Universidade Federal Fluminense

Giancarlo Vasconcelos Taveira do Nascimento

Automatic Alignment and Reconstruction of Facial Depth Images

NITERÓI

2012

Universidade Federal Fluminense

Giancarlo Vasconcelos Taveira do Nascimento

Automatic Alignment and Reconstruction of Facial Depth Images

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Master of Science. Topic Area: Visual Computing.

Advisor:
Prof. D.Sc. Leandro Augusto Frata Fernandes

NITERÓI

2012

Automatic Alignment and Reconstruction of Facial Depth Images

Giancarlo Vasconcelos Taveira do Nascimento

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Master of Science. Topic Area: Visual Computing.

Approved by:

Prof. D.Sc. Leandro Augusto Frata Fernandes / IC-UFF
(Advisor)

Prof. D.Sc. Esteban Walter Gonzalez Clua / IC-UFF

Prof. Ph.D. Ricardo Cordeiro de Farias / COPPE-UFRJ

Niterói, December 14th, 2012.

To my family and friends.

Acknowledgments

I would like to thank my parents José Taveira do Nascimento and Mércia Rejane de Vasconcelos Silva, who always supported me in every way, giving me unconditional love. You taught me right from wrong and showed me the importance of a good education. I can never thank you enough!

To Cristiane Salles, my dear girlfriend who always believed in me, even when I didn't. Your faith in me kept me going! I love you with all my heart!

To Leandro Augusto Frata Fernandes, for all your patience. Your commitment and incredible dedication taught me what it truly means to be a researcher.

To Esteban Walter Gonzalez Clua, MediaLab coordinator, for all the great opportunities and for encouraging all of us to pursue our dreams. Because of you, I had the opportunity to participate in several projects that allowed me to become a better professional.

To my colleagues at MediaLab, for all the tips and advices. My deepest and sincere thanks!

To my roommates, André Brandão and Christian Ruff, for your friendship and support during the good and bad times! I will never forget!

To all my friends, for being there for me.

To Universidade Federal Fluminense, for providing me a working environment, and CAPES, for the scholarship.

This project was partially sponsored by FAPERJ (Process E-26/111.468/2011).

To everyone who supported me, thank you!

Resumo

Sistemas classificadores de rosto, gênero, etnia e faixas etárias funcionam, de maneira geral, através de um processo de alinhamento, extração de características e identificação. A qualidade do processo de alinhamento é, portanto, primordial para o bom desempenho do processo de identificação. Além disso, a capacidade de classificação é dependente da qualidade dos dados fornecidos. Por exemplo, a classificação pode ser influenciada por dados de cor ou profundidade com pequenas falhas ou porções danificadas. Em consequência disto, a reconstrução das imagens de maneira apropriada é crucial para o correto funcionamento destes sistemas.

Este trabalho apresenta um método elegante e eficaz para alinhar e reconstruir imagens de profundidade de uma maneira completamente automática. A abordagem utiliza informação extraída de *pixels* válidos para ajustar uma função de interpolação suave que reconstrói de maneira natural a informação de profundidade em regiões danificadas e, também, que permite calcular a transição suave entre valores de *pixels* existentes. A abordagem também utiliza pontos notáveis da face para determinar a posição e rotação atual da imagem da face no espaço tridimensional. A relação entre o conjunto de ponto notáveis da face e um conjunto de pontos canônicos é utilizada para mapear a superfície no espaço atual para um espaço padronizado, onde a imagem alinhada resultante é obtida do traçado de raios de uma câmera *pinhole* em direção à superfície reconstruída.

Em comparação à soluções existentes, a abordagem proposta é direta e se encaixa facilmente nos *pipelines* de processamento comumente utilizados. Esta abordagem também pode ser estendida para produzir imagens coloridas de maneira correta para as imagens de profundidade finais. Os experimentos realizados mostram que os erros de aproximação decorrentes do uso do método proposto são até duas ordens de magnitude inferiores aos erros encontrados usando alinhamento bidimensional de imagens e interpolação linear.

Este trabalho apresenta também um estudo comparativo entre quatro funções de interpolação (*i.e.,* vizinho mais próximo, linear, vizinho natural e *Thin Plate Spline*) usando o método de alinhamento proposto (no domínio tridimensional). Cada método de interpolação foi aplicado como parte de um processo de classificação de gênero disponível na literatura. A partir dos resultados obtidos é possível afirmar qual abordagem de interpolação é a mais indicada para aplicação em conjunto com classificadores diretos ou de convergência iterativa.

Palavras-chave: imagem de profundidade, alinhamento, interpolação não-linear, reamostragem, imagem de face.

Abstract

Face, gender, ethnic and age group classification systems often work through an alignment, feature extraction, and identification pipeline. For that reason, the classification capacity depends on the quality of the source data. For instance, the classification can be affected by color or depth data with small flaws or damaged regions. Appropriate image reconstruction is therefore crucial for the correct operation of those systems.

This work presents an elegant and effective method for aligning and reconstructing facial depth images from damaged depth data in a completely automatic way. The approach uses information extracted from valid pixels to adjust a smooth interpolating function that naturally reconstructs the depth information of missing pixels and computes smooth transitions among existing ones. The approach also explores facial landmarks in order to determine the actual position and orientation of the imaged face in the 3-dimensional space. The relation between the set of landmarks in the actual face and a set of canonical landmarks is used to map the shape of the imaged face to a standard space where the resulting aligned image is generated by ray casting the reconstructed surface.

In contrast to existing solutions, the proposed approach is straightforward and easily fits into popular processing pipelines. It can also be extended to produce correct color images for resulting depth images. The experiments show that the approximation errors produced by the proposed method are up to two orders of magnitude smaller than those using 2-dimensional alignment with linear interpolation.

This work also presents a comparative study among four distinct interpolation methods (*i.e.,* nearest-neighbor, linear, natural-neighbor and Thin Plate Spline) using the proposed alignment method (in the 3-dimensional domain). Each interpolation method was applied as part of a gender classification process available in the literature. From the obtained results it is possible to state which interpolation approach is better to be applied with direct or iterative classifiers.

Keywords: depth image, alignment, non-linear interpolation, resampling, face image.

Contents

Glossary

DLL	:	Dynamic Link Library
FF	:	False Females
FM	:	False Males
GPU	:	Graphics Processing Unit
MCC	:	Matthews Correlation Coefficient
MEX	:	MATLAB Executable
PCA	:	Principal Component Analysis
PGA	:	Principal Geodesic Analysis
PPM	:	Portable Pixel Map
RM	:	Relief Mapping
SWPGA	:	Supervised Weighted Principal Geodesic Analysis
TF	:	True Females
TFR	:	True Females Rate
TM	:	True Males
TMR	:	True Males Rate
TPS	:	Thin-Plate Spline
UND	:	University of Notre Dame
XML	:	Extensible Markup Language

List of Figures

List of Tables

Chapter 1

Introduction

The ability to retrieve information from facial depth images has many practical applications including face recognition, age group estimation, gender and ethnic group classification [Lu et al. 2005, Wu et al. 2010, Wu et al. 2011]. Recently, damaged portions of data in colored images are less frequent due to the new digital capture systems. Unfortunately, depth data is often damaged due to limitations intrinsic to off-the-shelf depth-image capturing systems. Examples include, but are not limited to, depth shadowing and the influence of reflective, refractive and infrared absorbing materials in the scene. Also, the amount of pixels covering the imaged face and face's orientation often vary from image to image, making it difficult or even impossible the use of captured images without the proper alignment and reconstruction of depth data.

Each computer vision research that needs to perform alignment and reconstruction of facial depth data usually presents its own solution to the problem. An established technique in literature is to identify some facial features by curvature, and compute the alignment based on them [Moreno et al. 2005]. PCA-based solutions have also been proposed [Stormer and Rigoll 2008]. However, most of the techniques do not make proper use of depth information while performing the alignment, restricting the solution to the 2-dimensional image plane. Linear interpolation is commonly used to fill the holes [Wu et al. 2010], leading to unnatural flat artifacts on the facial surface.

This work presents an elegant and effective method for aligning and reconstructing facial depth images from damaged depth data in a completely automatic way. The approach uses information extracted from valid pixels to adjust a smooth interpolating function that naturally reconstructs the depth information of missing pixels (see Figure 1.1) and computes smooth transitions among existing ones. The approach also explores facial landmarks in order to determine the actual position and orientation of the imaged face in

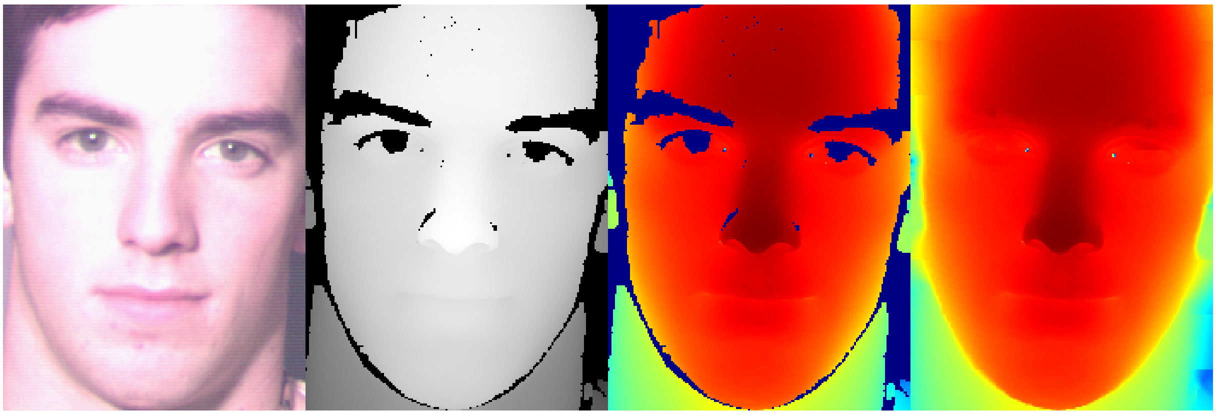

Figure 1.1: From left to right: color image, depth as grayscale, depth as colormap (navy blue pixels denote missing depth information), reconstructed depth information.

the 3-dimensional space. The relation between the set of landmarks in the actual face and a set of canonical landmarks is used to map the shape of the imaged face to a standard space where the resulting aligned image is generated by ray casting the reconstructed surface.

In contrast to many existing solutions, the proposed approach is straightforward and easily fits into popular processing pipelines. It can also be extended to produce correct color images for resulting depth images. The experiments show that the approximation errors produced by the proposed method are smaller than those using 2-dimensional alignment with linear interpolation (see Figure 4.3). Also, the experiments show that the proposed smooth interpolation method improves the results of existing computer vision techniques, such as state-of-the-art gender classification models.

1.1 Main Idea

Interpolation methods such as nearest-neighbor and linear interpolation are commonly used to fill portions of missing data. This work proposes the use of a sophisticated interpolation function to smoothly reconstruct damaged depth data in facial depth images. The key observation is that "better reconstructions will produce better depth maps". Thus, it is intuitive that a smoother interpolation will lead to better results. However, how better it will be in practical applications has not been presented in literature. Therefore, the objectives of this work are to present an alignment and resampling procedure for facial depth images and to investigate the quality impacts of surface reconstruction in the context of gender classification models.

Some of the primary challenges include:

1. Adjust an interpolation method that smoothly fits the face surface;
2. Align several distinct face images with varying poses;
3. Resample depth values at arbitrary coordinates; and
4. Compare different interpolation schemes against the proposed one.

The interpolation method chosen for this work uses Thin-Plate Splines (TPS). The first experiments showed that a naive approach (*i.e.,* trying to adjust a single TPS to fit the face surface) presents high computational costs. Not only it requires a large amount of memory but also its processing time increases rapidly as more control points are added (challenge 1). The proposed solution is to divide the image into several small blocks and fit an interpolation function to each block. This approach considerably reduces both the total processing time and the memory footprint during the calculation of the interpolation method's coefficients.

In facial classification systems, it is essential that all images are as perfectly aligned to each other as possible. However, it is quite challenging to achieve the perfect alignment. Since, in most cases, the end-user has no control of the image capturing step, the subjects' poses may vary (challenge 2). In order to correctly align several distinct images, this work uses a tetrahedron based on mean facial feature distances and after that all faces in the dataset are mapped to the space where this tetrahedron resides. By doing so, all images will be aligned in a standard space.

After the alignment step, it is often necessary to resample depth values at coordinates that were not obtained when the original images were captured (challenge 3). The proposed resampling procedure is based on the Relief Mapping (RM) [Policarpo et al. 2005] technique and consists in casting rays from a pinhole camera, which is in the standard space, and then find the first intersection of each casted ray with the smooth surface described by the interpolation function.

1.2 Demonstration and Analysis

The proposed alignment technique and smooth surface reconstruction were implemented. The details are discussed in Chapter 3. The proposed approach was tested with a well-known dataset, the UND Biometric dataset [Chang et al. 2003], which contains both color and range facial information of 277 subjects, totaling 953 images.

The presented 3-dimensional alignment method was compared to a common 2-dimensional alignment approach and the results proved that the error in the proposed method can be two orders of magnitude smaller.

In order to evaluate the proposed reconstruction method, the smooth interpolation scheme described in this work was compared against other three commonly used interpolation methods, namely (a) Nearest-neighbor, (b) Linear and (c) Natural-neighbor. Two gender classification models, presented in [Wu et al. 2011], were used to enforce an unbiased environment for testing. The gender classification source code was kindly provided by the authors. The results showed that, unlike the large differences found when comparing the two alignment procedures, the difference when comparing the interpolation methods was small, but favorable to the proposed interpolation scheme, specially when used with classifiers based on iterative training procedures. Figures 4.5 and 4.6 show those differences, where several statistical metrics are compared while running both gender classification models.

1.3 Contributions

The main contribution of this work is the presentation of an elegant and effective method for aligning and reconstructing facial depth images in a completely automatic way.

In this context, the contributions of this work include:

- An automatic 3-dimensional alignment and resampling approach based on Relief Mapping;
- A blocks division approach that allows the use of a sophisticated interpolation method that smoothly reconstructs damaged depth data.

In addition to these original results, this work also presents a set of experiments that can be used as a guide for researchers while they choose which alignment techniques and interpolation functions best fit the context of their problem. By comparing two different alignment procedures (based on 2-dimensional and 3-dimensional alignment) and by comparing four distinct interpolation methods, it is possible to not only state which approach is better but also numerically measure the practical impact of each method in the final classification process. In essence, other contributions of this work include:

- A comparative study that measures the error differences between 2-dimensional and 3-dimensional alignment;

- A comparative study of the performance of four different interpolation methods.

Finally, this work aims to provide enough evidence to prove that applying sophisticated alignment and reconstruction methods increase the performance of gender classification models, and then encourage the use of these methods in other applications that use depth images as input data.

Chapter 2

Related Work

Depth data can be interpreted as a 2-dimensional array or a matrix where each cell contains a value that represents depth information. The depth data is usually within a very well defined range (e.g. from 800 to 4000, values expressed in millimeters). However, not all cells of this matrix represent valid surface points. During the acquisition of the depth data, the infrared light emitted from the scanner may be reflected to different directions and, in such a case, a part of the emitted light is not captured back by the infrared sensor. This results in the depth data having invalid values. Therefore, it is necessary to replace the inconsistent values by estimated ones in order to reconstruct the missing portions of depth data.

Some depth file formats specifications, such as the one used by the UND Biometric Database (Collection D) [Chang et al. 2003], provide one matrix of boolean values stating whether the evaluated cell contains a valid measurement or not. As a result, the depth data ends up having a non-structured characteristic, usually referred to as *scattered data.*

When interpolating uniform data, the interpolation techniques take advantage of the uniformity or gridlike nature of the sample data to compute interpolated values. On the other hand, when the sample data is scattered, the interpolation techniques usually use a triangulation-based approach as a basis for computing interpolated values. There are many ways to create such a triangulation, but the triangulation that is best suited to interpolation is the Delaunay triangulation [Preparata and Shamos 1985].

A set of scattered points defined by locations X and corresponding values V can be interpolated using a Delaunay triangulation of X. This produces a surface of the form $V = F(X)$. The surface can be evaluated at any query location QX, using $QV = F(QX)$, where QX lies within the convex hull of X. The interpolant F always goes through the

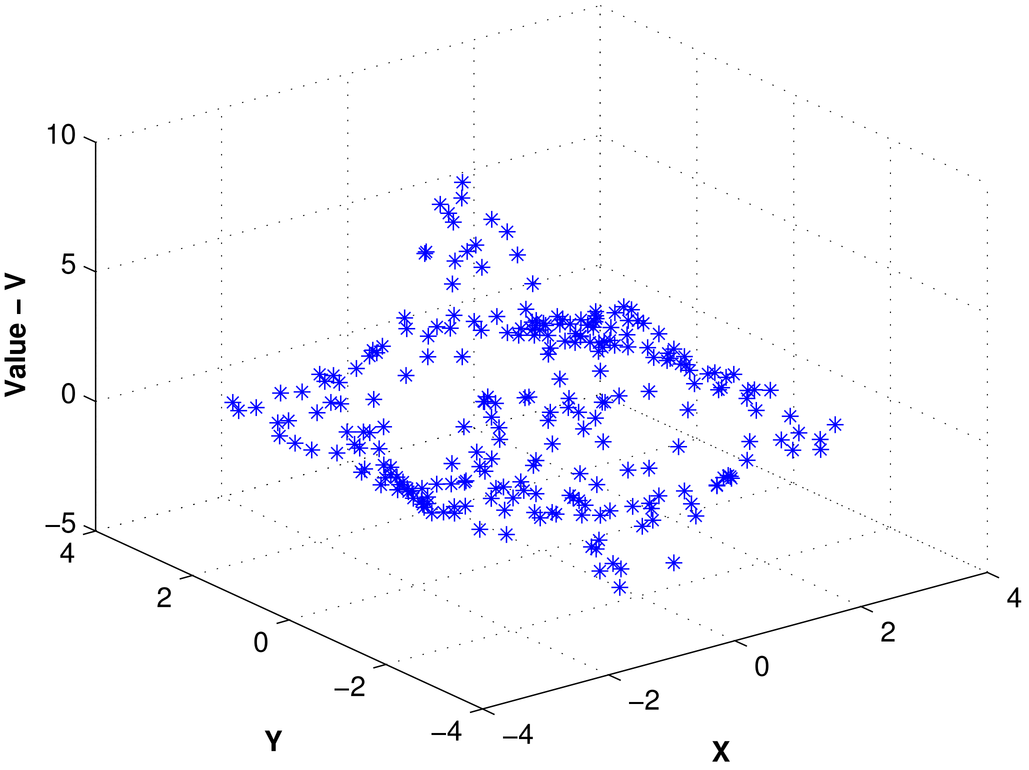

Figure 2.1: An example of scattered points with discontinuous data. The location of the points is defined in the XY-plane and the corresponding values are represented by the V-axis.

data points specified by the sample [MATLAB 2009a].

In facial depth images, there are three common interpolation methods that have been used to reconstruct missing data. For this reason, they are presented as related work. The common interpolation methods are:

1. Nearest-neighbor interpolation, where the interpolating surface is discontinuous (Section 2.1).

2. Linear interpolation, where the interpolating surface is C^0 continuous (Section 2.2).

3. Natural-neighbor interpolation, where the interpolating surface is C^1 continuous except at the sample points (Section 2.3).

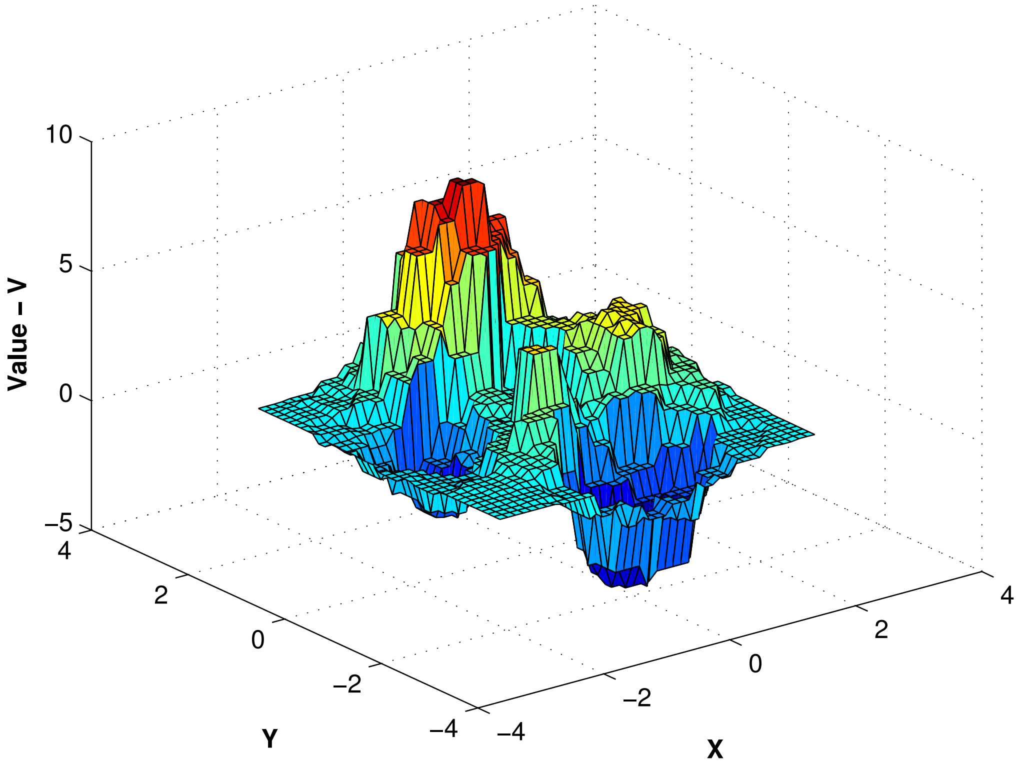

Figure 2.2: An example of the nearest neighbor interpolation of the points presented in Figure 2.1. The method does not generate any new data values (the vertical coordinate of the points in the figure) and therefore results in very rough transitions among data points.

2.1 Nearest Neighbor Interpolation

The nearest neighbor interpolation sets the value of an interpolated point to the value of the nearest data point. Therefore, this method does not generate any new data value. This results in very rough and abrupt transitions among data points, as can be seen in Figure 2.2. The input dataset used in this interpolation case is presented in Figure 2.1, where the location of the points is defined by the XY-plane and the corresponding values are represented by the V-axis.

2.2 Linear Interpolation

The linear interpolation fits a different linear polynomial between each pair of data points for curves, or between sets of three points for surfaces. An example of the application

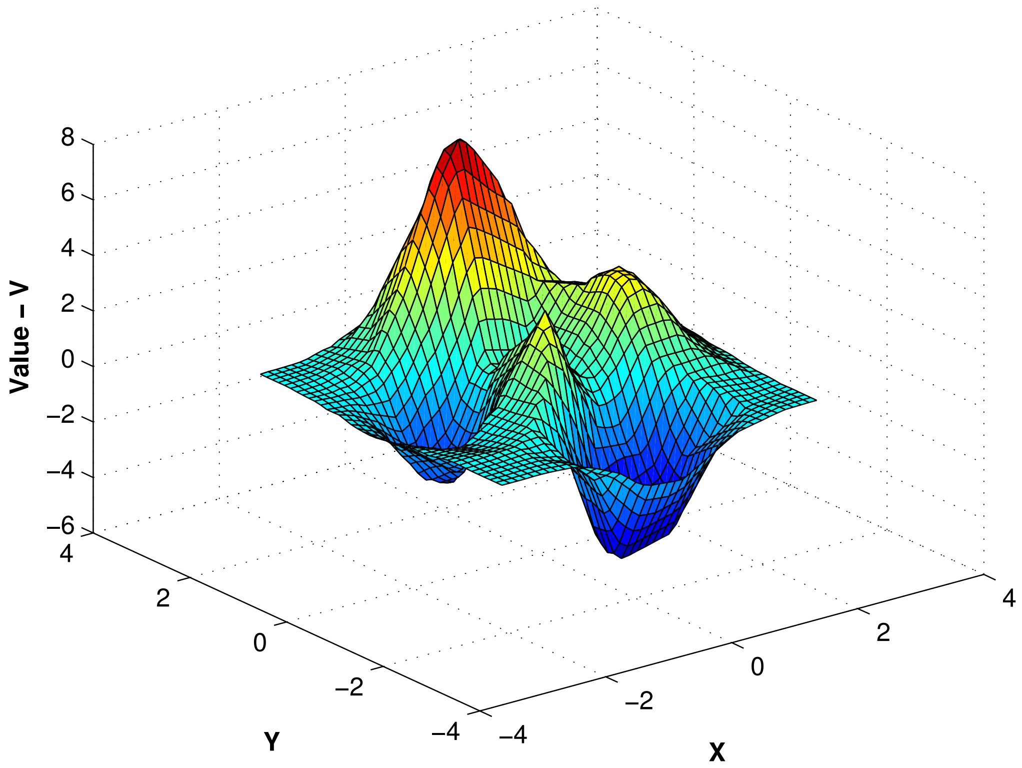

Figure 2.3: An example of the linear interpolation method of the points presented in Figure 2.1. Notice how the transitions are not so abrupt compared to the nearest neighbor approach in Figure 2.2.

of the linear interpolation method on the set of input points presented in Figure 2.1 is shown in Figure 2.3.

Linear interpolation uses *barycentric coordinates* since it provides a convenient way to interpolate a function on an unstructured grid or mesh, as long as the function's value is known at all vertices of the mesh. To interpolate a function f at a point r, it is necessary to loop through each triangular element of the mesh and transform r into the barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$ on that triangle. If $0 \leq \lambda_i \leq 1 \;\; \forall\, i \in \{1, 2, 3\}$, then the point lies in the triangle or on its edge. Also, the linear interpolation is automatically normalized since $\lambda_1 + \lambda_2 + \lambda_3 = 1$. By using those properties, the interpolated value of $f(r)$ is given by:

$$g(r) = \lambda_1 f(r_1) + \lambda_2 f(r_2) + \lambda_3 f(r_3) \tag{2.1}$$

In the context of facial depth images, the function g returns the interpolated depth value for a given location r.

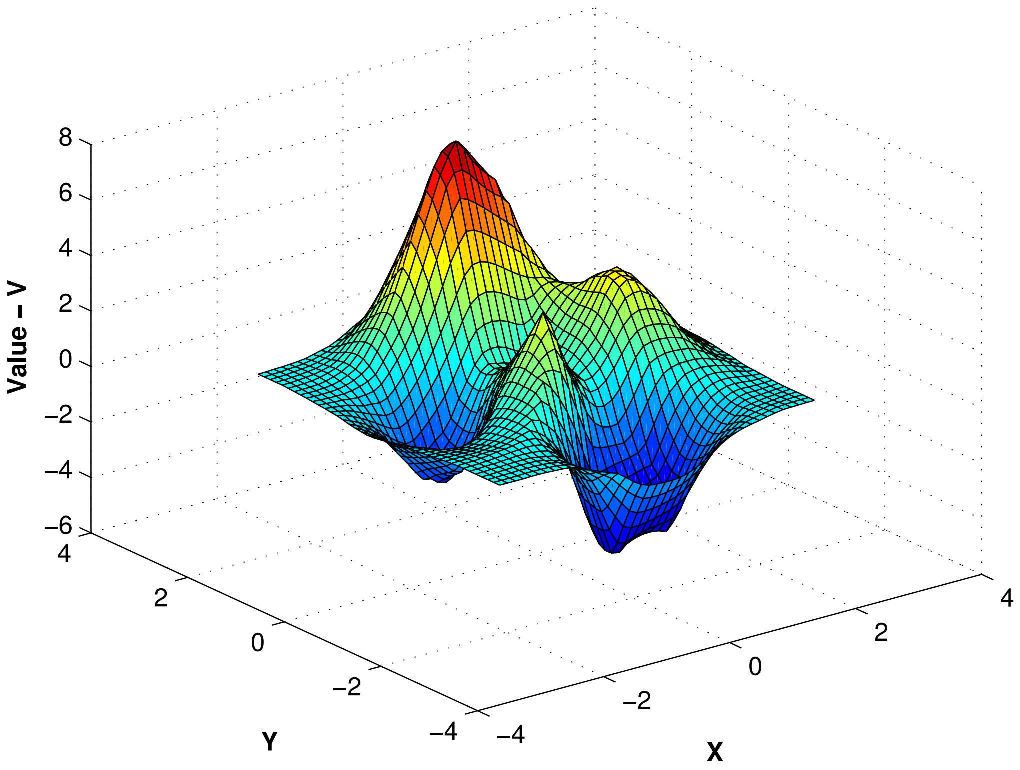

Figure 2.4: An example of the natural neighbor interpolation of the points presented in Figure 2.1. This method provides smoother approximations than simpler methods, such as the nearest neighbor (Figure 2.2) and linear interpolation (Figure 2.3).

2.3 Natural Neighbor Interpolation

The natural neighbor interpolation was proposed by Sibson [Sibson 1981]. To understand the idea behind natural neighbor interpolation, it is first required to be familiar with the Voronoi diagram [Voronoi 1908] and its related topics. More details in these topics are presented in [Preparata and Shamos 1985].

The natural neighbor interpolation is based on the Voronoi diagram of a discrete set of spatial points. The basic equation in 2D is given by:

$$g(x,y) = \sum_{i=1}^{n} w_i f(x_i, y_i), \tag{2.2}$$

where $g(x,y)$ is the estimate at (x,y), w_i are the weights and $f(x_i,y_i)$ are the known data at (x,y). The weights, w_i, are calculated by finding how much of each of the surrounding areas (*i.e.*, the Voronoi cells) is diverted when inserting (x,y) into the tessellation.

An illustrative example of the natural neighbor interpolation applied to the points in Figure 2.1 is shown in Figure 2.4. The results are smoother than the nearest neighbor but in this particular example they are very similar to those obtained with the linear interpolation method.

2.4 Discussion

The MATLAB® software [MATLAB 2009b] contains a class called **TriScatteredInterp** specifically created to perform scattered data interpolation based on an underlying Delaunay triangulation. It provides implementation for the three interpolation methods described in Sections 2.1, 2.2 and 2.3. It is believed that this is one of the reasons that has motivated computer vision researchers to use these simpler interpolation techniques in their work.

The class **TriScatteredInterp** has the following advantages [MATLAB 2009a]:

1. It produces an interpolating function that can be queried efficiently. That is, the underlying triangulation is created once and reused for subsequent queries.
2. The interpolation method can be changed independently of the triangulation.
3. The values at the data points can be changed independently of the triangulation.
4. Data points can be incrementally added to the existing interpolant without triggering a complete recomputation. Data points can also be removed and moved efficiently, provided that the number of points edited is small relative to the total number of sample points.

It is important to note that the previous illustrative examples (Figures 2.2, 2.3 and 2.4) dealt with the interpolation of point sets that were sampled on smooth surfaces. In addition, the points were relatively uniformly spaced. For example, clusters of points were not separated by relatively large distances. Also, the interpolant was evaluated well within the convex hull of the point locations. However, when dealing with real-world interpolation problems the data may be more challenging. It may come from measuring equipment that is likely to produce inaccurate readings or outliers. The underlying data may not vary smoothly and the values may jump abruptly from point to point.

To illustrate how these interpolation methods may be applied to a certain problem, it is important to mention the work of Wu et al. [Wu et al. 2011, Wu et al. 2010].

In [Wu et al. 2011], the authors use facial shape information to construct discriminating models for gender classification. They represent facial shapes using 2.5-dimensional fields of facial surface normals, and propose three different methods to improve the gender discriminating capacity of the model constructed using the standard eigenspace method. The three methods are variants of Principal Geodesic Analysis (PGA), presented in [Wu et al. 2010]. These three models are: (a) weighted PGA, (b) supervised weighted PGA, and (c) supervised PGA. Before applying the classification methods, each subject's facial surface normals (also called *needle-map*) need to be aligned to the needle-maps of other subjects and any discontinuities in the depth data have to be interpolated. The authors describe how this alignment is performed in [Wu et al. 2010] and they report that they used linear interpolation to fill missing data.

It is expected that a more robust interpolation method will provide an interpolant function that is less prone to errors. With that in mind, this work aims to investigate the impact of using four different interpolation methods with two different gender-classification methods and compare the outcome results.

Chapter 4 describes the use of the PGA and supervised weighted PGA models to measure how different interpolation methods can affect the gender classification accuracy. The classification methods are applied using linear, nearest neighbor, natural neighbor interpolation and a more sophisticated interpolation method, that will be presented in Chapter 3, to fill the holes in the needle-maps.

Chapter 3

The Proposed Alignment and Reconstruction Approach

The computation of the aligned facial depth image is comprised of *four main steps*. The first step consists in taking the input raw depth image and crop the rectangular region that contains the face of the imaged person (Section 3.1). Then, the approach adaptively subdivides the cropped region into smaller regions where Thin-Plate Splines (TPSs) are adjusted to the depth data in each of them (Section 3.2). The TPSs not only guarantee smooth interpolation of depth data while producing the final image but also provide the reconstruction of damaged and missing portions of depth information. In the third step, affine transformations are used in order to perform the 3-dimensional alignment of the landmarks of the given face with the landmarks of the face of an average person (Section 3.3). By applying the same transformations to the TPSs, one maps the input face to a standard space in which, in the last step, it is held ray casting in order to produce the final aligned facial image having arbitrary resolution (Section 3.4).

Section 3.5 describes how one can obtain the parameters of the average person from the particular dataset used in this project. Section 3.6 discusses how the approach can be extended to produce color images that are consistent with the resulting depth images.

3.1 Image Cropping

Four facial landmarks (namely, *left eye* (*le*), *right eye* (*re*), *nose* (*no*) and *chin* (*ch*)) are used to define the boundaries of the imaged face region. The axis-aligned cropping rectangle that contains the face is defined by these boundaries. More specifically, the location and the width and height of the cropping rectangle are computed, respectively,

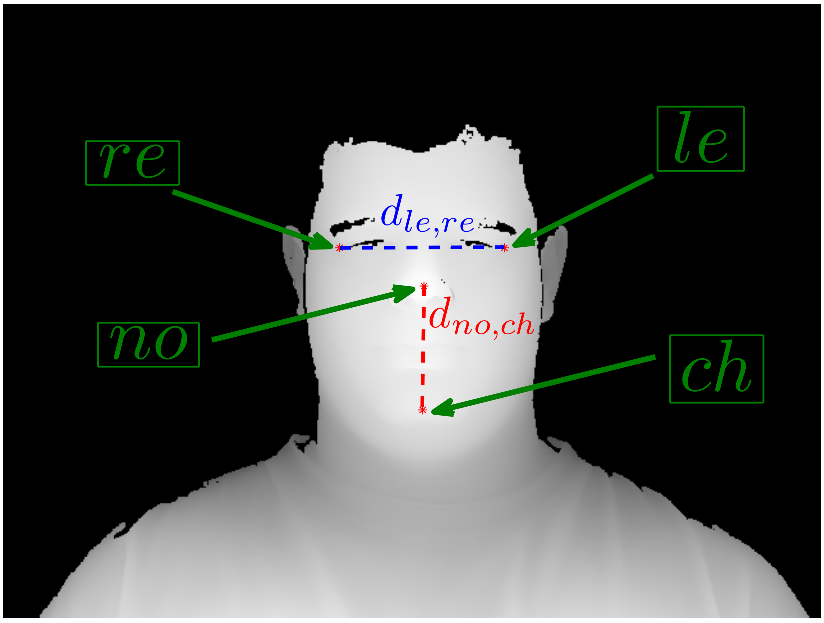

Figure 3.1: A grayscale visualization of an original depth image (640 × 480), before any cropping is done. The distances $d_{le,re}$ and $d_{no,ch}$ define the lower and upper corners of the cropping rectangle. Lighter shades of gray correspond to points closer to the camera.

from the image coordinates $(u_{no}, v_{no})^T$ of the nose as pivot point, the horizontal distance $d_{le,re}$ between imaged eyes and the vertical distance $d_{no,ch}$ between imaged nose and chin. Figure 3.1 shows a sample depth image where the measured distances $d_{le,re}$ and $d_{no,ch}$ have been highlighted. The lower and upper corners of the cropping rectangle are expressed as:

$$\boldsymbol{C}_- = \begin{pmatrix} \max\left(u_{no} - u_\Delta, 1\right) \\ \max\left(v_{no} - v_\Delta, 1\right) \end{pmatrix} \quad \text{and} \quad \boldsymbol{C}_+ = \begin{pmatrix} \min\left(u_{no} + u_\Delta, w_{ac}\right) \\ \min\left(v_{no} + v_\Delta, h_{ac}\right) \end{pmatrix}, \tag{3.1}$$

where w_{ac} and h_{ac} are, respectively, the width and height of the input image having pixel coordinates $u \in [1, w_{ac}]$ and $v \in [1, h_{ac}]$. In Equation 3.1, $u_\Delta = \min\left(\lceil 1.5\, d_{le,re}/2 \rceil, w_{ac} - u_{no}\right)$ and $v_\Delta = \min\left(\lceil 1.5\, d_{no,ch} \rceil, h_{ac} - v_{no}\right)$, and $\lceil . \rceil$ denotes the ceiling function. The proportion value 1.5 was empirically chosen from the experiments. An example of a cropped image is shown in Figure 3.2.

It is important to emphasize that defining a sub-image using a cropping rectangle is an optional step of the approach. By limiting data to the sub-image that contains the region of interest (*i.e.*, the subject's face) the computational cost of subsequent steps of the proposed algorithm is reduced. Furthermore, the cropped image does not have to be perfectly symmetrical to the imaged face because the actual alignment procedure will

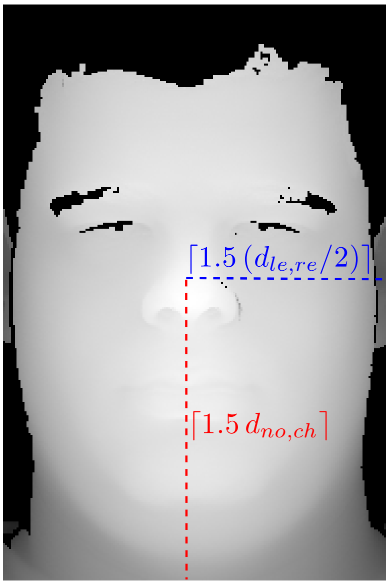
$\lceil 1.5\,(d_{le,re}/2)\rceil$
$\lceil 1.5\,d_{no,ch}\rceil$

Figure 3.2: An example of a cropped image (195 × 293). The *nose* is located at the center. The width and height of the final cropped image depend on u_Δ and v_Δ, respectively.

be performed by the final steps of the algorithm (see Sections 3.3 and 3.4). Also, the produced cropping rectangles may have different resolutions, since their dimensions are proportional to distances in a given input image. The only requirement for a sub-image is to contain the subject's face. Such a condition must be asserted by the user. During the experiments, the proposed proportion value (1.5) showed adequate for all testing cases as far as the landmarks were well positioned. The images having misplaced landmarks were removed from the original dataset, as described in Appendix A.

The location of the nose and chin is usually provided as part of image databases (*e.g.,* the UND Biometric Dataset, Collection D [Chang et al. 2003]). They can also be retrieved by automatic techniques [Romero-Huertas and Pears 2008, Perakis et al. 2009] or manually identified. The development of a technique for detecting landmarks is out of scope of this work.

3.2 Depth Data Interpolation and Reconstruction

Depth data interpolation and reconstruction is performed using TPSs adjusted to raw depth data. TPS is the 2-dimensional analog of the cubic spline in one dimension [Bookstein 1989]. It encodes a scalar height function that can be evaluated at a given $(u, v)^T$ coordinate in order to retrieve the respective scalar value z that best describes the height surface passing through N non-overlapping control points having coordinates $(u, v, z)^T$. In this paper, u and v are the coordinates of a valid pixel (*i.e.,* pixels storing valid depth information), and z is the associated depth value. It is important to notice that a TPS is adjusted to an unstructured set of control points, and also that it can be evaluated at any real $(u, v)^T$ position, returning a smoothly interpolated z value. Thus, it is clear that sub-pixel sampling and damaged facial depth reconstruction are naturally handled by the TPS-based interpolation scheme adopted in this work.

A TPS is described by $2\,(N + 3)$ parameters, which include six global affine motion parameters and $2\,N$ coefficients for correspondences of the control points. These parameters are computed by solving a linear system having a closed-form solution. Due to the large number of parameters, the computation of a single TPS to all valid pixels in a cropped image may be unfeasible. This happens because during the TPS kernel processing, there is an intermediate calculation where a matrix, having size proportional to the number of control points, is multiplied by itself [Bookstein 1989]. The necessary memory to store the intermediate data might not be sufficient. We avoid such an issue by dividing the cropped image into adaptive blocks having a small number of control points, and fit a different TPS to each one of the blocks. Such an approach has two advantages: (i) it allows our technique to handle images having arbitrary size; and (ii) the procedure is less prone to numerical instability.

The adaptive blocks are initially distributed uniformly over the cropped image as a regular grid comprised by square entries having fixed size. However, since depth data may be damaged, some of the blocks may not contain enough control points to define a TPS. In such a case, we incrementally change the size of ill-defined blocks by including a ring of surrounding pixels to them. An ill-defined block grows until it has enough valid pixels to solve the linear system of equations that computes the coefficients of the TPS.

In a general manner, larger blocks are better since they cover a larger area and have more valid points to interpolate with. Still, there has to be a trade-off between the block size and the time necessary to compute the TPS. In this work, the chosen size of the adaptive blocks where TPSs are adjusted is 32×32 since they are large enough to cover the largest holes in the images (typically in the eyes) and could be processed in feasible time.

The total number of blocks vary from image to image since the size of the cropped region also varies (Section 3.1). The vertical block size (b_h) and the horizontal block size (b_w) can also be different from each other, which means that the blocks do not necessarily have to be squares. Also, if the image size is not a multiple of the block size then the

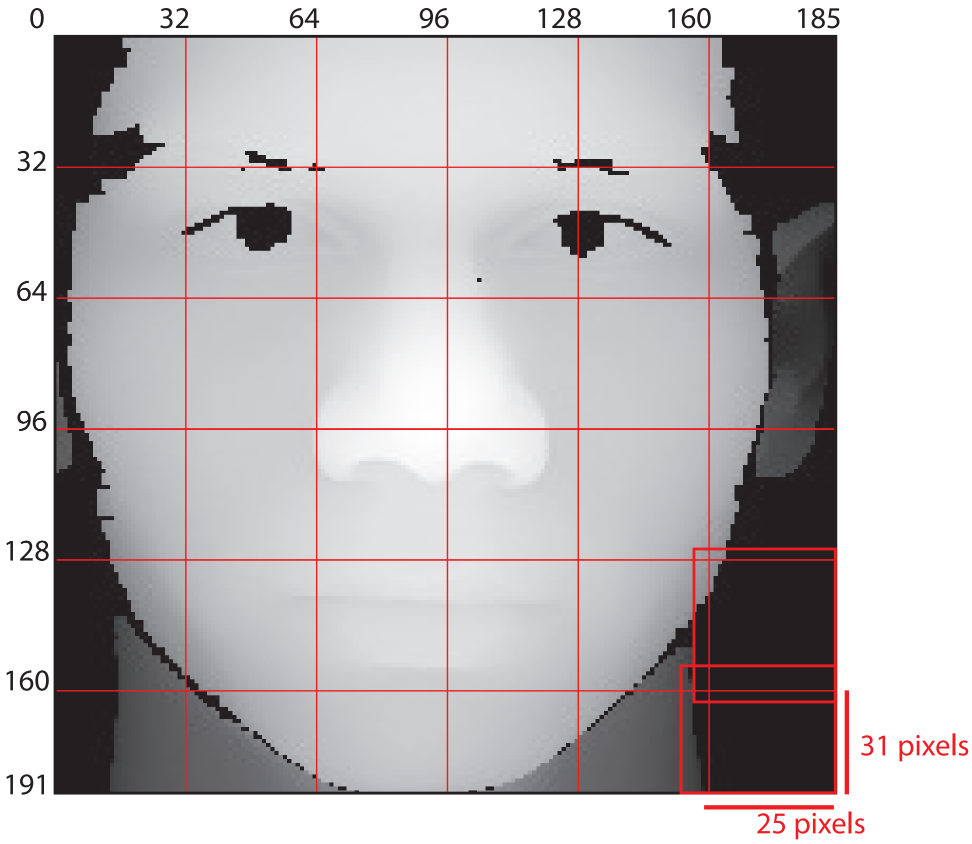

Figure 3.3: An example of how the last row and last column blocks may be smaller in size than the rest of the blocks in the image. In the proposed algorithm, each block may independently grow in size until enough control points are within its boundaries (see the blocks in the lower right corner of the image).

last row and column blocks are likely to be comprised of smaller blocks, as can be seen in Figure 3.3. For instance, recall from Section 3.1 that the width of the current image is w_{ac} and the height is h_{ac}. Suppose that m is the largest possible multiple of the block size, being equal or less than either w_{ac} or h_{ac}. Then, the width of the last column is given by $b_w = w_{ac} - m_w$ and the height of the last row is given by $b_h = h_{ac} - m_h$.

The maximum absolute error within each block was measured in order to assess the error that the block division approach could be introducing. When comparing the original valid points and the ones computed with the TPS, the maximum absolute error found was smaller than 10^{-7} mm.

In literature, the TPS has been used within face images. In [Guo et al. 2004], the authors describe a face reconstruction technique based on an average morphable shape represented by a TPS. In [Whitbeck and Guo 2006], TPSs are used on the interpolation of colored images. As a comparison, this work uses TPSs on single depth images.

3.3 Three-Dimensional Face Alignment

The alignment stage computes the affine transformation that maps a face defined in the *actual space* (*i.e.,* the 3-dimensional space where the imaged face resides) to a standard position and orientation defined in the *standard space* (*i.e.,* the 3-dimensional space where all faces will be aligned).

The transformation of a given imaged face is computed from the location of four landmarks in the actual coordinate frame (*i.e., left eye* (*le*), *right eye* (*re*), *nose* (*no*) and *chin* (*ch*)) and the equivalent locations in the standard coordinate frame. In the following equations, each location is represented by a point $\boldsymbol{P}_{\mathcal{S},\mathcal{F}}$, where $\mathcal{S}$ is *ac* or *st* for, respectively, actual or standard frames, and $\mathcal{F}$ is one of the labels in $\{le, re, no, ch\}$.

The coordinates of $\boldsymbol{P}_{ac,\mathcal{F}}$ are computed from the input depth image as:

$$\boldsymbol{P}_{ac,\mathcal{F}} = \begin{pmatrix} x_{ac,\mathcal{F}} \\ y_{ac,\mathcal{F}} \\ z_{ac,\mathcal{F}} \end{pmatrix} = z_{ac,\mathcal{F}} \left(\mathrm{K}^{-1} \boldsymbol{Q}_{ac,\mathcal{F}} \right), \tag{3.2}$$

where $\boldsymbol{Q}_{ac,\mathcal{F}} = (u_{\mathcal{F}}, v_{\mathcal{F}}, 1)^T$ is the location (in pixels) of the given label point in image space, $z_{ac,\mathcal{F}}$ is the depth retrieved from the $(u_{\mathcal{F}}, v_{\mathcal{F}})^T$ pixel, and K^{-1} is the inverse of the matrix that models the intrinsic camera parameters [Hartley and Zisserman 2000]:

$$\mathrm{K} = \begin{pmatrix} f\, m_u & \gamma & o_u \\ 0 & f\, m_v & o_v \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.3}$$

where f is the focal length, m_u and m_v are the scale factors relating pixels to distance, γ, o_u and o_v represent the skew and the coordinates of the principal point, respectively. The values of the aforementioned parameters were retrieved from the dataset.

The intrinsic parameters are usually provided by the depth camera, but they can also be computed by calibration procedures [Hartley and Zisserman 2000]. The retrieval process of these parameters from the dataset is described in Appendix A.

The formulas for computing the 3 × 3 matrix M and the 3 × 1 offset vector O modeling the intended affine transformation that maps from the actual space to the standard space is given by:

$$\mathrm{M} = \mathrm{Q}\,\mathrm{P}^{-1} \quad \text{and} \quad \mathrm{O} = \boldsymbol{P}_{st,le} - \mathrm{M}\,\boldsymbol{P}_{ac,le}, \tag{3.4}$$

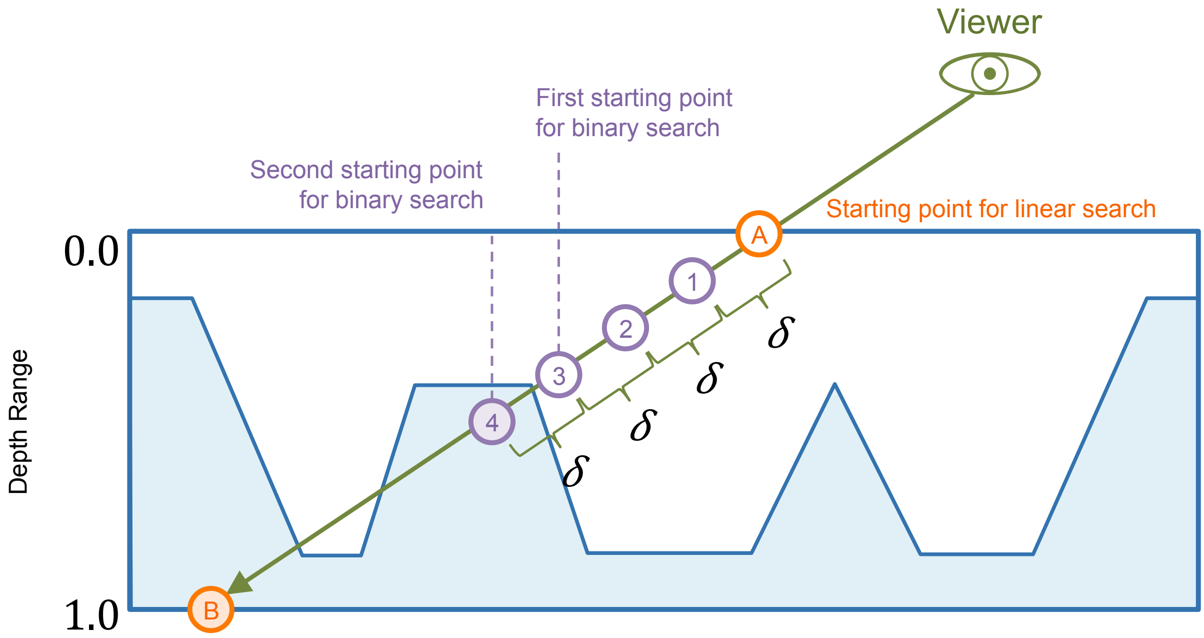

Figure 3.4: An illustration of a ray being cast into the surface direction. A point is incremented by δ in the ray direction until the point has penetrated the surface (4).

where P and Q are 3 × 3 matrices computed as:

$$\mathrm{P} = \begin{pmatrix} \boldsymbol{P}_{ac,re} - \boldsymbol{P}_{ac,le} & \boldsymbol{P}_{ac,no} - \boldsymbol{P}_{ac,le} & \boldsymbol{P}_{ac,ch} - \boldsymbol{P}_{ac,le} \end{pmatrix}, \tag{3.5}$$

$$\mathrm{Q} = \begin{pmatrix} \boldsymbol{P}_{st,re} - \boldsymbol{P}_{st,le} & \boldsymbol{P}_{st,no} - \boldsymbol{P}_{st,le} & \boldsymbol{P}_{st,ch} - \boldsymbol{P}_{st,le} \end{pmatrix}. \tag{3.6}$$

The procedure used to compute the coordinates of the landmarks in the standard space (*i.e.*, points $\boldsymbol{P}_{st,\mathcal{F}}$) is presented in Section 3.5.

Once M and O are known, the affine mapping of a general point $\boldsymbol{P}_{ac}$ in the actual space to the standard space is given by:

$$\boldsymbol{P}_{st} = \mathrm{M}\,\boldsymbol{P}_{ac} + \mathrm{O}. \tag{3.7}$$

3.4 Producing the Final Depth Image

The final depth image of a given face is computed by casting rays (one ray per resulting image pixel) from a pinhole camera defined in the standard space to the surface of the subject's face mapped from the actual coordinate frame to the standard coordinate frame (Section 3.3). The procedure is derived from a well-known rendering technique, the Relief Mapping (RM), which is extensively described in [Policarpo et al. 2005].

The central idea of the ray casting procedure is to use RM to quickly find the first intersection of each casted ray with the surface encoded by the set of TPSs (Section 3.2). Figure 3.4 illustrates a generic surface where the ray source is positioned at the viewer position. The depth range defines where the linear search should be applied. It can vary depending on the context and was set to $A = 0.0$ and $B = 1.0$ to keep it simple. Once the first intersection is found for a given ray, the z coordinate of the intersection point in camera's coordinate system is stored in its respective image pixel.

As in RM, the process starts with a linear search. Beginning at the center of projection $\boldsymbol{O}$ (indicated in Figure 3.4 as the *Viewer*), the process steps along the ray passing through the current pixel mapped to the image plane in 3-dimensional space at increments of $\delta = 1mm$ looking for the first point inside the surface. Once the first point under the TPS surface has been identified, the binary search starts using the last point outside the surface and the current one. The binary search will stop if the difference between the current and last approximation is smaller than $2,0 \times 10^{-1}\ mm$. The role of the linear search is to quickly approximate the first intersection between the casted ray and the TPS surface. The role of the binary search, on the other hand, is to find the exact location of such an intersection.

The ray casting procedure is performed in the actual coordinate frame. This is because depth information encoded by the TPS is defined in such a space and cannot be transformed to the standard space with the guarantee that it will be an unambiguous depth map after such a mapping. Thus, one has to (i) map the whole situation of the ray casting procedure (*i.e.,* the camera's center of projection $\boldsymbol{O}_{st}$ and pixels' points $\boldsymbol{Q}_{st}$ in the image plane) from the standard space to actual space using the inverse affine mapping:

$$\boldsymbol{P}_{ac} = \mathrm{M}^{-1}\ (\boldsymbol{P}_{st} - \mathrm{O}), \tag{3.8}$$

(ii) find the first intersection point using the proposed RM-based ray casting procedure, (iii) map the intersection point back to the standard space using Equation 3.7, and (iv) compute the final depth value as the signed distance between $\boldsymbol{O}_{st}$ and the intersection point.

In the UND Biometric Dataset, Collection D [Chang et al. 2003], there are some range images that contain noisy depth values identified as valid ones, especially around the eyes region. Those values can be either inside or outside the expected imaged face region and they can affect the interpolation results. When a noisy value is outside the expected surface of the imaged face, such as in the example highlighted by the red circle

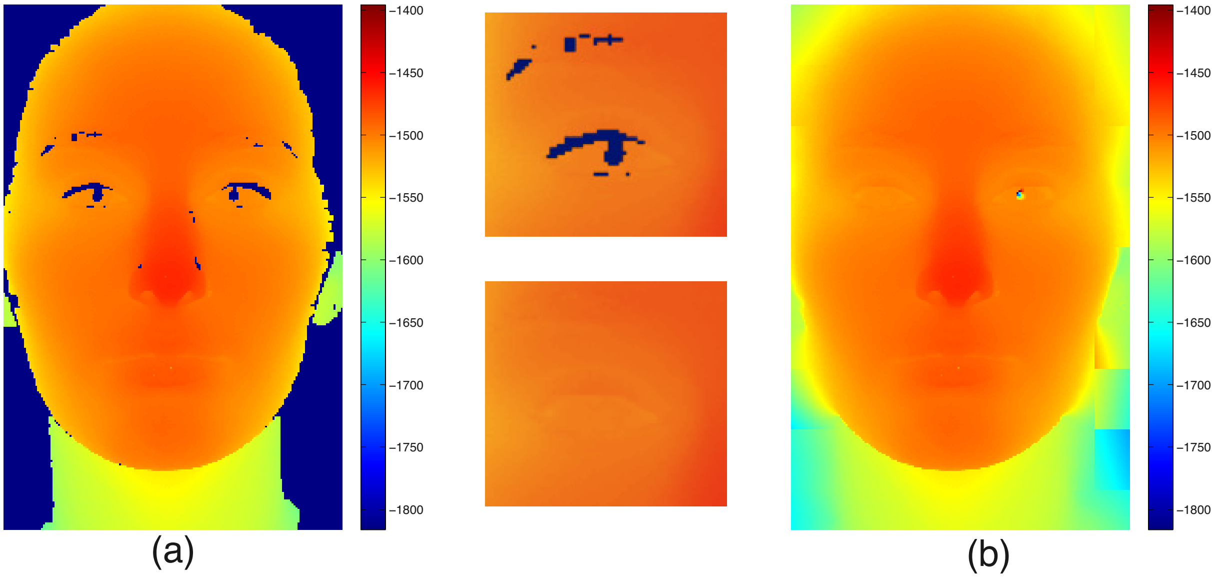

Figure 3.5: Resulting aligned depth image before (a) and after (b) the TPS interpolation. Notice how the missing depth information is smoothly reconstructed. In some images, like in these particular examples, the depth data may include incorrect points marked as valid ones, especially around the eyes. In such a case, the interpolation results in a disturbed (noised) region.

in Figure 3.6, it can negatively affect the RM technique. This is because the TPS is evaluated at every increment of δ during the linear search of the RM, and the resulting value is checked in order to determine whether or not the casted ray crossed the surface. When one of the rays is casted in the direction of one of these invalid points, the linear search stops prematurely at an illegal intersection point and the z coordinate returned by the TPS is mapped back to the standard space. As a consequence, every point that shares this particular direction receives an incorrect depth value and, as a collateral effect, the final image ends up with a projected straight line pattern (*i.e.,* a shadow), as can be seen in Figure 3.7b. However, when the noisy values are inside the imaged face, such as in the case highlighted by the green circle in Figure 3.6, the ray casting process is not greatly affected, as can be seen in the example shown in Figure 3.5.

The proposed approach presented in this section is able to smoothly reconstruct the missing portions of data while evaluating the surface encoded by the TPS. At the same time, each image is aligned through an affine transformation from the actual space to the standard space. The whole process is performed in a completely automatic way.

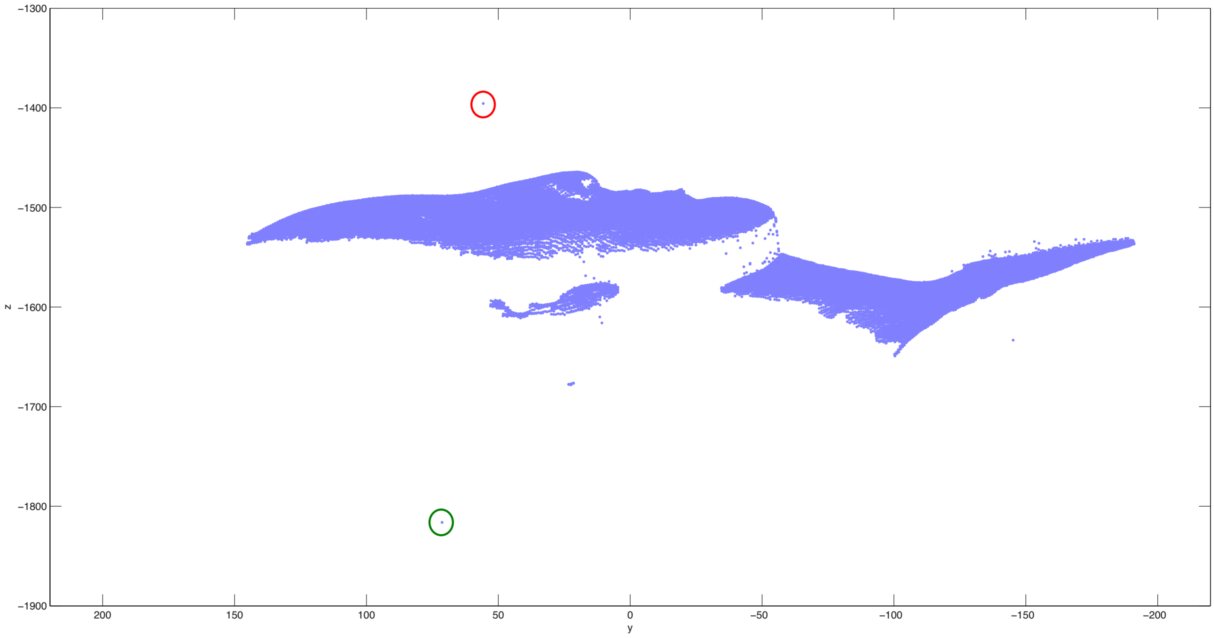

Figure 3.6: Depth data viewed from the side. Note that some noisy points having erroneous depth values (highlighted by the circles) may be marked as valid by the scanner.

3.5 Computing the Average Person

The average person is defined in the standard space to which all faces are mapped during the face-alignment stage of the procedure (Section 3.3).

In order to setup an average person one needs to specify the location $\boldsymbol{P}_{st,\mathcal{F}}$ of the four face landmarks. In the experiments, the coordinates of $\boldsymbol{P}_{st,\mathcal{F}}$ were computed from average values retrieved from the input dataset. However, it is important to notice that one can place the landmark in the way that is most convenient for a particular application.

A mean tetrahedron is built from the tetrahedra defined by the landmarks of each input face. The base of such tetrahedron is defined by the location of both *eyes* and *chin*. The apex is set to be the *nose*. The vertices are computed as:

$$\boldsymbol{P}_{st,le} = \frac{1}{2}\begin{pmatrix} w_{st} + \mu_{eye} \\ h_{st} - \mu_{chin} \\ 0 \end{pmatrix}, \qquad \boldsymbol{P}_{st,re} = \frac{1}{2}\begin{pmatrix} w_{st} - \mu_{eye} \\ h_{st} - \mu_{chin} \\ 0 \end{pmatrix},$$

$$\boldsymbol{P}_{st,ch} = \frac{1}{2}\begin{pmatrix} w_{st} \\ h_{st} + \mu_{chin} \\ 0 \end{pmatrix}, \qquad \boldsymbol{P}_{st,no} = \frac{1}{2}\begin{pmatrix} w_{st} \\ h_{st} - \mu_{chin} + 2\,\mu_{nose} \\ 2\,\mu_{tip} \end{pmatrix},$$

where w_{st} and h_{st} are, respectively, the width and height of the (standard) resulting image,

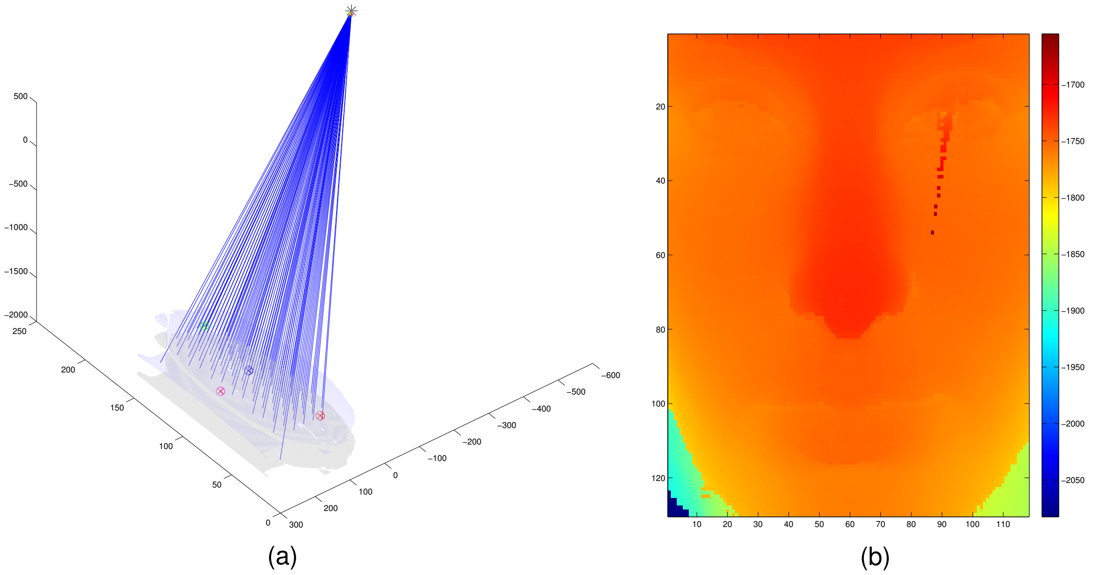

Figure 3.7: The Relief Mapping(RM) technique: (a) rays casted from the center of projection of the pinhole camera in standard space (one ray per resulting image pixel), (b) the resulting image after the sampling process. In this example it is shown a case where a pattern is produced in the resulting image due to noise in the captured depth data.

μ_{eye} is the mean distance between the eyes of faces in the dataset, μ_{chin} denotes the mean distance from the chin to the middle of the eyes, μ_{nose} is the mean distance from the orthogonal projection of the nose onto the base plane and the middle of the eyes, and μ_{tip} denotes the mean distance from the nose tip to the base plane. All those distances were measured in the actual 3-dimensional coordinate frame where each input face resides.

3.6 Producing Correct Color Images

The computation of correct color images to be used with the computed (aligned) depth images is straightforward. While performing the ray casting procedure described in Section 3.4, the color information related to the first intersection between the casted ray and the surface encoded by the TPS can be retrieved from another TPS encoding the RGB color of input image pixels. By doing so, one guarantees a smooth interpolation of color values as well as the reconstruction of missing portions of color information.

3.7 Discussion

The following subsections contain implementations remarks and relevant technical aspects of the proposed approach, described in this chapter.

Table 3.1: Execution times for different block sizes.

Block size	Time for 1 image (seconds)	Time for all images (hours)
32 × 32	97	26
30 × 30	70	19
28 × 28	48	12
25 × 25	30	8
16 × 16	2	0.5

3.7.1 Block division and parallel processing

The TPS implementation used in this work was written in C++ and it was compiled using Microsoft® Visual Studio® as Dynamic Link Libraries (DLLs) in order to be called as MEX functions (*i.e.,* an interface between MATLAB® and subroutines written in C, C++ or Fortran) [MATLAB 2009a].

In the first experiments, when trying to pass the whole image (640 × 480) as input to the TPS, the available memory was not enough (see Section 4.1 for a detailed description of the hardware settings used in this work). Thus, it was mainly due to this memory limitation issue that the cropping step described in Section 3.1 was developed.

Later, a cropped version of the image matrix with a testing size of 200 × 200 was used. Although the physical memory was enough this time, its consumption was over 15GB. In such trial, the process ran for several hours without completion. Since there were 953 images in the dataset, it was crucial that all images could be processed in feasible time. Therefore, each image was divided into small blocks and a TPS was fit to each block (Section 3.2). Even with small parts of the image being separately calculated, it was still possible to reconstruct the image as a whole.

With the blocks division, OpenMP was used to explore parallel computing in the computation of the TPSs' coefficients, by processing each block in a different core, the processor usage was increased from an average of 12% of its capacity to constant 100% use of its capacity. The execution times for different block sizes were measured and the results are shown in Table 3.1.

3.7.2 Reducing computational costs

After computing the coefficients, the TPS interpolant that was written for this work evaluates the z coordinates within the surface domain (x, y). During the RM procedure, a call to this function must be done after each increment step. Unfortunately, MATLAB® imposes some overhead for calling external MEX functions and, since there will be many calls for each ray and the number of rays is proportional to the final image size ($width \times height$), it is important to keep the interpolant function calls to a minimum. There are many ways to reduce the number of calls: (a) decrease the final image size, (b) increase the value of the linear search increment δ, and (c) make sure the current point is within the TPS's valid domain. In this work, all three approaches were used. It is believed that the linear search increment δ could be be a little higher than the value set during the tests. However, no experiments were run to test this hypothesis.

It is also important to mention that Graphics Processing Units (GPU) could be used to speed up the performance of the proposed method. As it was shown, the proposed approach runs in separate threads without any communication between jobs, hence it is parallelizable.

Chapter 4

Experiments and Results

Several tests were performed in order to evaluate the proposed approach. At first, the data was processed following the methodology described in Chapter 3. The preparation details are given in Section 4.1. Later, the proposed method was compared against a commonly used approach based on 2-dimensional alignment with linear interpolation (Section 4.2). In Section 4.3, the gender classification models presented in the work of [Wu et al. 2011] were applied to the UND dataset. Since the gender classification models receive surface normals of the facial depth images as input, the performance of these classification models indicate the influence and quality of each interpolation method. The results using the TPS were compared against other interpolation methods, described in Chapter 2. Lastly, the confusion matrix [Kohavi and Provost 1998] and the Matthews Correlation Coefficient (MCC) [Matthews 1975] were calculated using the k-fold cross-validation framework [Kohavi 1995]. The quality of the interpolation techniques is verified from the analysis of such metrics in Section 4.4.

4.1 Preparing the Data for the Experiments

The technique presented in this work was implemented using C++ and MATLAB®. The C++ code was compiled using Microsoft® Visual Studio®. OpenMP was used to explore parallel computing while adjusting TPSs to blocks of valid pixels, and while evaluating the TPSs for retrieving interpolated depth data.

The system was tested on several real depth images. The tests were applied to the UND Biometric Dataset (Collection D) [Chang et al. 2003]. This dataset is comprised by 953 images of 277 individuals. The computer used for testing was an Intel® Core™ i7-2600 CPU @ 3.40GHz, 16.0 GB RAM, AMD Radeon™ HD 6870 graphics processor,

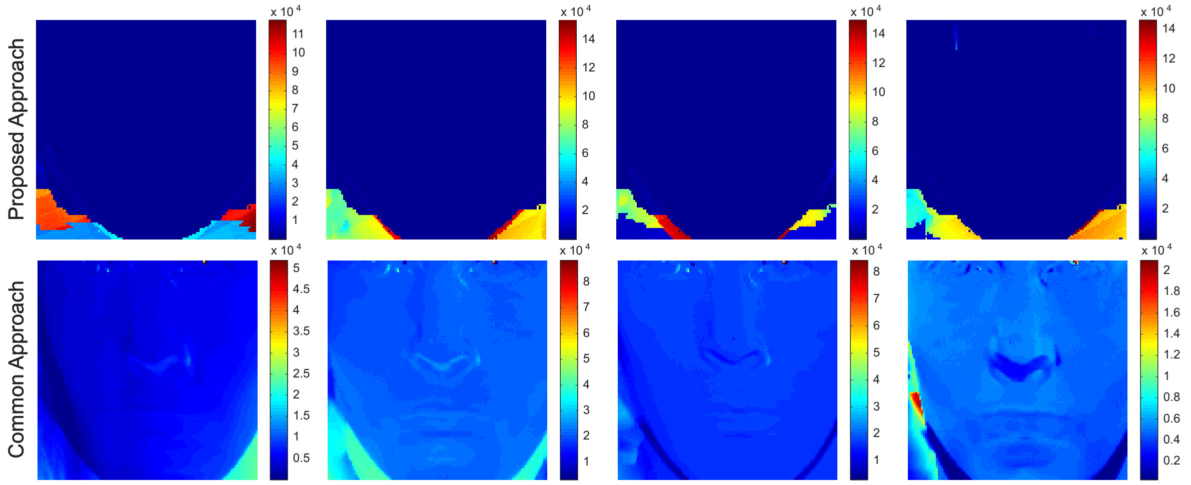

Figure 4.1: Color visualization of the squared error between images of the same subject: (top) Using the proposed method. (bottom) Using 2-dimensional alignment and Linear interpolation. Notice how the error is minimum (dark blue) at the central region of the face when using our method, while the regions having larger errors (dark red) are located on the neck, outside the face region.

Microsoft® Windows™ 7 64-bit.

The samples of the dataset were recorded using the Minolta Vivid series 3-dimensional scanner [Minolta 2003]. The UND dataset has the advantage that it contains the 2-dimensional color images, the corresponding range images, and the location of the facial landmarks in image space. See Appendix A for a detailed description of the UND dataset files and content.

In the experiments, the initial size of the adaptive blocks where TPSs are adjusted (Section 3.2) was set to 32 × 32. The final number of blocks depends on the size of the cropped image (Section 3.1). The resolution of the final images was set to $w_{st} = 118$ and $h_{st} = 130$. However, it is important to notice that our approach can produce smooth images having any resolution. The location of the facial landmarks in the input images were retrieved from the dataset. See Appendix A for a detailed description of how these landmarks can be retrieved.

The UND dataset was used to calculate the following mean values (expressed in millimeters) that define the average person (Section 3.5): $\mu_{eye} = 101.3238$, $\mu_{nose} = 40.0234$, $\mu_{chin} = 104.3238$, and $\mu_{tip} = 38.2279$. Notice that those values are consistent since they are proportional to the average subject's face. Hence, they could be used without change by any application where the proposed technique could be applied, on datasets having the same ethnic composition. The synthetic pinhole camera (Section 3.4) was placed 1750 mm

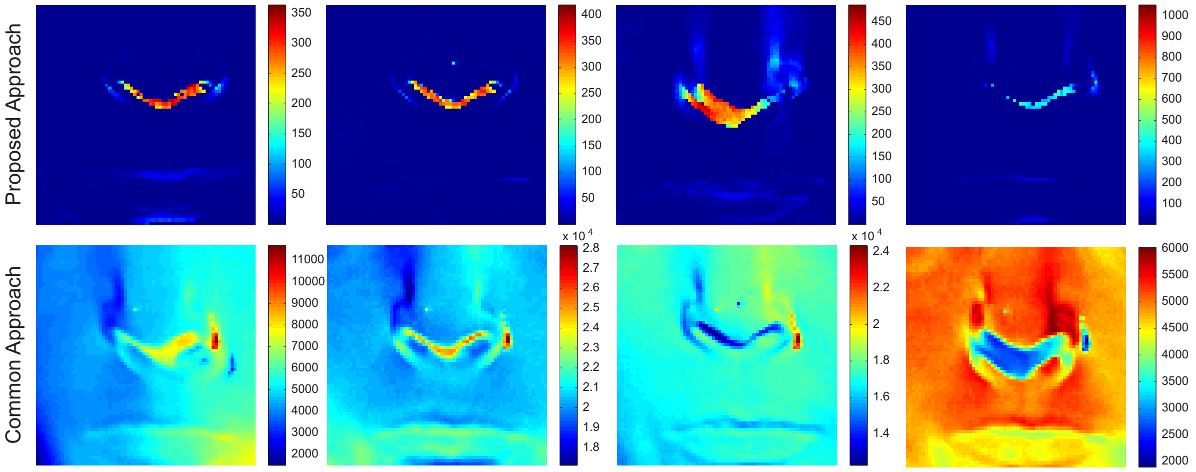

Figure 4.2: Color visualization of the squared error on the nose region. (top) Using our method: (bottom) Using 2-dimensional alignment and Linear interpolation. It is important to mention that the color bar scale is purposely different between the top and bottom rows in order to demonstrate where the errors in the proposed technique are located. Notice that, in the proposed approach, the maximum error values (hot colors) are much smaller than those presented in the common approach.

apart from the base plane of the mean tetrahedron, with optic axis coinciding with the displacement vector of the nose and having its x and y axes aligned, respectively, to the x and y axes of the standard coordinate frame.

4.2 Comparing to the Common Alignment Approach

In the first experiment, the results using the proposed method were compared to a widely used approach based on 2-dimensional alignment with linear interpolation. In this approach, the missing portions of depth information were first filled using the linear interpolation method provided by MATLAB®. In turn, it was applied the transformation matrix to align the triangle defined by the points $\boldsymbol{Q}_{ac,le}$, $\boldsymbol{Q}_{ac,re}$ and $\boldsymbol{Q}_{ac,no}$ to the same standard coordinates presented in Section 3.5. The sum of the square differences in the z values was calculated for both methods.

The UND dataset contains several depth images belonging to a same subject, varying from a single image up to 9 different recordings of the same volunteer. The larger time frame between the captures is four months. Within that time, a subject's face does not change much and it is expected that their alignment should match better than with other subjects' faces. That said, the comparison was done using different recordings of faces that belong to the same subject, taken at different times.

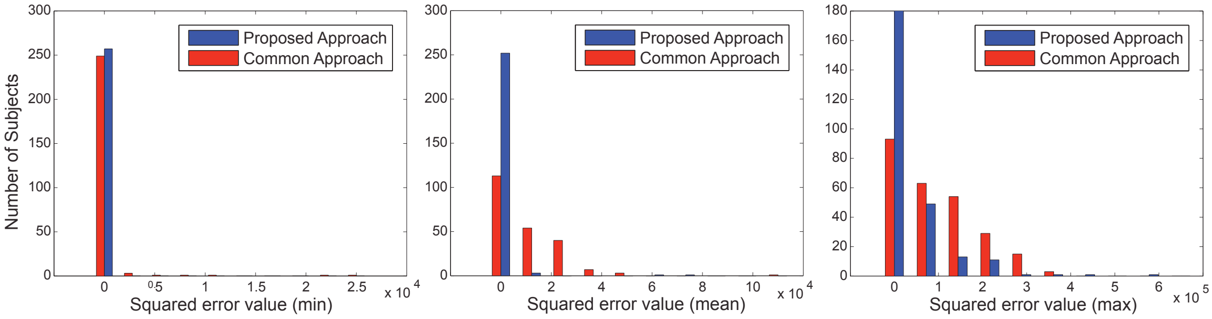

Figure 4.3: Histogram showing the distribution of minimum, mean and maximum squared error values found using the proposed (blue) and the common (red) approaches.

The analysis was performed using the recordings of 257 subjects. This is because the error measurement can only be computed for those individuals having two or more images. The mean squared error found using the proposed approach was 2.1319×10^2 mm^2 while the mean error using the common approach was 1.0135×10^4 mm^2. A histogram showing the minimum, mean and maximum error values computed for each subject is presented by Figure 4.3. It is important to comment that, in the proposed approach, the larger errors are gathered mostly on the neck region (outside the face). With the common 2-dimensional approach, the errors are scattered all over the imaged face, as shown in Figure 4.1. By analyzing just the nose region (Figure 4.2), the maximum squared error of the proposed method becomes 2 orders of magnitude smaller than the common approach.

4.3 Gender Classification

In their work, Wu *et al.* [Wu et al. 2011] use facial shape information to construct discriminating models for gender classification. They represent facial shapes using 2.5-dimensional fields of facial surface normals, and propose three different methods which are variants of Principal Geodesic Analysis (PGA).

PGA is a generalization of Principal Component Analysis (PCA) from data residing in a Euclidean space to data residing on a Riemannian manifold S^2 (*i.e.*, a Riemannian manifold (M, g) is a real differentiable manifold M in which each tangent space is equipped with an inner product g, a Riemannian metric, in a manner which allows a generalized study of distance and orthogonality). In other words, PGA is a generalization of PCA for points residing on the surface of a sphere in $\mathbb{R}^3$. Therefore, distances in this space are proportional to the arc between two points on that sphere. The goal of PCA is to locate a linear subspace of the space in which the data lies, and maximize the projected variance of the data. In PGA, the notion of linear subspace is replaced by that of a geodesic submanifold. The geodesics that traverse the submanifold are referred to as principal geodesics. They are analogous to the principal axes in PCA, except that each principal axis in PCA is a straight line. In the spherical case, a principal geodesic corresponds to a great circle. To project a point $n_1 \in S^2$ onto a great circle G is to find the point on G that is Nearest to n_1 in geodesic distance. Further details can be found in [Wu et al. 2010].

After request, the authors kindly provided the source code necessary to run the tests presented in this work. The main motivation to use a third-party code is that it represents an unbiased framework to evaluate the performance of the interpolation methods.

4.3.1 Calculating surface normals

Each image was processed following the steps presented in Chapter 3. Since the gender classification models use facial surface normals as input, it was first necessary to estimate the surface normals of each pixel associated to the processed images.

Recall from Chapter 2 that the interpolating surface of the Nearest neighbor method is discontinuous. The interpolating surface of the Linear method is C^0 continuous and there is also the Natural-neighbor interpolation, where the interpolating surface is C^1 continuous, except at the sample points. For comparison, the TPS has the advantage that its interpolating surface is C^∞, except at the sample points. This interesting characteristic motivated the use of the gender classification techniques presented by Wu *et al.* [Wu et al. 2011] since they use facial surface normals as input.

To calculate the normal vector for a certain pixel P, it is necessary to obtain the z depth of its immediate neighbors. Four z values are necessary: the value computed for the pixel to the left (P_l), the value for the pixel to the right (P_r), and also the z values for pixels relatively up (P_u) and down (P_d) in the image. From these values, the tangent vectors (V) for the point projected to the pixel P can be approximated by:

$$V_X = \begin{pmatrix} 2 \\ 0 \\ P_r - P_l \end{pmatrix} \quad \text{and} \quad V_Y = \begin{pmatrix} 0 \\ 2 \\ P_d - P_u \end{pmatrix}. \tag{4.1}$$

The normal vector ($\hat{N}$) is given by the cross product of the tangent vectors found with Equation 4.1. It is possible to get a normalized normal (N) with the following formula:

$$\hat{N} = V_X \times V_Y \quad \text{and} \quad N = \frac{\hat{N}}{\sqrt{\hat{N} \bullet \hat{N}}}, \tag{4.2}$$

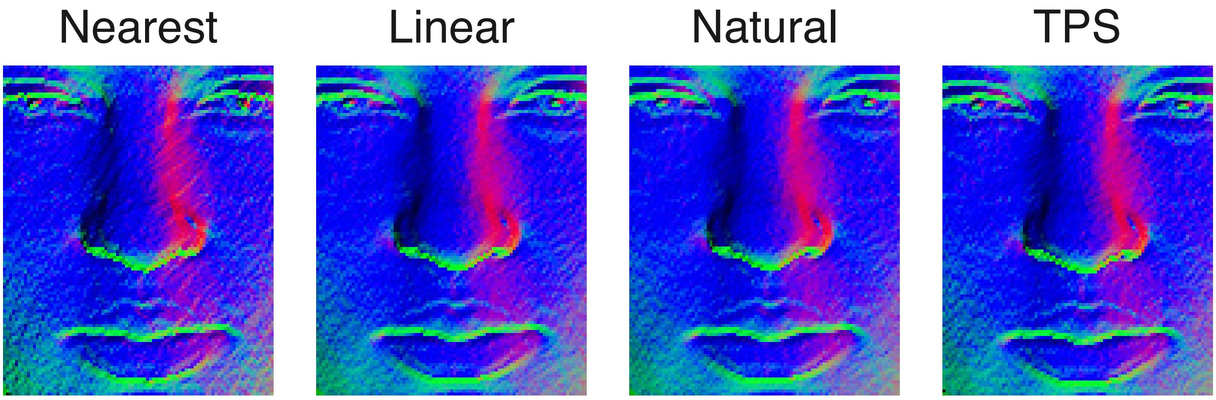

Figure 4.4: Visualization of the normal vectors computed for the surfaces defined by each interpolation method. The x, y, z normals coordinates of the unit normals were respectively mapped to the R, G, B colors.

where $\bullet$ denotes the scalar product between the vectors. Note that it is not possible to calculate the normals on the first and last rows nor the first and last columns of the image, since in that cases not all neighbors are available. An example of the normals computed for each interpolation method can be seen in Figure 4.4.

After the normals were calculated, it was necessary to reorganize the needle-maps (*i.e.,* each subject's facial surface normals) in the manner that the classification models expect. The data input format is a 3-dimensional array where each row is associated with a particular pixel. The matrix has three columns, one for each coordinate of the normal $(x, y, z)^T$. In other words, the first two dimensions of the array represent a specific subject's image. All these 2-dimensional matrices are then combined pixel-by-pixel in the third dimension, creating a collection of images.

The normals are used by the gender classification models to find the mean face (*i.e.*, an intrinsic mean) of each gender. Suppose that there are K example facial needle-maps each having N pixel locations. The surface normal at the pixel location l for the k^{th} needle-map is n_l^k. The authors calculate the intrinsic mean μ_l of the distribution of surface normals $n_l^1, ...n_l^K$ at each pixel location l. Note that the *intrinsic mean* in this context is different from the mean facial tetrahedron presented in Section 3.3. The goal here is to compute the mean face of each gender, using their facial surface normals, in order to obtain the intrinsic characteristics related to a given gender.

4.3.2 Standard PGA

The Standard PGA gender classification model, referred in this work as PGA, is presented in [Wu et al. 2010]. The authors apply PGA to a set of example facial needle-maps for the purpose of learning a statistical model of facial shape.

In order to evaluate the performance of the PGA model using different interpolation methods, several statistical metrics are calculated in Section 4.4.

4.3.3 Supervised Weighted PGA

The Supervised Weighted PGA (or SWPGA) method is an improvement of the weighted PGA method [Wu et al. 2007b]. First, it is important to mention that the Weighted PGA constructs a weight map from the mean faces of the two genders. The weight in each pixel of the map is calculated based on the angular difference between the unit surface normal of the mean faces of each gender. The SWPGA method enhances the Weighted PGA method by using an iterative learning process to construct the weight map. Note that the Weighted PGA method is not used in this work.

The SWPGA method relies on the proper setting of the parameter d, which indicates the number of features (*i.e.,* relevant dimensions of the facial feature space) to be used during the iterative construction process of the weight map. In the experiments presented in this work, it was chosen $d = 5$ (where d is the number of features to be used) because it has been shown in [Wu et al. 2007a] that five gender discriminating features are sufficient for gender classification. The number of weight map iterations was set to 6000, as it was used in [Wu et al. 2011].

4.3.4 Classifier

With the extracted feature vectors of the training and test facial needle-maps at hand, the Euclidean distance between the feature vectors of the test face and the two mean faces is calculated. Suppose $p = (p_1, p_2, ..., p_n)$ and $q = (q_1, q_2, ..., q_n)$ are two points in Euclidian n-feature space, then the Euclidian distance between p and q is given by:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (4.3)$$

The Euclidian distance is used as a criterion to classify the test faces in the two possible genders. The mean faces of the two genders are obtained from the *training* data using each gender's intrinsic mean. The Euclidian distance of each test face is calculated twice: from the test face to the *male* intrinsic mean and from the test face to the *female* intrinsic mean. The test face will be classified depending on which gender it is closer to in the feature space.

4.4 Statistical Metrics

Cross-validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set [Devijver and Kittler 1982, Kohavi 1995]. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

One round of cross-validation involves partitioning the data samples into complementary subsets, performing the analysis on one subset (called the *training* set), and validating the analysis on the other subset (called the validation set or *testing* set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for *testing* the model, and the remaining $k - 1$ subsamples are used as *training* data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds then can be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

The image set used for both *training* and *testing* was comprised of 180 images (90 females and 90 males) from different individuals. Although each subject may have several different images in the dataset, only one image per subject was used during the tests to avoid bias.

The cross-validation was run with $k = 5$ folds, therefore each fold contained 36 subjects (18 females and 18 male). At each round, one fold was reserved for testing and the others were used for training. After all rounds have finished, the measured statistics are averaged by the number of rounds. In this work, k-fold was used to calculate the metrics derived from the gender classification confusion matrix (Subsection 4.4.1) and to calculate the MCC (Subsection 4.4.2).

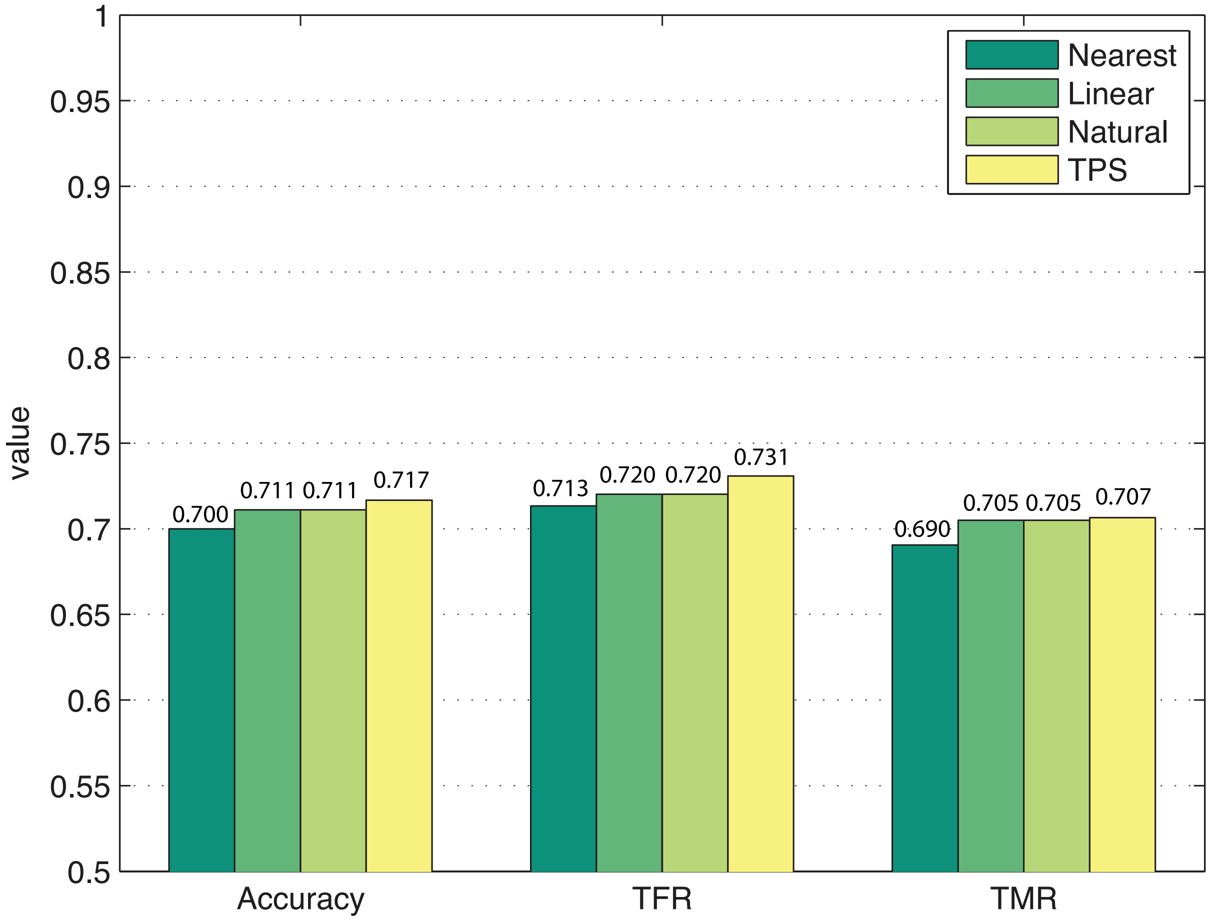

Figure 4.5: Statistical Metrics for standard PGA. TFR stands for True Females Rate and TMR stands for True Males Rate.

4.4.1 Confusion matrix-based metrics

The gender classification can be interpreted as a binary classification. For that matter, it is necessary to fit the two genders within two classes: Female and Male. The following nomenclature is used to refer to the cells of the resulting confusion matrix: *TF* (True Females) are the females identified as such, *TM* (True Males) are the males identified as such, *FF* (False Females) are the females incorrectly classified as being male and *FM* (False Males) are the males incorrectly classified as being female. The confusion matrix provides information about the number of correct classifications in comparison to the predict classifications for each class.

The *accuracy* of a measurement system is defined as the degree of closeness of measurements of a quantity to that quantity's actual (true) value [BIPM et al. 2008]. In other words, the *accuracy* is the proportion of true results (both true females and true males) in the population. The *accuracy* can be calculated as follows:

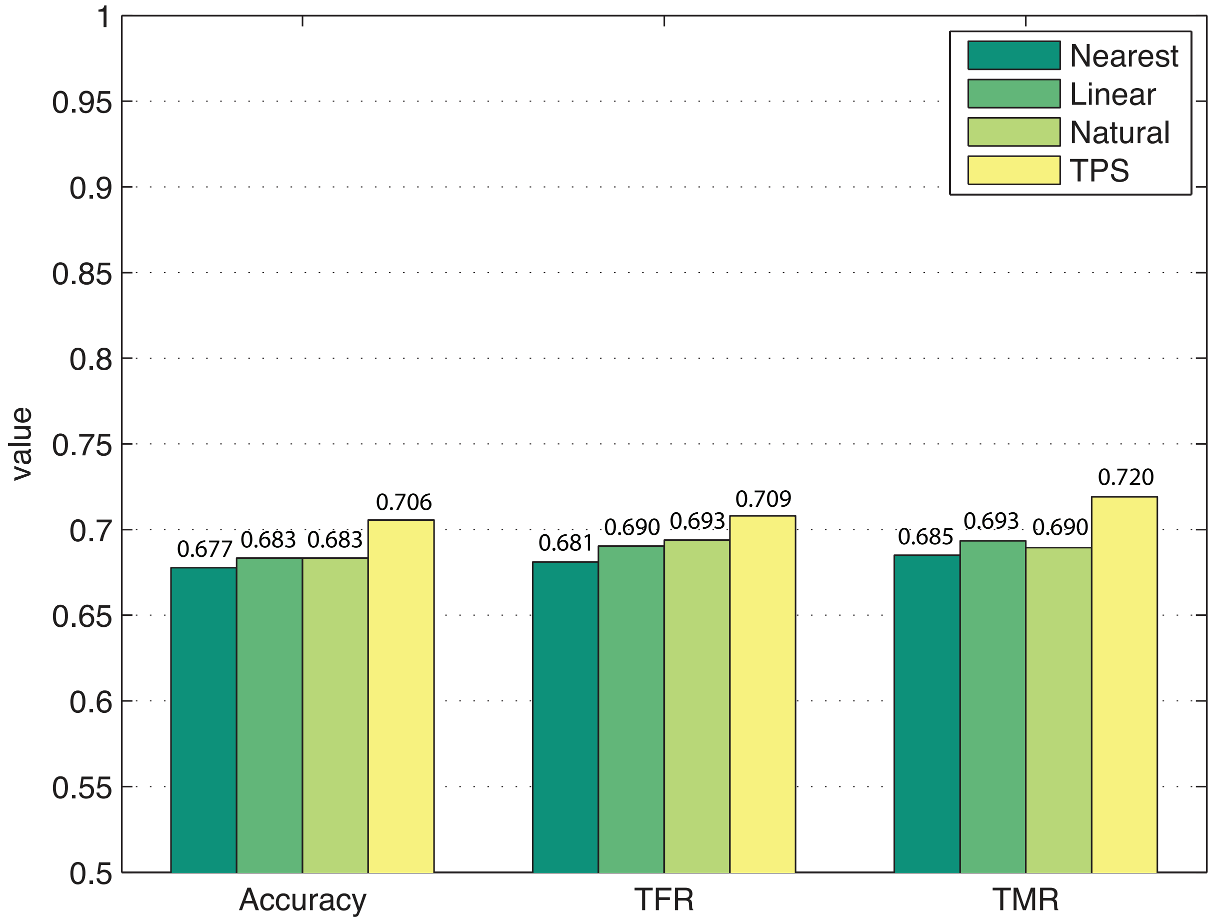

Figure 4.6: Statistical Metrics for SWPGA. TFR stands for True Females Rate and TMR stands for True Males Rate.

$$accuracy = \frac{TF + TM}{TF + FF + FM + TM}. \tag{4.4}$$

True Females Rate (TFR) and *True Males Rate* (TMR) are also statistical measures of the performance of the binary classification test presented in this work. TFR measures the proportion of actual females which are correctly identified as such. TMR measures the proportion of males which are correctly identified. A perfect predictor would be described as 100% TFR and 100% TMR, however theoretically any predictor will possess a minimum error bound known as the Bayes error rate, which is the lowest possible error rate for a given class of classifier [Fukunaga 1990, Tumer and Ghosh 1996]. The TFR and TMR are described as:

$$TFR = \frac{TF}{TF + FM} \quad \text{and} \quad TMR = \frac{TM}{TM + FF}. \tag{4.5}$$

By looking at Equation 4.5, TFR can be interpreted as a bias towards female classifi-

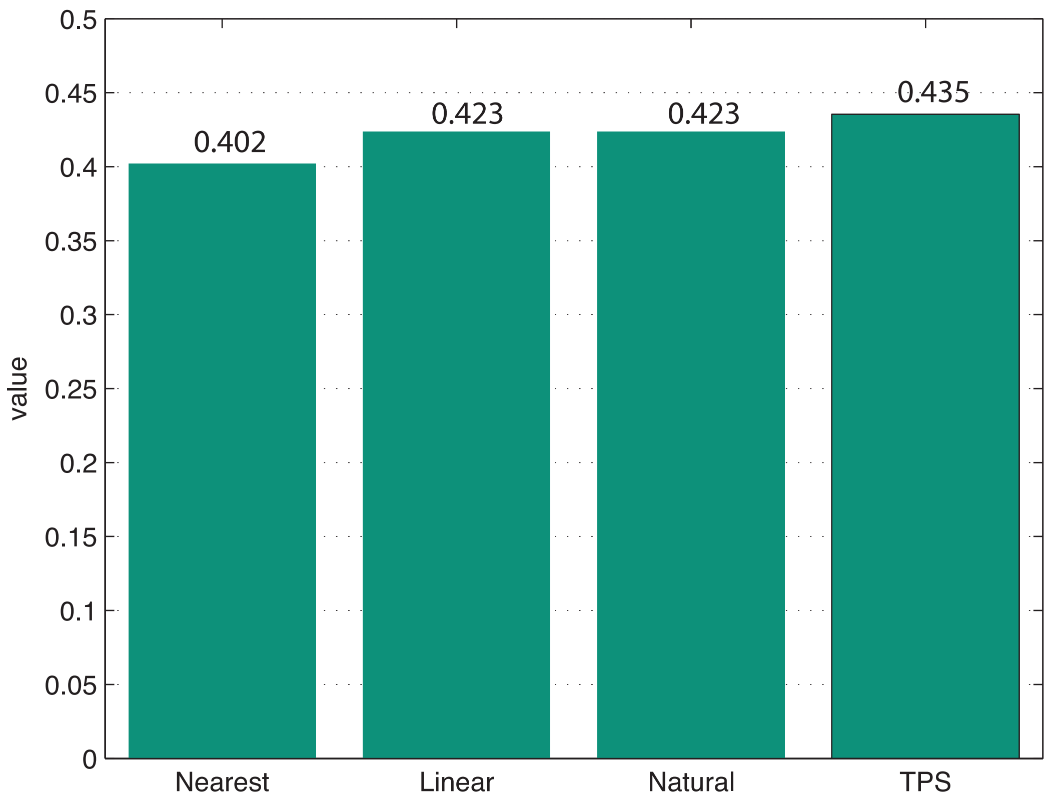

Figure 4.7: Matthews Correlation Coefficient (MCC) for standard PGA.

cation or the capacity of correctly identifying the female gender. Similarly, TMR can be interpreted as the bias towards male classification or the capacity of correctly identifying the male gender.

The described metrics were calculated for each interpolation method (Nearest-neighbor, Linear, Natural-neighbor and TPS). The average of each metric was obtained for the k folds and the results were plotted as vertical bars, as shown in Figures 4.5 and 4.6, respectively for PGA and SWPGA. As expected, the Nearest-neighbor method had the poorest accuracy, while the Linear and Natural-neighbor are tied. The TPS had a higher accuracy, proving that smoother interpolation can increase the gender classification performance.

By comparing Figures 4.5 and 4.6, it is important to notice that the accuracy differences in the PGA model are smaller than those found in the SWPGA model. The TPS in the SWPGA showed an increased accuracy of up to 3%. It is believed that, since the SWPGA iteratively creates a weight map to describe relevant discriminating regions, iterative methods have a tendency to amplify the errors introduced by the interpolating functions and, as a consequence, the TPS resulted in higher *accuracy* when applied in the

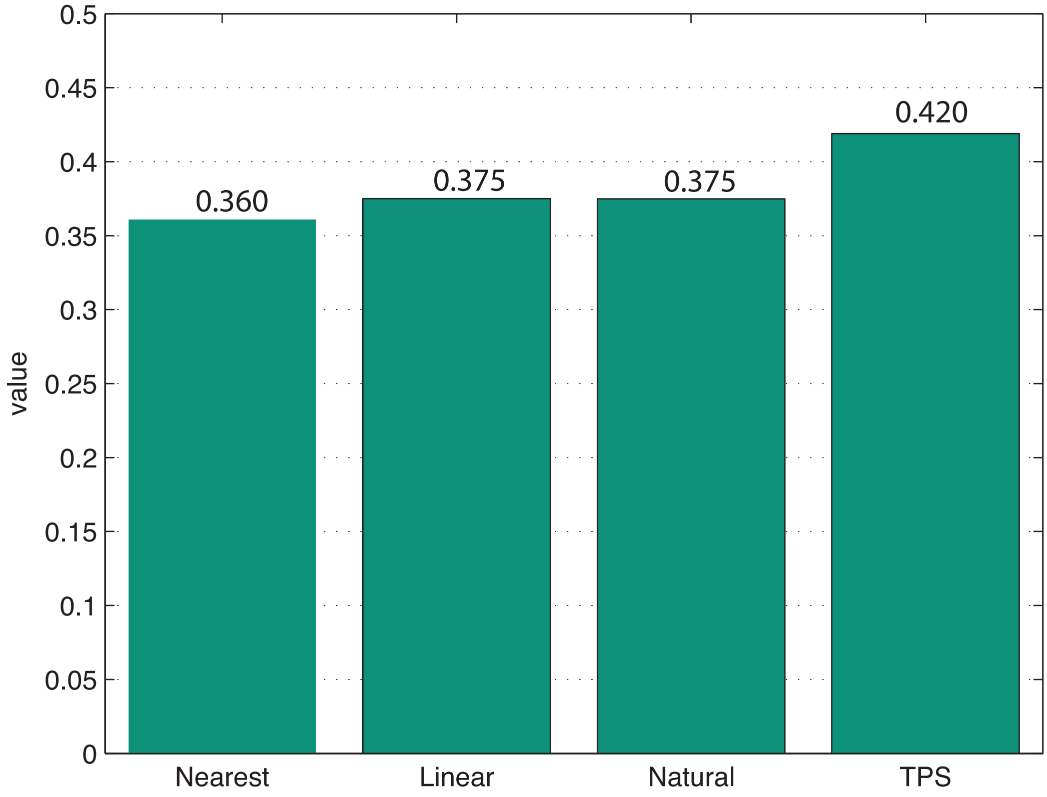

Figure 4.8: Matthews Correlation Coefficient (MCC) for SWPGA.

gender discriminating model.

4.4.2 Matthews correlation coefficient

The Matthews Correlation Coefficient (MCC) is used in machine learning as a measure of the quality of two-class (or binary) classifications. It takes into account true and false classifications and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications. It returns a value between -1 and $+1$. A coefficient of $+1$ represents a perfect prediction, 0 means no better than random prediction and -1 indicates total disagreement between prediction and observation.

The MCC can be calculated using the formula:

$$MCC = \frac{TF \times TM - FF \times FM}{\sqrt{(TF + FF)(TF + FM)(TM + FF)(TM + FM)}}. \tag{4.6}$$

The MCC of each interpolation method was averaged for the k folds and the results can be seen in Figures 4.7 and 4.8. Similarly to the aforementioned metrics, the TPS also showed better performance in the PGA and SWPGA models. The differences were even higher when comparing the interpolation methods in the SWPGA model. The experiments show that the Nearest-neighbor, Linear and Natural-neighbor interpolations are equivalent to each other. To the researcher this means that, in non-iterative techniques, these simpler interpolation methods, specially the Linear and Natural-neighbor interpolations, can be applied with no major impact on the final classification result. However, it is recommended to use the TPS in techniques that may amplify the interpolation errors (*i.e.*, iterative methods).

Chapter 5

Conclusions and Future Work

Several systems use facial depth images to perform tasks such as face, gender, ethnic and age group classification. Therefore, it is crucial to correctly align a given set of facial data and reconstruct any missing portions of depth information, since it can greatly affect the results of those systems. Unfortunately, most of the techniques do not make proper use of depth information while performing the alignment and the usage of poor interpolation methods often lead to unnatural flat artifacts on the facial surface.

This work presented a completely automatic approach for aligning and reconstructing damaged facial depth images. The approach uses TPS to smoothly interpolate existing data, facial landmarks to ensure data alignment, and RM-based ray casting to render the final aligned depth image having arbitrary resolution. It has been demonstrated the effectiveness of the proposed techniques by implementing it and using it to align faces from several real depth images available in a well-known biometric dataset.

In order to reduce the high computational costs of the TPS, a block division approach was introduced where separate TPSs are adjusted to each block, considerably reducing the time needed to fit an interpolation.

The alignment of several facial images was performed by creating a tetrahedron based on the mean distances between facial landmarks, and by transforming the tetrahedron defined for each input face to the mean one. These affine transformations are used to map all original image coordinates to the space where this new tetrahedron is located.

To resample depth values, a ray-casting procedure based on the RM technique is presented. The approach consists in casting rays from a pinhole camera and then increment the ray at small steps until it finds the first intersection with the facial surface.

The proposed alignment and interpolation methods were compared against a common approach based on 2-dimensional alignment and linear interpolation that operates on intensity images. The errors in the proposed method were up to two orders of magnitude smaller than the common approach. This result suggests that it is recommended the use of the proposed techniques in order to achieve better results on procedures that are currently based on naive alignment and reconstruction of depth data.

When using the proposed alignment method (in the 3-dimensional domain) while varying only the interpolation scheme it has been shown that the difference in the final results were better in favor of the proposed technique. More specifically, the experiments suggest that the TPS has better results when used within iterative methods since the feedback mechanisms of this kind of procedure tend to amplify errors (*e.g.,* interpolation errors). Also, the TPS has proven able to increase the gender classification accuracy of the SWPGA model by 3%. With that in mind, the presented conclusion is that when the research is in a prototyping phase he/she may use a simple interpolation method (*i.e.,* Linear) in order to validate their implementation and later use a more sophisticated interpolation method (*i.e.,* TPS) to improve the final results.

5.1 Future Work

In comparison to other interpolation methods, presented in Chapter 2, the TPS has the disadvantage that it does not allow new control points to be added or removed once the coefficients have been calculated. One way to work around this limitation would be the use of a system similar to that of the natural-neighbor interpolation (Section 2.3). The idea is to construct a *Voronoi diagram* (or use its dual, the Delaunay triangulation) to define regions where TPSs could be fit. Then, when one point or a set of points needs to be added or removed, it would be possible to recalculate the TPS only for that specific region. This approach would overcome the TPS's current limitation of being immutable. On the other hand, the type of data used in this work (*i.e.,* facial depth images) does not require points to be added or removed.

The UND Biometric dataset contained several images which had noisy data being identified as having valid depth values. The presence of those values may negatively affect the final result of all interpolation approaches during the RM step. In order to reduce the influence of these points, some non-linear noise reduction filter (*e.g.,* a median filter) could be applied to the original data to eliminate the incorrect depth data. It is important to note that this problem does not affect only the TPS, but also any interpolant that might be used with the RM technique.

It is expected that, by increasing the resolution of final image during the reconstruction step the smooth interpolation provided by the TPS will improve the results even further. Unfortunately, it was not possible to test this hypothesis in this work due to two reasons: (a) this would require more processing time and, (b) as pointed out by Wu et al. [Wu et al. 2011], the dimensionality of the facial needle-maps cannot be too large in order to keep the manipulation of the covariance matrix feasible. In the gender classification context, larger matrices could introduce numerical instability. For that matter, the numerically efficient snap-shot method of Sirovich [Sirovich 1987] could be used to compute the eigenvectors of the covariance matrix. It is important to note that the large processing time and the numerical instability are related to the iterative gender classification method of Wu et al. [Wu et al. 2011].

Bibliography

[BIPM et al. 2008] BIPM, I., IFCC, I., ISO, I., and IUPAP, O. (2008). *The International Vocabulary of Metrology - Basic and general concepts and associated terms (VIM).* Joint Committee for Guides in Metrology (JCGM), 3rd edition.

[Bookstein 1989] Bookstein, F. L. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585.

[Chang et al. 2003] Chang, K., Bowyer, K., and Flynn, P. (2003). Face recognition using 2D and 3D facial data. In *ACM Workshop on Multimodal User Authentication*, pages 25–32, Santa Barbara.

[Devijver and Kittler 1982] Devijver, P. and Kittler, J. (1982). *Pattern recognition: A statistical approach.* Prentice-Hall International, London, GB.

[Fukunaga 1990] Fukunaga, K. (1990). *Introduction to statistical pattern recognition.* Academic Pr. ISBN 0122698517 pages 3 and 97.

[Guo et al. 2004] Guo, H., Jiang, J., and Zhang, L. (2004). Building a 3D morphable face model by using thin plate splines for face reconstruction. In *Proceedings of the 5th Chinese Conference on Biometric Recognition, SINOBIOMETRICS*, pages 258–267, Guanzhou, China.

[Hartley and Zisserman 2000] Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2nd edition.

[Kohavi 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 2, pages 1137–1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Kohavi and Provost 1998] Kohavi, R. and Provost, F. (1998). Glossary of terms. *Machine Learning*, 30(2-3):271–274.

[Lu et al. 2005] Lu, X., Chen, H., and Jain, A. (2005). Multimodal facial gender and ethnicity identification. In Zhang, D. and Jain, A., editors, *Advances in Biometrics*, volume 3832 of *Lecture Notes in Computer Science*, pages 554–561. Springer Berlin, Heidelberg.

[MATLAB 2009a] MATLAB (2009a). *MATLAB Documentation.* The MathWorks Inc., Natick, Massachussets.

[MATLAB 2009b] MATLAB (2009b). *MATLAB version 7.9.0.* The MathWorks Inc., Natick, Massachussets.

[Matthews 1975] Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, 405(2):442.

[Minolta 2003] Minolta (2003). Non-contact 3D digitizer VIVID 900/910, specification [online] `http://www.konicaminolta.com/instruments/products/3d/non-contact/vivid910/specifications.html`.

[Moreno et al. 2005] Moreno, A. B., Sanchez, A., Velez, J. F., and Diaz, F. J. (2005). Face recognition using 3D local geometrical features: PCA vs. SVM. In *Proceedings of the 28th Annual Conference of the International Society of Parametric Analysts.*

[Netpbm 2000] Netpbm (2000). PPM file format specification [online] `http://netpbm.sourceforge.net/doc/ppm.html`.

[Perakis et al. 2009] Perakis, P., Theoharis, T., Passalis, G., and Kakadiaris, I. A. (2009). Automatic 3D facial region retrieval from multi-pose facial datasets. In *Proceedings of the 3rd Eurographics Workshop on 3D Object Retrieval*, pages 37–44, Munich, Germany.

[Policarpo et al. 2005] Policarpo, F., Oliveira, M. M., and Comba, J. (2005). Real-time relief mapping on arbitrary polygonal surfaces. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, pages 155–162, Washington, DC, USA.

[Preparata and Shamos 1985] Preparata, F. and Shamos, M. (1985). Computational geometry: An introduction. pages 1–35.

[Romero-Huertas and Pears 2008] Romero-Huertas, M. and Pears, N. (2008). 3D facial landmark localisation by matching simple descriptors. In *Proceedings of IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, Washington, DC, USA.

[Sibson 1981] Sibson, R. (1981). A brief description of natural neighbour interpolation. In Barnet, V., editor, *Interpreting Multivariate Data*, pages 21–36. Chichester: John Wiley.

[Sirovich 1987] Sirovich, L. (1987). Turbulence and the dynamics of coherent structures. part I: Coherent structures. *Quarterly of applied mathematics*, 45(3):561–571.

[Stormer and Rigoll 2008] Stormer, A. and Rigoll, G. (2008). A multi-step alignment scheme for face recognition in range images. In *Proceedings of IEEE International Conference on Image Processing*, pages 2748–2751, San Diego, California, USA.

[Tumer and Ghosh 1996] Tumer, K. and Ghosh, J. (1996). Estimating the Bayes error rate through classifier combining. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 2, pages 695–699, Vienna, Austria.

[Voronoi 1908] Voronoi, G. (1908). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(134):198–287.

[Whitbeck and Guo 2006] Whitbeck, M. and Guo, H. (2006). Multiple landmark warping using thin-plate splines. In *Proceedings of the International Conference on Image Processing, Computer Vision and Pattern Recognition*, pages 256–263, Las Vegas, Nevada, USA.

[Wu et al. 2007a] Wu, J., Smith, W., and Hancock, E. (2007a). Learning mixture models for gender classification based on facial surface normals. In *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, pages 39–46, Girona, Spain.

[Wu et al. 2007b] Wu, J., Smith, W., and Hancock, E. (2007b). Weighted principal geodesic analysis for facial gender classification. In *Proceedings of the Congress on Pattern Recognition 12th Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications*, pages 331–339, Valparaiso, Chile.

[Wu et al. 2010] Wu, J., Smith, W., and Hancock, E. (2010). Facial gender classification using shape-from-shading. *Image and Vision Computing*, 28(6):1039–1048.

[Wu et al. 2011] Wu, J., Smith, W., and Hancock, E. (2011). Gender discriminating models from facial surface normals. *Pattern Recognition*, 44(12):2871–2886.

APPENDIX A – The UND Biometric Dataset

The University of Notre Dame (UND) Biometric Dataset (Collection D) is comprised of 953 depth images and the corresponding colored frontal face images from 277 human subjects, captured in 2003. These images were acquired with a Minolta Vivid 900 3-dimensional range scanner [Minolta 2003].

The 2-dimensional image is a Portable Pixel Map (PPM) file [Netpbm 2000]. It contains a header which informs relevant information to calibrate the camera's intrinsic parameter matrix, such as the focus distance, the scale factors relating pixels to distance, the skew and the center coordinates of the image.

This is an example of the header of a PPM image file:

```
P6
#vivid 3D Data
#width 640
#height 480
# ViewInfo
# pitchX 0.007400
# pitchY 0.007400
# focus 14.380219
# viewPoint 0.000000 0.000000 21.660923
# interest 0.000000 0.000000 −1362.000000
# centerX 320.000000
# centerY 240.000000
# date  Thu Feb 13 14:14:37 2003
# upVector 0.000000 1.000000 0.000000
640 480
255
```

The **.abs** file can be interpreted as a text file and starts with the following header:

```
480 rows
640 columns
pixels (flag X Y Z):
```

By looking at the first two rows, it is known that the depth image has a resolution of 640 × 480 (*width* × *height*). The third row contains some dummy text that can be ignored.

After these three rows, there are four other containing the actual data, being float numbers separated by blank spaces. The first row is the flag data which informs the current pixel contains valid data, with either 0 (false) or 1 (true). The following three lines contain, respectively, the X, Y and Z values of the image's point in world coordinate system.

Along with the depth and colored images, the dataset comes with several XML files that provide information regarding the sensor, stage, illuminant and environment conditions. However, the relevant ones are actually the **recordings.xml** and **subjects.xml**.

The **recordings.xml** file contains several relevant information about each image, such as the position of lateral canthus of both eyes, the positions of the nose tip and the pogonion (*i.e.,* the pogonion is the most forward-projecting point on the anterior surface of the chin), providing four facial landmarks.

See an example of the contents of the **recordings.xml** file:

Listing A.1: Example of the contents of the **recordings.xml** file.

```
<Recording id="nd1R51294">
  <URL root="nd1/" relative="Spring2003range/04229d350.abs.gz"/>
  <CaptureDate>01/21/2003</CaptureDate>
  <CaptureTime>13:00:00</CaptureTime>
  <Format value="abs" />
  <Subject id="nd1S04229">
   <Application>
    <Face>
     <Pose yaw="0" pitch="0" roll="0" />
     <Wearing glasses="No" source="Retrospectively" />
     <Emotion type="BlankStare" source="Given" rank="1"/>
     <LeftEye>
      <CanthusLateral x="430" y="227" />
     </LeftEye>
     <RightEye>
      <CanthusLateral x="285" y="224" />
     </RightEye>
     <Nose x="360" y="281" />
     <Chin>
      <Pogonion x="358" y="371" />
     </Chin>
    </Face>
   </Application>
   <Stage id="nd1T00002" />
  </Subject>
  <Collection id="nd1C00003" />
  <Environment id="nd1E00014" />
  <Sensor id="nd1N00004" />
  <Illuminant id="nd1I00002" />
  <Illuminant id="nd1I00003" />
  <Weather condition="Inside" />
</Recording>
```

With the subject id field at hand, the file **subjects.xml** provides classification-relevant information, such as the gender, year-of-birth and ethnicity.

See below an example of the **subjects.xml** file's content:

Listing A.2: Example of the contents of the **subjects.xml** file.

```
<Subject id="nd1S04229" restricted="No">
  <Gender value="Male" source="Given" />
  <YOB value="1973" source="Given" />
  <Race value="Asian" source="Retrospectively" />
</Subject>
```

A.1 Inconsistent Data

The alignment step, presented in Section 3.3, is heavily dependent on the correctness of the four facial landmarks. As the bottom row of Figure A.1 shows, there are some images in the dataset where the facial landmarks do not match correctly. Due to these findings, the whole data had to be manually inspected and the images that presented incorrect landmark values were removed from both the training and testing sets.

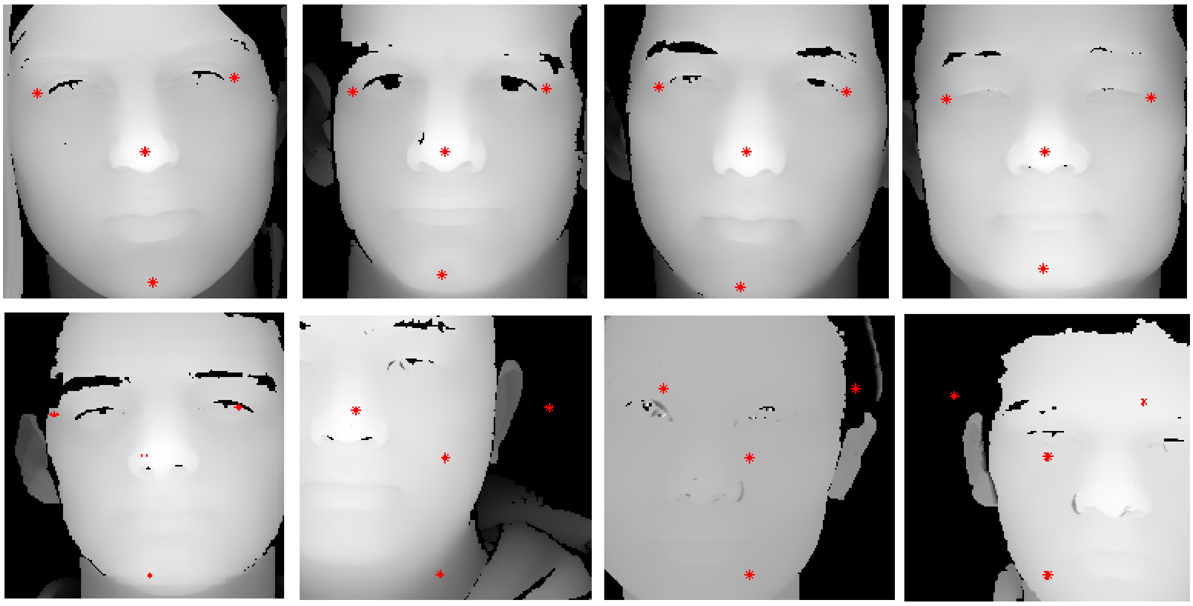

Figure A.1: Facial landmarks plotted with red stars using the positions provided by the XML files. Top row: examples of images with correct location of the landmarks. Bottom row: examples of images with incorrect location of the landmarks.

During the inspection of the images, it was also found a colored image that did not match with its corresponding depth image. As shown in Figure A.2, it is possible to observe a large difference in how the subject is framed. Also, the subject's clothing dras-

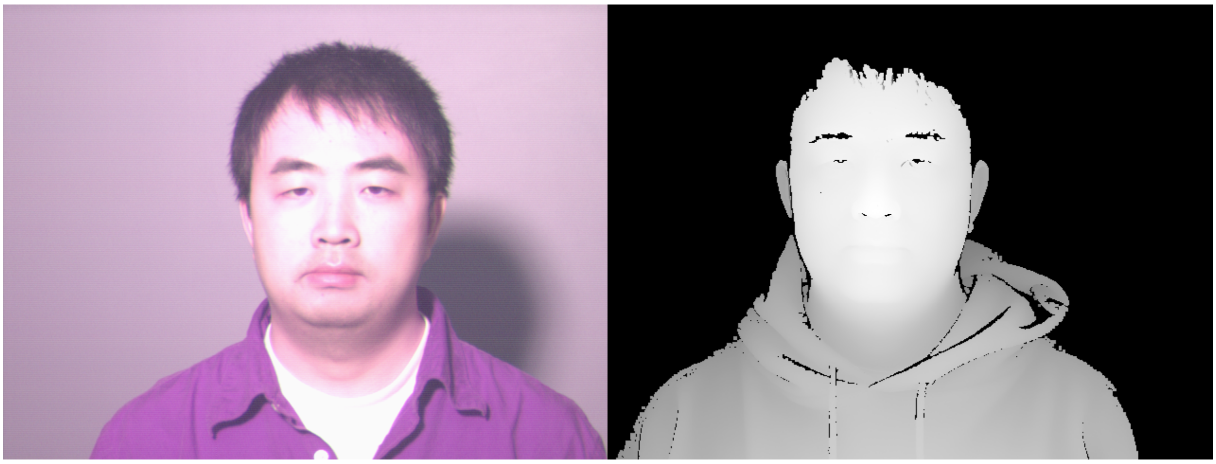

Figure A.2: An example of colored and depth images that do not match whereas expected. Those images were incorrectly associated in the XML file.

tically changes from an image to another. Note that the subject's position and clothing suggest that those images were incorrectly associated in the XML file.

The images containing the problems described in this Section were removed from the training and testing sets. The identification number of the images not used in this work due to the aforementioned problems are: 107, 123, 127, 150, 215, 226, 241, 245, 246, 248, 255, 285, 292, 309, 454, 600, 618, 626, 663, 674, 699, 704, 731, 743, 745, 762, 882, 815, 824, 847, 926, 953.