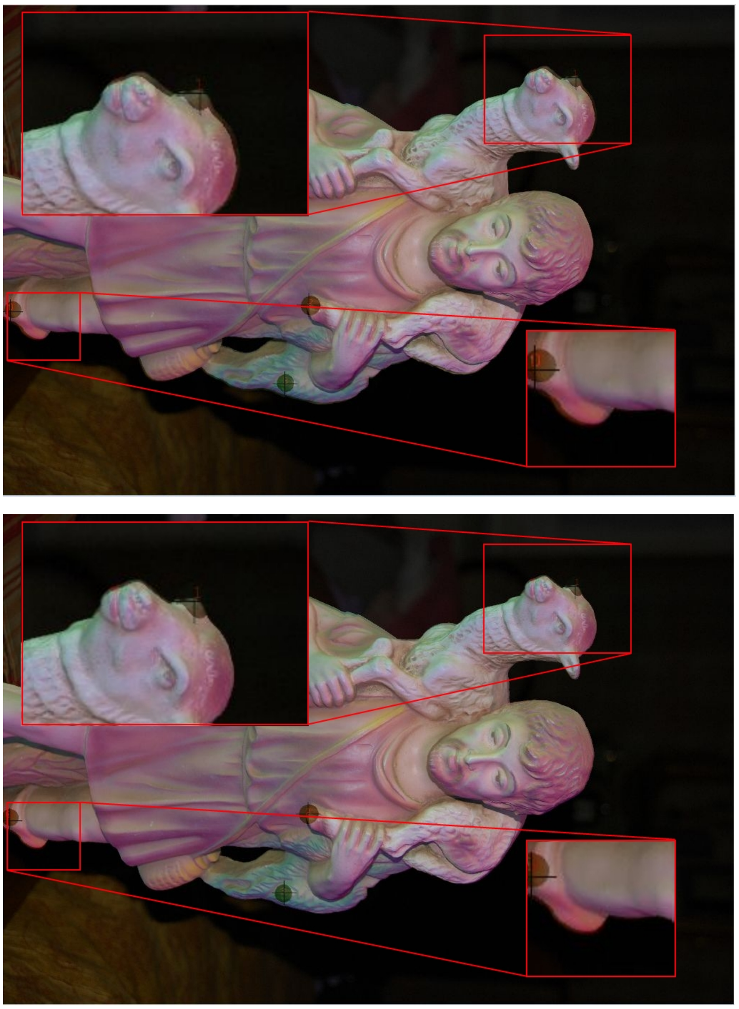


Figure 5.8: *Shepherd nativity statue: MI alignment (top), MC alignment (bottom)*

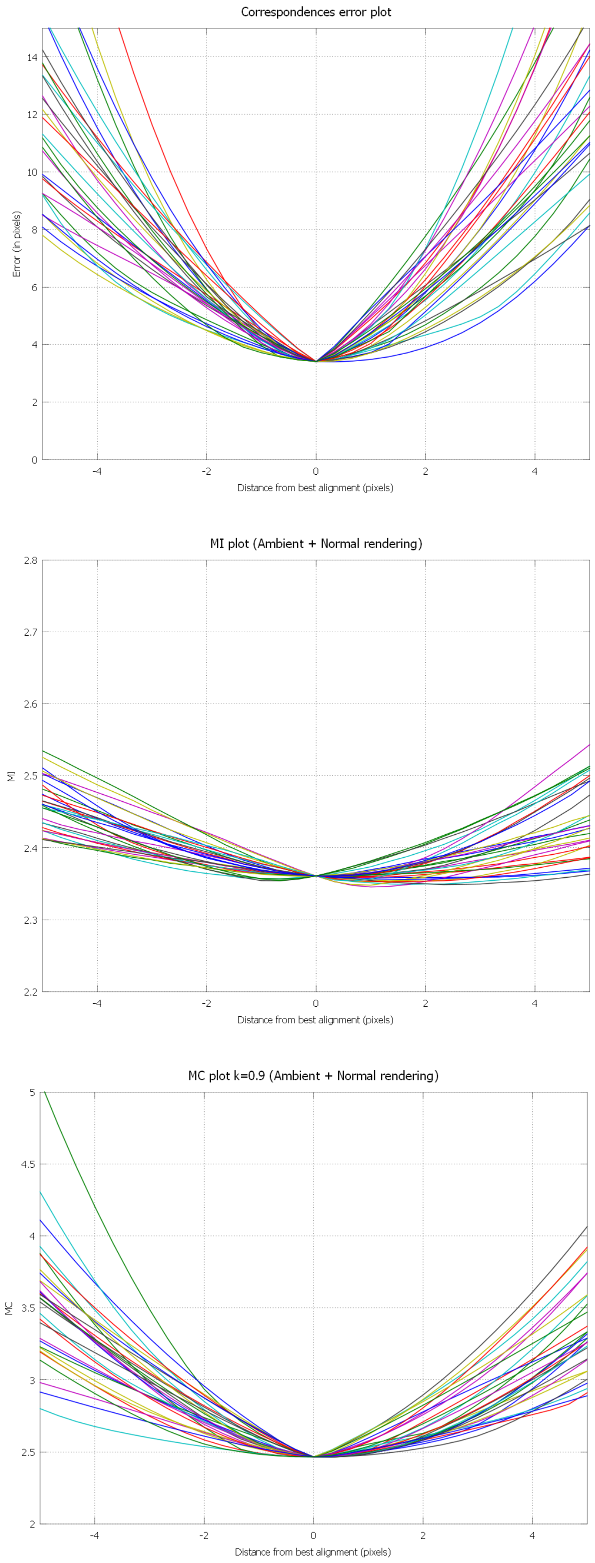


Figure 5.9: *Correspondence error plot* $(k = 0)$*(top), MI function plot* $(k = 1)$*(middle), MC function plot* $(k = 0,9)$ *(bottom)*

5.2.2 Satyr statue

This test run on a 3216x2136 pixels photo and a 3D model of about 1.4 millions triangles with 4 correspondences (Figure 5.10). The two data sources don't show the same level of detail, in addition to an incomplete and probably inaccurate geometry. Figure 5.11 shows that the best result MI achieves doesn't match a correct alignment. Probably the MI is badly affected by the color of the background which is very similar to the color of the statue.

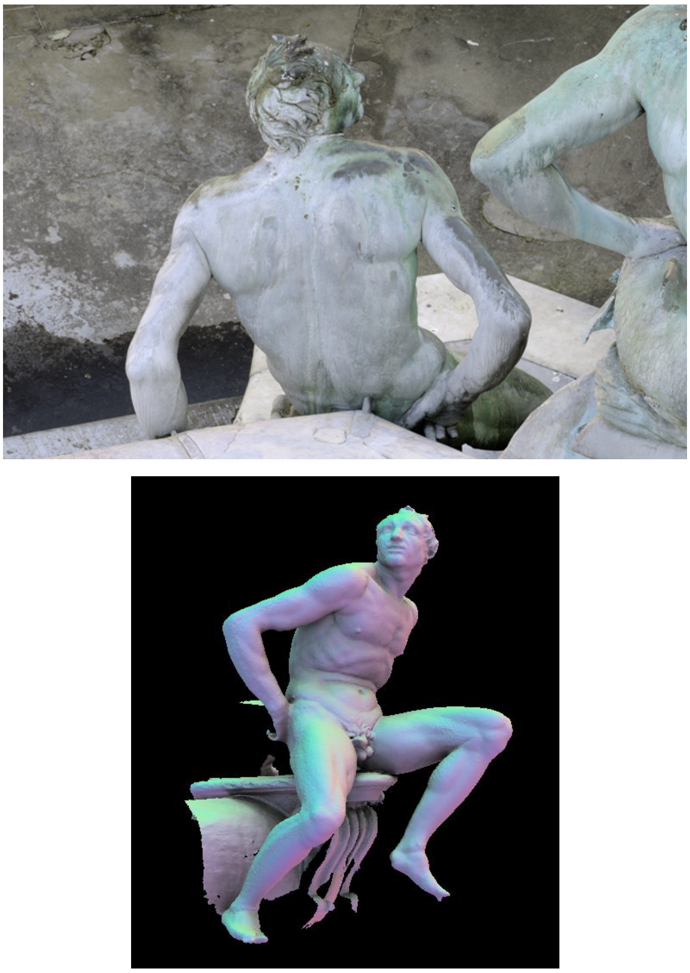


Figure 5.10: *Photo (left) and model (right)of the satyr statue*

Even if MC alignment shows an error around the elbow and the thigh, the improvement is evident. The error is only of a few pixels. The plots in Figure 5.12 validate the results. While the MI plot is noisy and full of local minima, MC plot presents a flattened minimum.

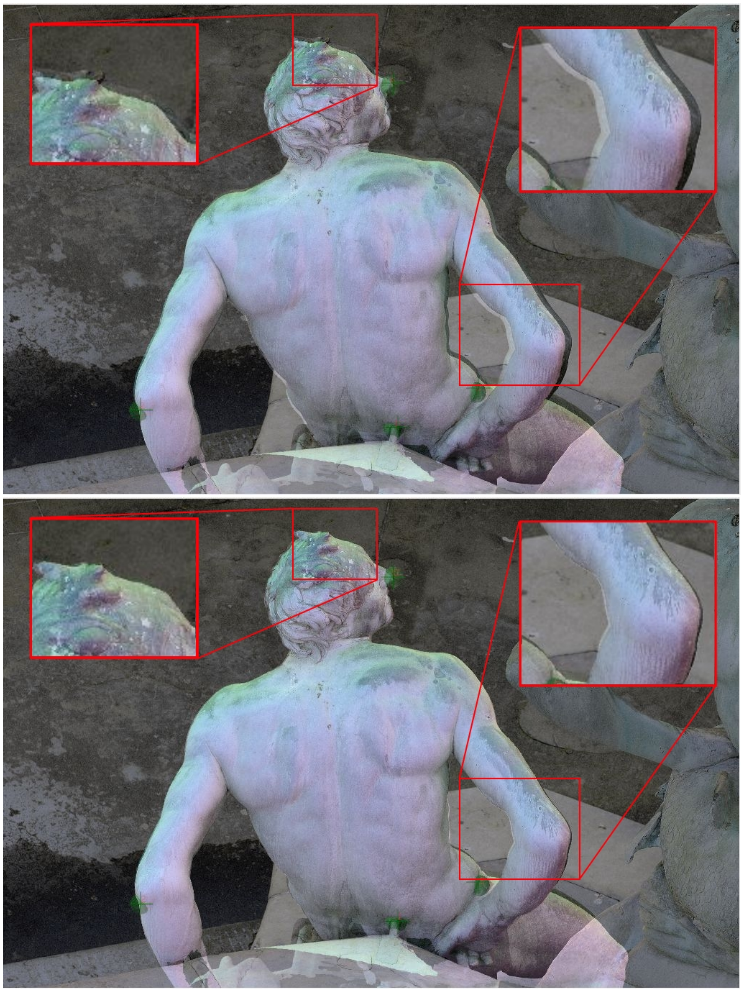


Figure 5.11: *Satyr statue: MI alignment (top), MC alignment (bottom)*

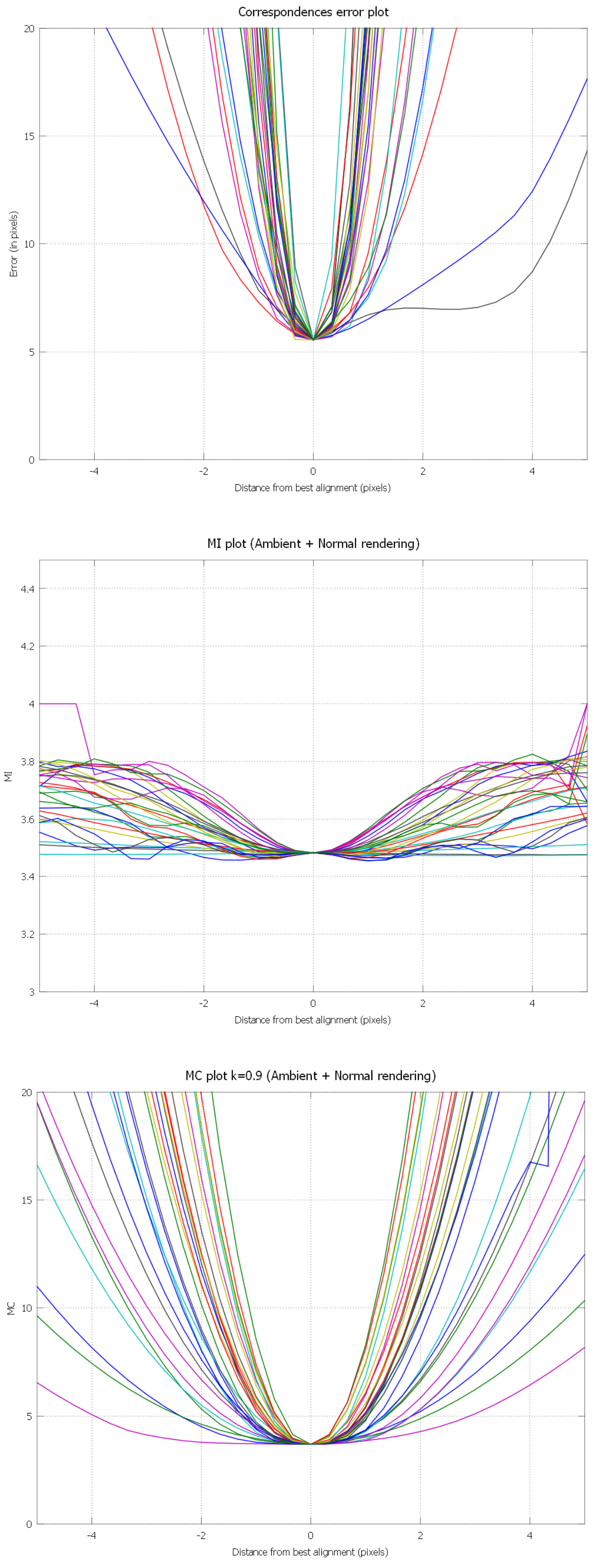


Figure 5.12: *Correspondence error plot* $(k = 0)$*(top), MI function plot* $(k = 1)$*(middle), MC function plot* $(k = 0,9)$*(bottom)*

5.2.3 Michelangelo's David

This test tries to align a 3D model of about 2 millions triangles on a 2705x3605 pixels photo with 5 correspondences (Figure 5.13). The two data sources have both an high level of detail. The 3D geometry, though extremely accurate, is incomplete. A careful analysis of Figure 5.14 highlights the limit of Mutual

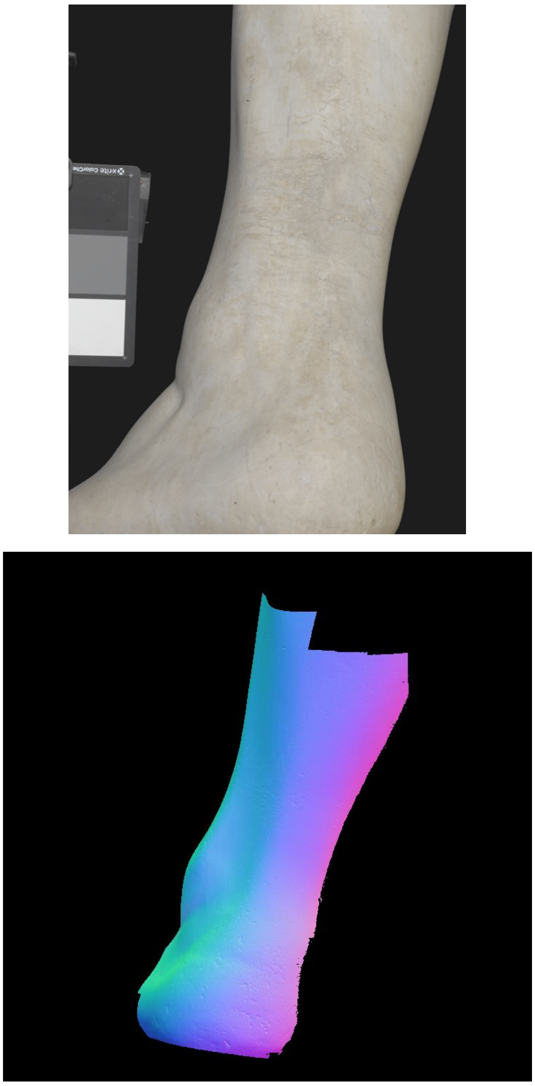


Figure 5.13: *Photo (left) and model (right)of the David's ankle*

Information. The best MI alignment doesn't achieve an optimal alignment. The mesh models only the part of David's ankle featured in the photo making any information from the silhouette useless. The lack of distinctive features doesn't help either. Findings meaningful correspondences becomes more difficult, which prevents the user from using correspondences based methods. The plots in Figure 5.15 are produced with a perturbation of 10 pixels from

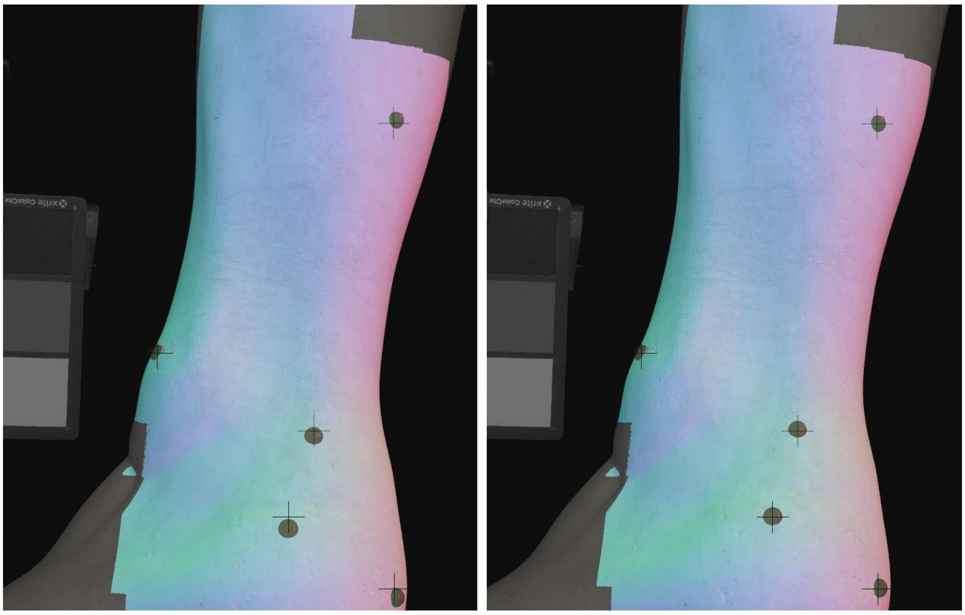


Figure 5.14: *Michelangelo's David: MI alignment (top), MC alignment (bottom)*

ndfigure

the optimal position. This time MC plot doesn't show a sharp minimum. Nevertheless the estimated error is less than one pixel and the visual feedback proves that the whole alignment is the best possible. MI plot, even if shows a minimum, looks very flat, which doesn't help a minimization algorithm to find that minimum.

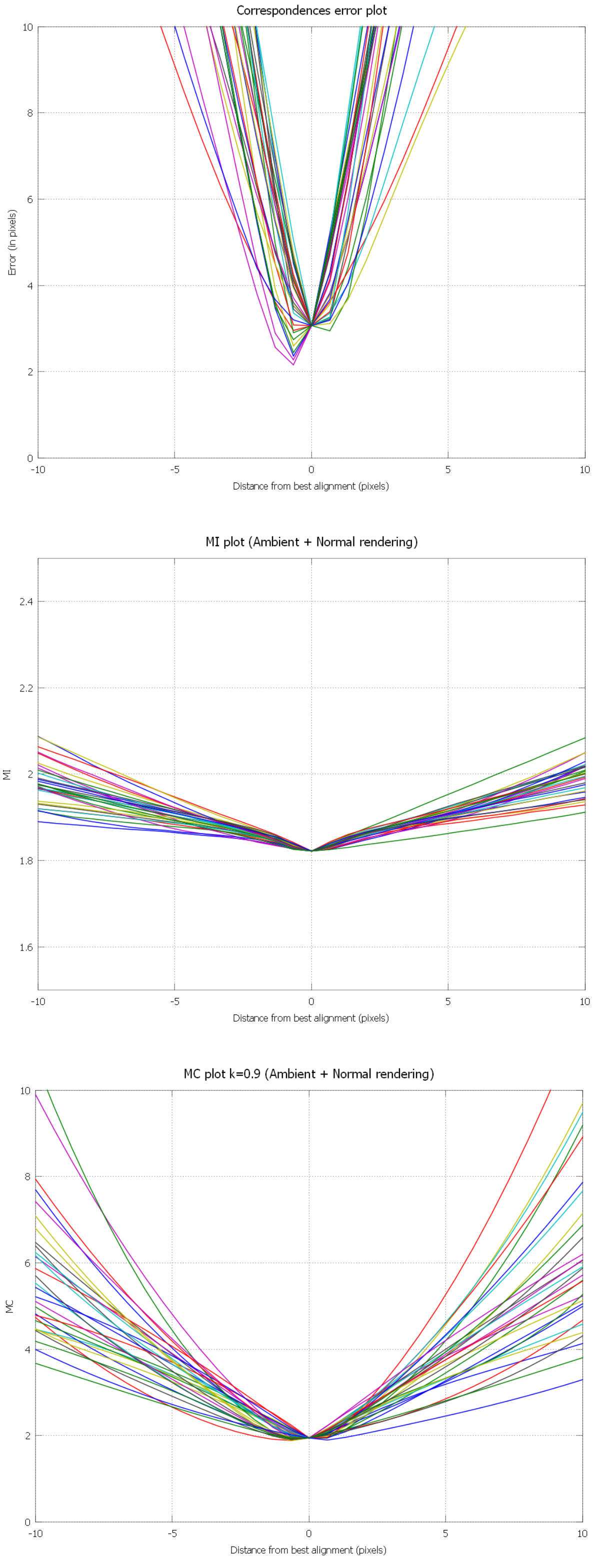


Figure 5.15: *Correspondence error plot* $(k = 0)$*(top), MI function plot* $(k = 1)$*(middle), MC function plot* $(k = 0,9)$*(bottom)*

Chapter 6

Conclusions and Future Work

The thesis proposes an original method for the image to geometry registration. This goal has been reached through the development of the Mutual correspondences alignment. It gets into the classic approaches overcoming most of their issues.

The main idea is to create a new approach merging to old methodologies. In order to make the right choice, a detailed analysis of the existing algorithms have been proposed. Correspondences error based alignment and Mutual Information based alignment have been chosen because they show complementary pros and cons, boding well for a combination of the two. Moreover they are based on the same idea, the minimization of a function, even though they deal with different quantities.

It wasn't obvious that the two approaches would work together. Adding two different quantities isn't mathematically correct, but each one is related, indirectly, to the same feature. This intuition turns out to be correct through tests targeted to specific scenarios. In these cases both correspondences error and Mutual Information can't achieve a good alignment.

The new methodology has been implemented within the TextAlign Suite tool, an open source software developed by the candidate. The tool benefits from a simple and intuitive interface, that even a inexpert user can work with. Its strength lies in its flexibility, that allows the user to switch between different modes. Very different cases can be handled, not only related to the size of the model, but also to the task being accomplished.

In the future, TextAlign Suite will be turned into a plug in for Meshlab [mes], an open source, portable, and extensible system for the processing and editing of unstructured 3D triangular meshes. The system is aimed to help the processing of the typical not-so-small unstructured models arising in 3D scanning, providing a set of tools for editing, cleaning, healing, inspecting, rendering and converting this kind of meshes. Integrating TextAlign Suite to Meshlab framework represents an important addition to a software that is embedding more and more functionalities. User interface needs to be updated, along with the architectural design. Various images would be aligned to the same model.

There are a lot of possible future extension that can be developed in order to improve the Mutual Correspondences based alignment. They can deal with:

- The estimation of focal length; this still represents an open issue. An original method to estimate the focal could be the goal of further researches. As an alternative, an adaptive mechanism could be proposed. A MC value can be chosen as a threshold. The focal length would be estimated only under this threshold.

- The smart calibration, an automatic combination of the different approaches. For example, the recommended usage of the TexAlign Suite suggests that the user first uses Levmar mode setting few correspondences and them the MC mode. This procedure can be more automatic. An adaptive system could be developed. According to some parameters (features in the image, background weight, number of correspondences, starting position) it could run different algorithms in the smartest order, to make convergence faster.

- Inferring correspondences through more images. TexAlign [FDG$^+$05, FCG$^+$05] uses a a particular data structure called correspondences graph to infer 2D-3D correspondences from 2D-2D correspondences between calibrated images. In TexAlign user needs to set correspondences. An interesting improvement could be to automatically align 2D images according to shared features and to use bundle adjustment to improve the performances of the system.

- Trying to boost MI finding new meaningful renderings or other combinations;

- Finding an automatic system that can iteratively align a set of images to a model. Each iteration would use the information computed in previous step;

- Adding distortion parameters to the variables to be estimated in MC function;

- Performing a set of ground truth tests. They should be made in a equipped lab where all the camera parameters are known in advance. Comparing them with the results of various alignment methods could give interesting information about their quality. Nevertheless this kind of tests are usually difficult and expensive. Moreover, in order to have comparable scenarios and results, every case should present comparable difficulties for each methods.

In conclusion Mutual Correspondences represents a significant improvement over the previous image to geometry registration algorithms and could be an interesting starting point for further researches.

List of Figures

Bibliography

[Ali04] D. Aliaga. Capturing, modeling, rendering 3d structures course. [web page] http://www.cs.purdue.edu/homes/aliaga/cs590g-04/, 2004.

[ARM06] Sven Andres, Niklas Röber, and Maic Masuch. Hrtf simaulations through acoustic raytracing. Technical report, Otto v. Guericke Un. Magdeburg, Germany, 2006.

[BCC$^+$07] Clara Baracchini, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Ute Dercks, Dagmar Keultjes, Claudio Montani, Matteo Scognamiglio, Roberto Scopigno, Roberto Sigismondi, and Gerhard Wolf. Cenobium cultural electronic network online: Binding up interoperably usable multimedia. In Cappellini Vito and Hemsely James, editors, *Proceedings of Electronic imaging & the visual arts, EVA 2007 Florence,*. Pitagora Editrice, March 2007.

[BLS92] Lionel Brunie, Stéphane Lavallée, and Richard Szeliski. Using force fields derived from 3d distance maps for inferring the attitude of a 3d rigid object. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 670–675, London, UK, 1992. Springer-Verlag.

[CS07] Ioan Cleju and Dietmar Saupe. Stochastic optimization of multiple texture registration using mutual information. In *DAGM-Symposium*, pages 517–526, 2007.

[DCOY03] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. *ACM Trans. Graph.*, 22(3):303–312, 2003.

[Del09] Matteo Dellepiane. *Uses of uncalibrated images to enrich 3d models information.* PhD thesis, UniversitÃ di Pisa, 2009.

[DG97] F. Dornaika and C. Garcia. Robust camera calibration using 2d to 3d feature correspondences. In *Proceedings of the International Symposium SPIE –Optical Science Engineering and Instrumentation, Videometrics V, Volume 3174*, pages 123–133, 1997.

[DVS09] Matteo Dellepiane, Andrea Venturi, and Roberto Scopigno. Image guided reconstruction of un-sampled data: a coherent filling for uncomplete cultural heritage models, 2009.

[FCG$^+$05] Thomas Franken, Paolo Cignoni, Fabio Ganovelli, Matteo Dellepiane, Roberto Scopigno, and Claudio Montani. Assisting the user in image to geometry alignment. In *3D Digital Imaging and Modeling: Applications of Heritage, Industry, Medicine and Land, Workshop Italy-Canada*, 17-18 May 2005 2005.

[FDG$^+$05] T. Franken, M. Dellepiane, F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. Minimizing user intervention in registering 2D images to 3D models. *The Visual Computer*, 21(8-10):619–628, sep 2005.

[FT86] O.D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings CVPR '86, Miami Beach, Florida*, pages 15–20, 1986.

[IOT$^+$07] Katsushi Ikeuchi, Takeshi Oishi, Jun Takamatsu, Ryusuke Sagawa, Atsushi Nakazawa, Ryo Kurazume, Ko Nishino, Mawo Kamakura, and Yasuhide Okamoto. The great buddha project: Digitally archiving, restoring, and analyzing cultural heritage objects. *Int. J. Comput. Vision*, 75(1):189–208, 2007.

[IY96] Horace Ho-Shing Ip and Lijun Yin. Constructing a 3d individualized head model from two orthogonal views. *The Visual Computer*, 12(5):254–266, 1996.

[Kat01] B. Katz. Boundary element method calculation of individual head-related transfer function. part I: Rigid model calculation. *Journal Acoustical Soc. Am.*, 110(5):2440–2448, 2001.

[KH94] Rakesh Kumar and Allen R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Underst.*, 60(3):313–342, 1994.

[KN06] Y. Kahana and P.A. Nelson. Numerical modelling of the spatial acoustic response of the human pinna. *Journal of Sound and Vibration*, 292(1-2):148–178, april 2006.

[LHS00] Hendrik P. A. Lensch, Wolfgang Heidrich, and Hans-Peter Seidel. Automated texture registration and stitching for real world models. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, page 317, Washington, DC, USA, 2000. IEEE Computer Society.

[Lou04] M.I.A. Lourakis. Levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. `http://www.ics.forth.gr/~lourakis/levmar`, July 2004. [Accessed in 2009.].

[Low91] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:441–450, 1991.

[LPC+00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D scanning of large statues. In *SIGGRAPH 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 131–144. Addison Wesley, July 24-28 2000.

[LS05] Lingyun Liu and Ioannis Stamos. Automatic 3d to 2d registration for the photorealistic rendering of urban scenes. *CVPR*, 2:137–143, 2005.

[LSY+06] Lingyun Liu, Ioannis Stamos, Gene Yu, George Wolberg, and Siavash Zokai. Multiview geometry for texture mapping 2d images onto 3d range data. *Computer Vision and Pattern Recognition*, 02:2293–2300, 2006.

[MCS09] F. Ponchio M. Corsini, M. Dellepiane and R. Scopigno. Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. In *Proceeding of Pacific Graphics*, pages Volume 29, Number 7. Springer-Verlag, 2009.

[MCV+97] F. Maes, A. Collignon, D. Vandeermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions in Medical Imaging*, 16:187–198, 1997.

[mes] Meshlab,an open source, portable, and extensible system for the processing and editing of unstructured 3d triangular meshes. `http://meshlab.sourceforge.net/`. [Accessed in 2009.].

[MK99] K. Matsushita and T. Kaneko. Efficient and handy texture mapping on 3D surfaces. In P. Brunet and R. Scopigno, editors, *Computer Graphics Forum (Eurographics '99)*, volume 18(3), pages 349–358. Blackwell Publishers, 1999.

[MVS99] F. Maes, D. Vandermeulen, and P. Suetens. Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information. *Medical Image Analysis*, 3(4):373ï¿$\frac{1}{2}$386, 1999.

[NK99] Peter J. Neugebauer and Konrad Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, 18(3):245–256, 1999.

[Par62] E. Parzen. On estimation of a probability density function and mode. 33:1065–1076, September 1962.

[PK08] Giorgio Panin and Alois Knoll. Mutual information-based 3d object tracking. *Int. J. Comput. Vision*, 78(1):107–118, 2008.

[PMV03] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8):986–1004, 2003.

[Pow04] M.J.D. Powell. newuoa: software for unconstrained optimization without derivatives in C. `http://www.inrialpes.fr/`

bipop/people/guilbert/newuoa/newuoa.html, 2004. [Accessed in 2009.].

[TDLD07] Nicolas Tsingos, Carsten Dachsbacher, Sylvain Lefebvre, and Matteo Dellepiane. Instant sound scattering. In *Proc. of the Eurographics Symposium on Rendering*, 2007.

[Tsa87] R. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), August 1987.

[Tsa95] Roger Tsai. This software package contains routines for calibrating roger tsai's perspective projection camera model. http://www.cs.cmu.edu/~rgw/TsaiCode.html, 1995. [Accessed in 2009.].

[VWMW97] Paul Viola and III William M. Wells. Alignment by maximization of mutual information. *Int. J. Computer Vision*, 24(2):137–154, 1997.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.

Ringraziamenti

E' buffo come, spesso, alla fine di un evento tanto atteso, ci si senta arricchiti, stanchi, entusiasti e tristi allo stesso tempo. Probabilmente è quello che prova un regista quando batte l'ultimo ciak di un film molto personale; quando l'ultimo ballo della serata è quello con la ragazza che non avevi il coraggio di invitare; quando scende il sipario dell'ultima replica di una piéce in cui non reciterai più; quando scrivo i ringraziamenti di una tesi da cui ho appreso tanto, a cui ho dato molto.
Ma il regista avrà già un'idea per il prossimo film; la ragazza che non avevi il coraggio di invitare potrà tornare martedì prossimo alla stessa milonga; il sipario si riaprirà e gli attori torneranno sul palco. Io non penso di scrivere altre tesi (abbiamo già dato), ma spero che si presenteranno sempre nuove ed interessanti opportunità.
Un importante capitolo si chiude, ma bisogna "guardare sempre avanti" e conservare la memoria di ogni vittoria e sconfitta.
Ringraziamenti sinceri vanno:

al dottor Matteo Dellepiane, per il continuo supporto, la costante presenza (fisica e online), e il contributo dato a questa tesi. Non penso di aver mai sentito di un relatore altrettanto disponibile, competente, stimolante e incoraggiante, oltre che dotato di un acuto senso dell'umorismo, cosa che non guasta mai;

al professor Paolo Cignoni, indiscusso punto di riferimento, nonchè colui che mi ha avviato in questo campo;

ai dottori Giampaolo Palma, Federico Ponchio, Paolo Brivio e a tutto il Visual Computing Lab dell'istituto ISTI-CNR di Pisa per l'aiuto e i consigli, ma soprattutto per il loro spirito di squadra;

a mio padre, perchè con una mano indica gli ostacoli e con l'altra stringe la mia;

a mia madre, perchè basta un suo abbraccio e tutti i brutti sogni svaniscono;

a Sabino, perchè solo insieme possiamo godere appieno delle gioie della nostra famiglia;

a mia nonna perchè è stata la prima a mettermi la matita in mano e perchè tiene vivo il ricordo; in esso il sorriso cortese di mio nonno non si spegnerà mai;

ad Angela perchè crede in me e in ciò che voglio;

ad Alessia perchè è la più bella persona che avrei mai potuto incontrare; per la nostra unità di interessi, per il suo buongusto, per il suo intuito, per il tango e per quel RATMAN lasciato strategicamente in bagno.

ad Antonio per le conoscenze che condivide e la disponibilità;

a Daniele, Carlo, Ferdinando e Fiorella per aver riempito e reso casa un appartamento;

a Matteo per le sue risposte sempre pronte; a Francesco per il suo burbero buon cuore; a Daniele per il suo essere tanto divertente quanto giudizioso, a Roberto per le sue arguzie, a Ugo per la sua fermezza, ad Andrea per il suo senso pratico, a Sabrina per il suo animo gentile;

a Mimmo perchè esiste davvero, a Olga perchè guarda alla vita con romanticismo e concretezza, ad Antonella perchè è sincera e fidata e sa farsi valere, a Lara perchè per noi sarà sempre come al liceo;

a tutti gli amici del tango e della compagnia per ciò che mi avete dato e che avete tirato fuori da me;

all'Artista perchè non smette di animare e colorare la vita.