Dynamic 2D/3D Registration

Sofien Bouaziz Andrea Tagliasacchi Mark Pauly
École Polytechnique Fédérale de Lausanne

Abstract

Image and geometry registration algorithms are an essential component of many computer graphics and computer vision systems. With recent technological advances in RGB-D sensors, such as the Microsoft Kinect or Asus Xtion Live, robust algorithms that combine 2D image and 3D geometry registration have become an active area of research. The goal of this course is to introduce the basics of 2D/3D registration algorithms and to provide theoretical explanations and practical tools to design computer vision and computer graphics systems based on RGB-D devices. To illustrate the theory and demonstrate practical relevance, we briefly discuss three applications: rigid scanning, non-rigid modeling, and realtime face tracking. Our course targets researchers and computer graphics practitioners with a background in computer graphics and/or computer vision. An up-to-date version of the course notes as well as slides and source code can be found at `http://lgg.epfl.ch/2d3dRegistration`.

About the lecturers

Sofien Bouaziz is a PhD student in the Computer Graphics and Geometry Laboratory at the École Polytechnique Fédérale de Lausanne (EPFL) under the supervision of Prof. Mark Pauly. He received his MSc degree in Computer Science from EPFL in 2009. His research interests include computer graphics, computer vision, and machine learning. Sofien co-developed the facial motion capture software *faceshift studio*.

e-mail: sofien.bouaziz@epfl.ch

website: `http://lgg.epfl.ch/~bouaziz`

Andrea Tagliasacchi is a post-doctoral scholar in the Computer Graphics and Geometry Laboratory at the Ecole Polytechnique Federale de Lausanne (EPFL). He received his MSc from Politecnico di Milano and a PhD from Simon Fraser University (SFU) under the joint supervision of Prof. Richard Zhang and Prof. Daniel Cohen-Or. His research interests include computer graphics, geometry processing and computer vision with a focus on geometry tracking.

e-mail: andrea.tagliasacchi@epfl.ch

website: `http://drtaglia.github.io`

Mark Pauly is an associate professor of computer science at EPFL in Lausanne, Switzerland, where he directs the Computer Graphics and Geometry Laboratory. Prior to joining EPFL he was an assistant professor at ETH Zurich and a postdoctoral scholar at Stanford University. He received his Ph.D. degree in 2003 from ETH Zurich. His research interests include computer graphics and animation, shape analysis, geometry processing, and architectural design.

e-mail: mark.pauly@epfl.ch

website: `http://lgg.epfl.ch`

Sofien and Mark are co-founders of faceshift AG (www.faceshift.com), an EPFL spin-off that brings high-quality markerless facial motion capture to the consumer market.

1 Introduction

Recent technological advances in RGB-D sensing devices, such as the Microsoft Kinect, facilitate numerous new and exciting applications, for example in 3D scanning [24] and human motion tracking [26, 19, 6]. While affordable and accessible, consumer-level RGB-D devices typically exhibit high noise levels in the acquired data. Moreover, difficult lighting situations and geometric occlusions commonly occur in many application settings, potentially leading to a severe degradation in data quality. This necessitates a particular emphasis on the robustness of image and geometry processing algorithms. The combination of 2D and 3D registration is one important aspect in the design of robust applications based on RGB-D devices. This lecture introduces the main concepts of 2D and 3D registration and explains how to combine them efficiently. An up-to-date version of these course notes as well as slides and source code can be found at `http://lgg.epfl.ch/2d3dRegistration`.

2 2D/3D Registration

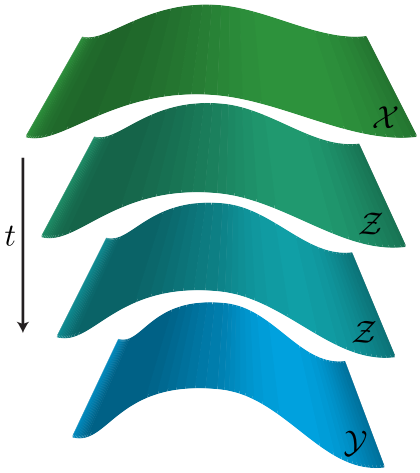
In the first part of the course we introduce the theory of 2D/3D registration algorithms suitable for processing RGB-D data. We focus on pairwise registration to compute the alignment of a source model onto a target model. This alignment can be rigid or non-rigid, depending on the type of object being scanned. We formulate the registration as the minimization of an energy

$$E_{\text{reg}} = E_{\text{match}} + E_{\text{prior}}. \tag{1}$$

The matching energy E_{match} defines a measure of how close the source is from the target. The prior energy E_{prior} quantifies the deviation from the type of transformation or deformation that the source is allowed to undergo during the registration, for example, a rigid motion or an elastic deformation. The goal of registration is to find a transformation of the source model that minimizes E_{reg} to bring the source into alignment with the target. For data acquired with RGB-D devices, registration can utilize both the geometric information encoded in the 3D depth map, as well as the color information provided by the recorded 2D images. We show that Equation 1 provides a unified way to formulate both 2D and 3D registration, which simplifies their integration.

2.1 3D Registration

In 3D registration we want to align a source surface $\mathcal{X}$ embedded in $\mathbb{R}^3$ to a target surface $\mathcal{Y}$ in $\mathbb{R}^3$. To formalize this problem, we introduce a surface $\mathcal{Z}$ that is a transformed or deformed version of $\mathcal{X}$ that eventually aligns with $\mathcal{Y}$. To solve the registration problem numerically, we represent the continuous surface $\mathcal{X}$ by a set of points $X = \{\mathbf{x}_i \in \mathcal{X}, i = 1 \dots n\}$ and define their corresponding points on the deformed surface $\mathcal{Z}$ as $Z = \{\mathbf{z}_i \in \mathcal{Z}, i = 1 \dots n\}$. Different sampling strategies have been presented by Rusinkiewicz and Levoy [21].



2.1.1 Matching energy

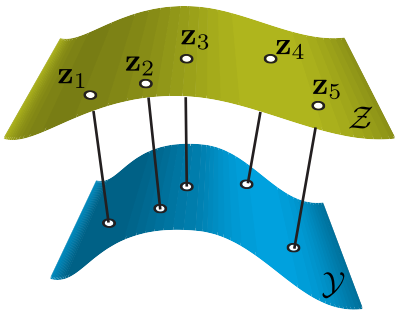
The matching energy measures how close the surface $\mathcal{Z}$ is to the surface $\mathcal{Y}$ and is defined as

$$E_{\text{match}}(\mathcal{Z}) = \int_{\mathcal{Z}} \varphi(\mathbf{z}, \mathcal{Y}) d\mathbf{z}, \tag{2}$$

where $\mathbf{z} \in \mathbb{R}^3$ is a point on $\mathcal{Z}$. The accuracy of the registration is evaluated by the metric φ that measures the distance to $\mathcal{Y}$. For simplicity, we will first use the squared Euclidian distance as metric. Robust metrics [17] could be use instead to increase the robustness of the registration to noise and outliers and will be presented later on. Using the set of points Z, we can discretize the matching energy as

$$E_{\text{match}}(Z) = \sum_{i=1}^{n} \|\mathbf{z}_i - P_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2. \tag{3}$$

where $P_{\mathcal{Y}}(\mathbf{z}_i) : \mathbb{R}^3 \to \mathbb{R}^3$ returns the closest point (using Euclidian distance) on the surface $\mathcal{Y}$ from $\mathbf{z}_i$. $P_{\mathcal{Y}}(\mathbf{z}_i)$ can also be seen as the orthogonal projection of $\mathbf{z}_i$ onto $\mathcal{Y}$.



2.1.2 Prior energy

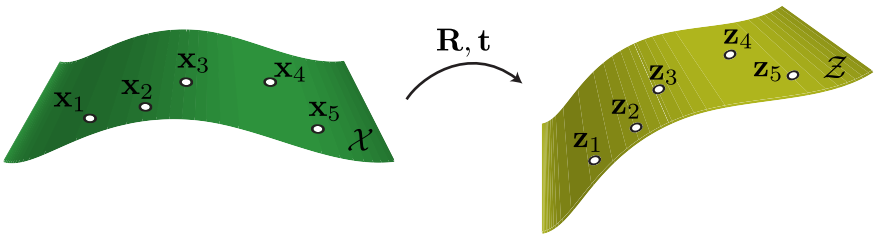
In this section we present several prior energies that can be used for registration. These energies can also be combined to build more sophisticated priors. Priors encode properties of the scanned objects. For example, when scanning rigid objects, a global rigidity

prior can be used to limit the allowed transformations to rotations and translations. For deforming objects, for example a human body, geometric priors are often employed that try to mimic physical behavior such as an elastic deformation. We describe a simple local rigidity prior that approximates elastic deformations and facilitates efficient implementations. More complex deformation behavior can be captured using a data-driven approach. One popular method is based on a collection of sample shapes that represent the space of space of allowed deformations. Using dimensionality reduction, for example principal component analysis, efficient linear models can be derived that are suitable for realtime registration algorithms.

Global rigidity. The global rigidity of the 3D registration can be measured as

$$E_{\text{rigid}}(Z, \mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2, \tag{4}$$

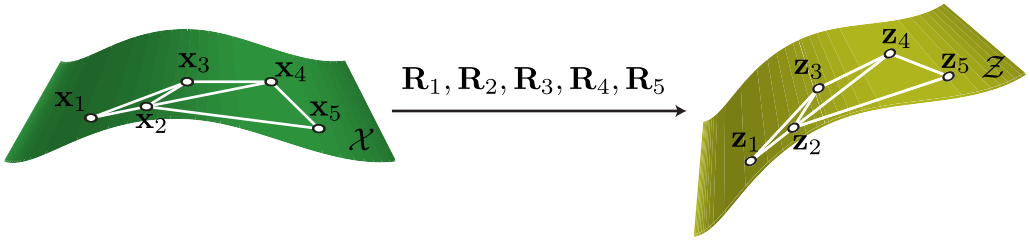
where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ a translation vector. In this case, the deformed surface $\mathcal{Z}$ tries to follow a rigid transformation of the original surface $\mathcal{X}$.



Local rigidity. The local rigidity energy, following [22, 4], can be expressed as

$$E_{\text{arap}}(Z, \mathbf{R}_i|_{i=1}^{n}) = \sum_{i=1}^{n} \sum_{j\in\mathcal{N}_i} \|(\mathbf{z}_j - \mathbf{z}_i) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_i)\|_2^2, \tag{5}$$

where the $\mathbf{R}_i \in \mathbb{R}^{3\times3}$ are rotation matrices and $\mathcal{N}_i$ is the set of indices of the neighboring points of $\mathbf{x}_i$. In this case, each local neighborhood on the surface $\mathcal{Z}$ tries to follow a rigid transformation of its corresponding local neighborhood on the surface $\mathcal{X}$. Other *local rigidity* energies can also be used as prior, see for example [3, 23].



Linear model. A 3D linear shape model can be defined using a matrix $\mathbf{P}$ containing the shape model basis, and a mean shape vector $\mathbf{m}$ [10]. A new shape $\mathbf{s}$ can be defined as

$$\mathbf{s} = \mathbf{P}\mathbf{d} + \mathbf{m}, \tag{6}$$

where $\mathbf{d}$ is a vector containing the basis coefficients. A linear model prior energy can be formulated as the deviation of the vertices from the linear model

$$E_{\text{prior}}(Z, \mathbf{d}) = \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i)\|_2^2, \tag{7}$$

where $\mathbf{P}_i$ and $\mathbf{m}_i$ are the part of $\mathbf{P}$ and $\mathbf{m}$ corresponding to the vertex $\mathbf{z}_i$.

2.1.3 Optimization

How to best optimize the registration energy depends on the prior energy. In this section we show, as an example, how to optimize a registration energy for two applications: rigid scanning and non-rigid modeling.

In-hand rigid scanning. Since single depth maps acquired with the RGB-D sensor exhibit high noise levels and do not cover the whole surface of the 3D object, an aggregation procedure is typically applied to obtain a complete model with reduced noise level. In order to aggregate multiple scans over time, different methods can be used [28, 29, 18]. The classical approach is to perform a 3D rigid registration of the currently acquired scan of the object with the already accumulated 3D data. The pairwise 3D alignment can be formulated as

$$\begin{aligned} E(Z, \mathbf{R}, \mathbf{t}) &= w_1 E_{\text{match}} + w_2 E_{\text{rigid}} \\ E_{\text{match}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - P_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 \\ E_{\text{rigid}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2 \end{aligned} \tag{8}$$

where the matching energy is combined with a global rigidity prior. To optimize $E(Z, \mathbf{R}, \mathbf{t})$ we linearize the rotation matrix [20] approximating $\cos\theta$ by 1 and $\sin\theta$ by θ

$$\mathbf{R} \approx \tilde{\mathbf{R}} = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix}. \tag{9}$$

The alignment is computed by solving iteratively

$$\underset{Z^{t+1}, \tilde{\mathbf{R}}, \tilde{\mathbf{t}}}{\arg\min} \sum_{i=1}^{n} w_1 \|\mathbf{z}_i^{t+1} - P_{\mathcal{Y}}(\mathbf{z}_i^t)\|_2^2 + w_2 \|\mathbf{z}_i^{t+1} - (\tilde{\mathbf{R}}(\mathbf{R}^t\mathbf{x}_i + \mathbf{t}^t) + \tilde{\mathbf{t}})\|_2^2, \tag{10}$$

where t is the iteration number and $\mathbf{z}_i^0 = \mathbf{x}_i$. As $P_{\mathcal{Y}}(.)$ is a non linear function that is difficult to optimize with, we use in the optimization the previous estimate $P_{\mathcal{Y}}(\mathbf{z}_i^t)$. This correspond to the *point-to-point* matching error [1]. To speed up the convergence of the optimization one can linearize $\|\mathbf{z}_i^{t+1} - P_{\mathcal{Y}}(\mathbf{z}_i^t)\|_2$ at $P_{\mathcal{Y}}(\mathbf{z}_i^t)$ which gives $\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - P_{\mathcal{Y}}(\mathbf{z}_i^t))$, where $\mathbf{n}_i$ is the normal of the surface $\mathcal{Y}$ at $P_{\mathcal{Y}}(\mathbf{z}_i^t)$. This leads to the *point-to-plane* matching error [8]. The optimization can be reformulated as

$$\underset{Z^{t+1},\tilde{\mathbf{R}},\tilde{\mathbf{t}}}{\arg\min} \sum_{i=1}^{n} w_1(\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - P_{\mathcal{Y}}(\mathbf{z}_i^t)))^2 + w_2\|\mathbf{z}_i^{t+1} - (\tilde{\mathbf{R}}(\mathbf{R}^t\mathbf{x}_i + \mathbf{t}^t) + \tilde{\mathbf{t}})\|_2^2. \tag{11}$$

Both Equation 10 and Equation 11 are quadratic, and therefore, can be optimized by setting the partial derivatives to zero by solving a linear system. During the optimization, it can be advantageous to apply a Tikhonov regularization to the parameters of the rigid motion as linearizing the rotation matrix assumes that the angles are small.

Title for this section?. It is interesting to note that when $w_2 = +\infty$ then $\mathbf{z}_i$ can be replaced into the matching energy by $\mathbf{R}\mathbf{x}_i + \mathbf{t}$ leading to a registration energy

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - P_{\mathcal{Y}}(\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2. \tag{12}$$

This energy can be minimized in a similar spirit by linearizing the rotation matrix and iteratively solving a linear system. Other approaches can be found in [11].

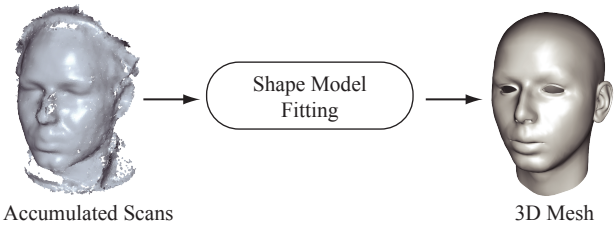


Figure 1: Registration of a morphable model towards the scanned face.

Non-rigid registration. Registering a shape template towards a scanned 3D object allows to obtain a complete and clean 3D mesh [15]. An example is given below in the context of face modeling. In this case, the morphable model of Blanz and Vetter [2] that represents the variations of different human faces in neutral expression is registered to a

scan of a face. Non-rigid modeling using a morphable model can be formulated as

$$
\begin{aligned}
E(Z, \mathbf{d}, \mathbf{R}_i|_{i=1}^n, \mathbf{R}, \mathbf{t}) &= w_1 E_{\text{match}} + w_2 E_{\text{rigid}} + w_3 E_{\text{model}} + w_4 E_{\text{arap}} \qquad (13)\\
E_{\text{match}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - P_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 \\
E_{\text{rigid}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2 \\
E_{\text{model}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i)\|_2^2 \\
E_{\text{arap}} &= \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j - \mathbf{z}_i) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_i)\|_2^2
\end{aligned} \qquad (14)
$$

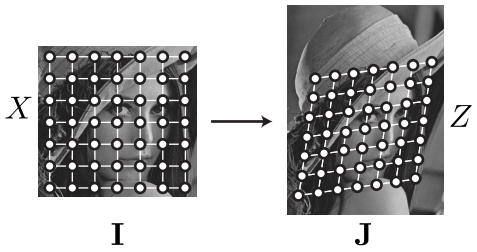
A local rigidity energy is added to the optimization in order to get an accurate result, as the morphable model represents the large-scale variability but might not capture small scale details. As previously, we solve iteratively

$$
\begin{aligned}
\underset{Z^{t+1}, \mathbf{d}, \tilde{\mathbf{R}}_i|_{i=1}^n, \tilde{\mathbf{R}}, \tilde{\mathbf{t}}}{\arg\min} \sum_{i=1}^{n} w_1(\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - P_{\mathcal{Y}}(\mathbf{z}_i^t)))^2 + w_2\|\mathbf{z}_i^{t+1} - (\tilde{\mathbf{R}}(\mathbf{R}^t\mathbf{x}_i + \mathbf{t}^t) + \tilde{\mathbf{t}})\|_2^2 + \\
w_3\|\mathbf{z}_i^{t+1} - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i)\|_2^2 + w_4 \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j^{t+1} - \mathbf{z}_i^{t+1}) - \tilde{\mathbf{R}}_i\mathbf{R}_i^t(\mathbf{x}_j - \mathbf{x}_i)\|_2^2, \qquad (15)
\end{aligned}
$$

which corresponds to solving a linear system.

2.2 2D Registration

In 2D registration we want to register a source image $\mathbf{I}$ to a target image $\mathbf{J}$. During the registration process, the 2D pixel grid of the source image $X = \{\mathbf{x}_i \in \mathbb{R}^2, i = 1 \dots n\}$ is deformed to $Z = \{\mathbf{z}_i \in \mathbb{R}^2, i = 1 \dots n\}$ to match the target image.



2.2.1 Matching energy

We define $\mathbf{I}(\mathbf{x})$ as the pixel value of the image $\mathbf{I}$ located at the position $\mathbf{x}$. The matching energy measures the color similarity between the source image and the target image

wrapped onto the deformed grid Z .

$$E_{\text{match}}(Z) = \sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{z}_i)\|_2^2. \tag{16}$$

2.2.2 Prior energy

Similarly to 3D geometry registration, we can use different prior energies that can be combined to build more complex priors.

Lucas-Kanade. In the Lucas-Kanade algorithm [16] the deformation is assumed to be constant within a patch around each pixel. This corresponds to the prior energy

$$E_{\text{LK}}(Z) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j - \mathbf{x}_j) - (\mathbf{z}_i - \mathbf{x}_i)\|_2^2, \tag{17}$$

where $\mathcal{N}_i$ is the set of indices of the neighbors of $\mathbf{x}_i$.

Horn-Schunck. In the Horn-Schunck algorithm [14] the smoothness of the flow is defined using a Laplacian operator

$$E_{\text{HK}}(Z) = \sum_{i=1}^{n} \|(\mathbf{z}_i - \mathbf{x}_i) - |\mathcal{N}_i|^{-1} \sum_{j \in \mathcal{N}_i} (\mathbf{z}_j - \mathbf{x}_j)\|_2^2, \tag{18}$$

where $|\mathcal{N}_i|$ is the cardinality of $\mathcal{N}_i$. This energy measures for each grid vertex the deviation of its deformation from the mean deformation of its neighbors.

2.2.3 Optimization

In this section we show, as an example, how to optimize the matching energy combined with the laplacian smoothness energy. This is similar to the method presented in [14]. Our optimization energy is

$$\begin{aligned} E(Z) &= w_1 E_{\text{match}} + w_2 E_{\text{HK}} \\ E_{\text{match}} &= \sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{z}_i)\|_2^2 \\ E_{\text{HK}} &= \sum_{i=1}^{n} \|(\mathbf{z}_i - \mathbf{x}_i) - |\mathcal{N}_i|^{-1} \sum_{j \in \mathcal{N}_i} (\mathbf{z}_j - \mathbf{x}_j)\|_2^2 \end{aligned} \tag{19}$$

To solve this optimization we linearize $\mathbf{J}(.)$ at the current estimate and solve iteratively

$$\arg\min_{Z^{t+1}} \sum_{i=1}^{n} w_1 \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{z}_i^t) - \nabla\mathbf{J}(\mathbf{z}_i^t)^T(\mathbf{z}_i^{t+1} - \mathbf{z}_i^t)\|_2^2 + w_2 \|(\mathbf{z}_i^{t+1} - \mathbf{x}_i) - |\mathcal{N}_i|^{-1} \sum_{j \in \mathcal{N}_i} (\mathbf{z}_j^{t+1} - \mathbf{x}_j)\|_2^2. \quad (20)$$

where $\nabla\mathbf{J} = \begin{bmatrix}\nabla\mathbf{J}_x & \nabla\mathbf{J}_y\end{bmatrix}^T$ is the image gradient, with $\nabla\mathbf{J}_x$ the image gradient in x direction and $\nabla\mathbf{J}_y$ the image gradient in y direction. As previously, the minimization can be computed by setting the partial derivative to zero, which corresponds to solving a linear system.

2.3 2D/3D Registration

We show how to combine 2D image registration and 3D geometry registration to best utilize the data provided by the RGB-D sensor. More specifically, we want to register a surface $\mathcal{X} \subset \mathbb{R}^3$ with color information $\mathbf{I}$, i.e. a texture mapped surface, to a 3D surface $\mathcal{Y}$ with corresponding color image $\mathbf{J}$. As previously, the source $\mathcal{X}$ is deformed to a surface $\mathcal{Z}$. We sample the continuous surface $\mathcal{X}$ to obtain a set of points $X = \{\mathbf{x}_i \in \mathcal{X}, i = 1 \dots n\}$. We define their corresponding points on the deformed surface $\mathcal{Z}$ as $Z = \{\mathbf{z}_i \in \mathcal{Z}, i = 1 \dots n\}$. The color information of sample point $\mathbf{x}_i$ is given by $\mathbf{I}(\mathbf{x}_i)$.

2.3.1 Matching energy

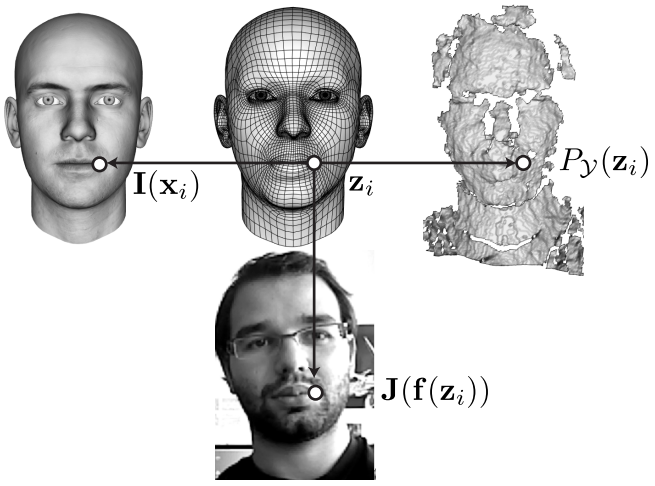
We formulate the energy measuring the quality of the 2D and 3D alignment as follow

$$E_{\text{match}}(Z) = \sum_{i=1}^{n} w_1 \|\mathbf{z}_i - P_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 + w_2 \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i))\|_2^2. \quad (21)$$

The first term is the matching energy presented in Section 2.1. The second term is similar to the 2D matching energy presented in Section 2.2. The only difference is the additional function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^2$ that projects a 3D point $\mathbf{z}_i$ to the 2D image $\mathbf{J}$. For example this function could be a perspective projection of the form $\mathbf{f}(\mathbf{z}_i) = \begin{bmatrix} f\frac{\mathbf{z}_{i,x}}{\mathbf{z}_{i,z}} & f\frac{\mathbf{z}_{i,y}}{\mathbf{z}_{i,z}} \end{bmatrix}^T$.

2.3.2 Optimization

We illustrate 2D/3D registration in the context of a face tracking system that combines the 2D/3D matching energy with a 3D blendshape prior. A blendshape representation is a linear model defined as a set of blendshape meshes $\mathbf{B} = [\mathbf{b}^0, ..., \mathbf{b}^n]$ where $\mathbf{b}_0$ is the rest pose and $\mathbf{b}_i, i > 0$ are different expressions. A new expression can be generated as $\mathbf{T} = \mathbf{b}^0 + \mathbf{B}\mathbf{d}$, where $\mathbf{B} = [\mathbf{b}^1 - \mathbf{b}^0, ..., \mathbf{b}^n - \mathbf{b}^0]$. The blendshape model shown below is inspired from Ekmans Facial Action Coding System [12]. Realtime face tracking using



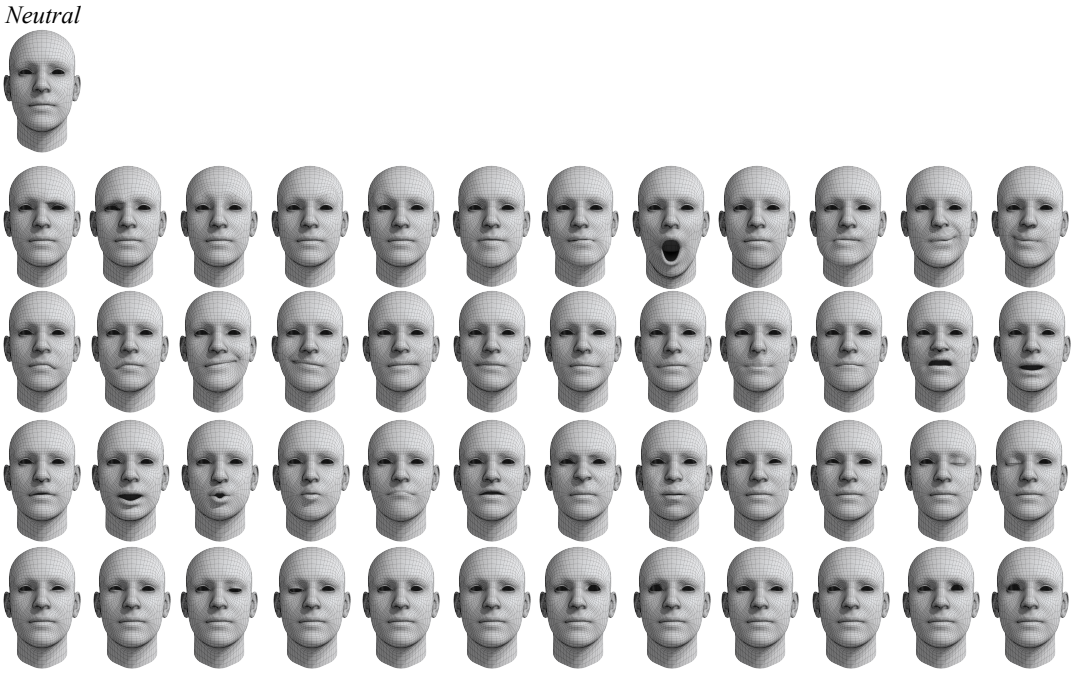


Figure 2: A blendshape model composed of 48 expressions.

an RGB-D device can be formulated as a 2D/3D registration of the blendshape model to the 2D and 3D data [27]. The registration energy can be formulated as

$$
\begin{aligned}
E(Z, \mathbf{d}, \mathbf{R}, \mathbf{t}) &= w_1 E_{\text{match geometry}} + w_2 E_{\text{match color}} + w_3 E_{\text{model+rigid}} \qquad (22)\\
E_{\text{match geometry}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - P_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 \\
E_{\text{match color}} &= \sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i))\|_2^2 \\
E_{\text{model+rigid}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}(\mathbf{B}_i\mathbf{d} + \mathbf{b}_i^0) + \mathbf{t})\|_2^2
\end{aligned}
$$

To solve this optimization we linearize $\mathbf{J}(\mathbf{f}(.))$ at the current estimate

$$\sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i^{t+1})\| \approx \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i^t)) - \nabla \mathbf{J}(\mathbf{f}(\mathbf{z}_i^t))^T \frac{\partial f(\mathbf{z}_i^t)}{\partial \mathbf{z}_i}(\mathbf{z}_i^{t+1} - \mathbf{z}_i^t)\|_2^2. \tag{23}$$

For a perspective projection $\mathbf{f}(\mathbf{z}_i) = \begin{bmatrix} \frac{f\mathbf{z}_{i,x}}{\mathbf{z}_{i,z}} & \frac{f\mathbf{z}_{i,y}}{\mathbf{z}_{i,z}} \end{bmatrix}^T$ we have

$$\frac{\partial f(\mathbf{z}_i)}{\partial \mathbf{z}_i} = \begin{bmatrix} \frac{f}{\mathbf{z}_{i,z}} & 0 & -\frac{f\mathbf{z}_{i,x}}{\mathbf{z}_{i,z}^2} \\ 0 & \frac{f}{\mathbf{z}_{i,z}} & -\frac{f\mathbf{z}_{i,y}}{\mathbf{z}_{i,z}^2} \end{bmatrix}. \tag{24}$$

In [27], the global rigidity is decoupled leading to a two steps optimization procedure. In a first step, a 2D/3D alignment of the blendshape model is computed

$$\begin{aligned} \arg\min_{Z^{t+1}, \mathbf{d}^{t+1}} \sum_{i=1}^{n} w_1(\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - P_{\mathcal{Y}}(\mathbf{z}_i^t)))^2 + \\ w_2\|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i^t)) - \nabla \mathbf{J}(\mathbf{f}(\mathbf{z}_i^t))^T \frac{\partial f(\mathbf{z}_i^t)}{\partial \mathbf{z}_i}(\mathbf{z}_i^{t+1} - \mathbf{z}_i^t)\|_2^2 + \\ w_3\|\mathbf{z}_i^{t+1} - (\mathbf{R}^t(\mathbf{B}_i\mathbf{d}^{t+1} + \mathbf{b}_i^0) + \mathbf{t}^t)\|_2^2, \end{aligned} \tag{25}$$

in a second step, a 3D rigid alignment is performed

$$\arg\min_{\mathbf{R}^{t+1}, \mathbf{t}^{t+1}} \sum_{i=1}^{n} \|\mathbf{z}_i^{t+1} - (\mathbf{R}^{t+1}(\mathbf{B}_i\mathbf{d}^{t+1} + \mathbf{b}_i^0) + \mathbf{t}^{t+1})\|_2^2. \tag{26}$$

These two steps are repeated alternatively until convergence. The first step can be computed by solving a linear system. The second step can be solved using [11] or by linearizing the rotation matrix. For tracking, another 2D matching energy can be added to the system:

$$E_{\text{match}}(Z^{t+1}) = \sum_{i=1}^{n} \|\mathbf{J}_t(\mathbf{f}(\mathbf{z}_i^t)) - \mathbf{J}_{t+1}(\mathbf{f}(\mathbf{z}_i^{t+1}))\|_2^2. \tag{27}$$

This optical flow energy enforces color consistency over time by measuring the variation of color from the previous image frame $\mathbf{J}_t$ to the current frame $\mathbf{J}_{t+1}$ for each $\mathbf{z}_i$.

3 Robust Registration

In registration, outliers are not only introduced by corrupted sensor measurements, but also by partial overlaps - many samples on the source simply do not have an ideal corresponding point on the target shape. To address this problem, various techniques rely on a set of heuristics to either *prune* or *downweigh* low quality correspondences. Typical criteria include discarding correspondences that are too far from each other, have dissimilar normals, or involve points on the boundary of the geometry; see [21] for details. As we will see next these heuristics are related to the optimization of robust functions. In this section we will consider robust functions as alternatives to the Euclidean metric and introduce a suitable optimization technique to use them efficiently.

In previous sections, we always considered an energy composed by terms like $\varphi(\epsilon(\mathbf{p}))$, where $\varphi(\epsilon) = \epsilon^2$ and $\epsilon(\mathbf{p})$ is the euclidean norm of the *residual* vector with parameters $\mathbf{p}$. This *squared Euclidian distance* metric is ideal for the data corrupted by Gaussian noise as it is the *maximum-likelyhood* solution of the problem [7, Sec. 7.1.1]. However, it is not robust to outliers which are common in real world data acquired by RGB-D devices.

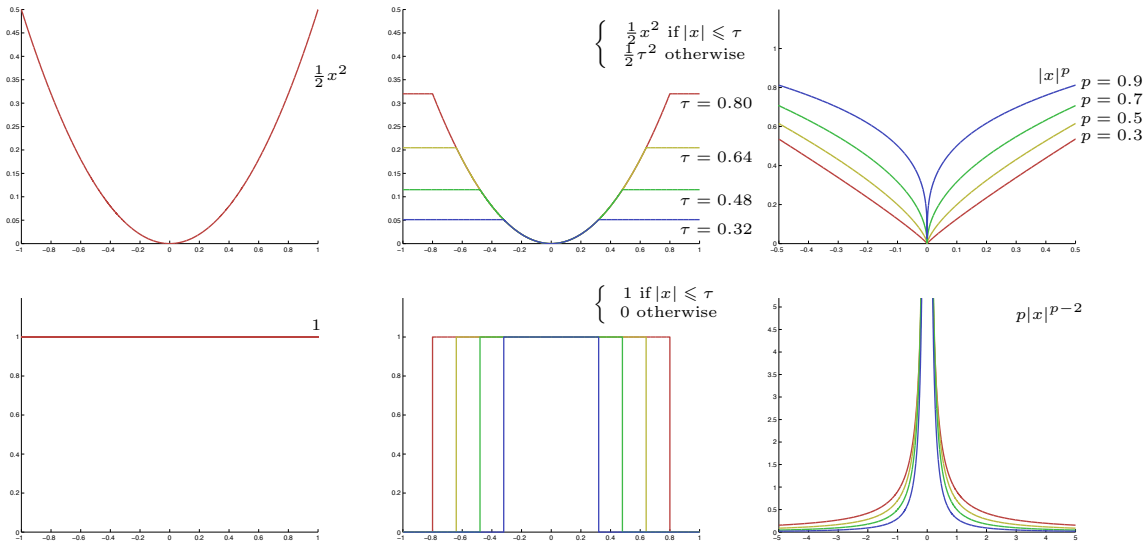


Figure 3: (top) The robust norms φ. (bottom) The associated weight functions w.

In registration, robustness can be obtained by exploiting robust functions [17]. In this framework, $\varphi(\epsilon)$ acts as a "penalty" function – a function measuring the influence that a certain residual has in the optimization. Given one of these functions, our robust optimization can be expressed as

$$\arg\min_{\mathbf{p}} \quad \sum_{i=1}^{n} \varphi(\epsilon_i(\mathbf{p})). \tag{28}$$

In Fig. 3 we show a few examplar commonly used penalty functions, note how these all posses properties like radial monotonicity and symmetry [13]. This optimization problem in Equation 28 can be solved using *Iteratively Re-Weighted Least Squares (IRLS)*

by solving a sequence of problems of the form

$$\arg\min_{\mathbf{p}} \quad \sum_{i=1}^{n} \alpha_i \, \epsilon_i(\mathbf{p})^2. \tag{29}$$

To understand how to compute the weights α_i first notice that the optima of Eq. 28 can be obtained by vanishing its gradient, which can be computed by a simple application of the chain rule (note we only look at one element of the sum)

$$\frac{\partial \varphi(\epsilon(\mathbf{p}))}{\partial \mathbf{p}} = \psi(\epsilon(\mathbf{p})) \frac{\partial \epsilon(\mathbf{p})}{\partial \mathbf{p}} = w(\epsilon(\mathbf{p}))\epsilon(\mathbf{p}) \frac{\partial \epsilon(\mathbf{p})}{\partial \mathbf{p}}, \tag{30}$$

where $\psi(x) = \partial \varphi(x)/\partial x$ for compactness of notation and $w(x) = \psi(x)/x$ is the so called *weighting function*. Interestingly, the gradient of Eq. 29 is

$$\frac{\partial \alpha \, \epsilon(\mathbf{p})^2}{\partial \mathbf{p}} = \alpha \, \epsilon(\mathbf{p}) \frac{\partial \epsilon(\mathbf{p})}{\partial \mathbf{p}}. \tag{31}$$

We can now see that by setting $\alpha = w(\epsilon(\mathbf{p}))$ the two gradients become equal. However, as the optimal weights $\alpha_i^* = w(\epsilon_i(\mathbf{p}^*))$ are not available, we use an iterative approach where at each iteration the weights are computed using the previous iteration

$$\arg\min_{\mathbf{p}^{t+1}} \sum_{i=1}^{n} w(\epsilon_i(\mathbf{p}^t))\epsilon_i(\mathbf{p}^{t+1})^2. \tag{32}$$

This scheme is know as *Iteratively Re-Weighted Least Squares (IRLS)* and is related to majorization-minimization. The basic idea of majorization-minimization is to iteratively minimize a function always larger or equal to the objective function and with at least one point in common. If these requirements are fullfilled the algorithm converges to a minimum [25].

Trimmed Metrics. Discarding unreliable correspondences is undoubtedly the simplest and most common way of dealing with outliers [21]. This can as well be formulated by Eq. 28, as it corresponds to a weight function like the one in Fig. 3 (bottom-middle) whose corresponding penalty function is a truncated squared euclidean norm Fig. 3 (top-middle). Even though this is trivial to implement, the local support of the weight function is problematic: if the souce surface is too far from the target surface the registration process will not proceed as all the weights would be zero valued. A possible solution is to *dynamically* adapt the threshold value by analyzing the distribution of residuals. For example, when the ratio of outliers versus inliers is known a priori, then the threshold can be readily estimated [9].

Sparse Metrics. The shortcomings of trimmed metrics can be overcome by considering sparse metrics. The penalty functions for sparse metrics take the form $\varphi(\epsilon) = |\epsilon|^p$, see Fig. 3 (bottom-right). An important observation is that the weight functions of p-norms tend to infinity as we approach zero giving a very large reward to inliers. Moreover, contrary to trimmed metrics, p-norms weakly penalize outliers leading to a more stable approach when target and source are far apart. This metric has been demonstrated successful in [5].

4 Conclusion

In this course, we introduced 2D/3D registration algorithms and show their applications for data captured with RGBD devices, such as the Microsoft Kinect or Asus Xtion Live. Image and geometry registration algorithms are an essential component of many computer graphics and computer vision systems. With recent technological advances in RGB-D sensors, robust algorithms that combine 2D image and 3D geometry registration have become an active area of research. The goal of this course was to introduce the basics of 2D/3D registration algorithms and to provide theoretical explanations and practical tools to design robust computer vision and computer graphics systems based on RGBD devices. We have shown that 2D and 3D registration can be expressed and combined in a common framework. Numerous application based on RGB-D devices can benefit from this formulation that allows to combine different priors in an easy manner. To illustrate the theory and demonstrate practical relevance, we briefly discuss three applications: rigid scanning, non-rigid modeling, and realtime face tracking.

References

[1] P. Besl and H. McKay. A method for registration of 3d shapes. *PAMI*, 1992.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. Proc. of ACM SIGGRAPH, 1999.

[3] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. Primo: coupled prisms for intuitive surface modeling. SGP, 2006.

[4] S. Bouaziz, M. Deuss, Y. Schwartzburg, T. Weise, and M. Pauly. Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum*, 2012.

[5] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. *SGP*, 2013.

[6] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Trans. Graph.*, 2013.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[8] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, 1991.

[9] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548. IEEE, 2002.

[10] T. Cootes and C. Taylor. Statistical models of appearance for computer vision, 2000.

[11] D. W. Eggert, A. Lorusso, , and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 1997.

[12] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, 1978.

[13] J. Fox. *An R and S-Plus companion to applied regression*. Sage, 2002. http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-robust-regression.pdf.

[14] B. K. P. Horn and B. G. Schunck. "determining optical flow". *Artif. Intell.*, 1981.

[15] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*, 2009.

[16] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. IJCAI, 1981.

[17] M. Mirza and K. Boyer. Performance evaluation of a class of m-estimators for surface parameter estimation in noisy range data. *IEEE Transactions on Robotics and Automation*, 9:75–85, 1993.

[18] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. ISMAR, 2011.

[19] I. Oikonomidis, N. Kyriazis, and A. Argyros. Tracking the articulated motion of two strongly interacting hands. CVPR, 2012.

[20] S. Rusinkiewicz. Derivation of point to plane minimization, 2013. `http://www.cs.princeton.edu/~smr/papers/icpstability.pdf`.

[21] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. *3DIM*, 2001.

[22] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. SGP, 2007.

[23] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 2007.

[24] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *TVCG*, 2012.

[25] P. Verboon. Majore ation wtthiteratively reweighted least squares: A general approach to optimize a class of resistant loss functions.

[26] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, 2012.

[27] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans. Graph.*, 2011.

[28] T. Weise, B. Leibe, and L. V. Gool. Accurate and robust registration for in-hand modeling. CVPR, 2008.

[29] T. Weise, T. Wismer, B. Leibe, and L. Van Gool. In-hand scanning with online loop closure. 3DIM, 2009.