Modern Techniques and Applications for Real-Time Non-rigid Registration

Sofien Bouaziz	Andrea Tagliasacchi	Hao Li	Mark Pauly
EPFL	University of Victoria	USC/ICT	EPFL

Abstract. Registration algorithms are an essential component of many computer graphics and computer vision systems. With recent technological advances in RGB-D sensors (color plus depth), an active area of research is in techniques combining color, geometry, and learnt priors for robust real-time registration. The goal of this course is to introduce the mathematical foundations and theoretical explanation of registration algorithms, in addition to the practical tools to design systems that leverage information from RGBD devices. We present traditional methods for correspondence computation derived from geometric first principles, along with modern techniques leveraging pre-processing of annotated datasets (e.g. deep neural networks). To illustrate the practical relevance of the theoretical content, we discuss applications including static and dynamic scanning/reconstruction as well as real-time tracking of hands and faces. An up-to-date version of the course notes, as well as slides and source code can be found at `http://gfx.uvic.ca/teaching/registration`.

Course Syllabus (SIGGRAPH Asia'16)

1.1 (5min, Tagliasacchi) Introduction, motivation and sensing hardware

1.2 (20min, Tagliasacchi) Hausdorff distances, rigid registration, ICP

1.3 (20min, Tagliasacchi) Robust registration, articulated registration

(10 minutes break)

2.1 (20min, Li) Non-rigid registration and face tracking

2.2 (20min, Li) Correspondences with Convolutional Neural Networks

2.3 (10min, Li) Conclusions and Q&A

SA '16 Courses, December 05-08, 2016, Macao. ACM 978-1-4503-4538-5/16/12 `http://dx.doi.org/10.1145/2988458.2988490`

About the course notes

Previous versions of this course have been offered at SIGGRAPH 2013, EG 2014 and SGP 2015. These authors have all contributed to the creation of these course notes:

Dr. Sofien Bouaziz (me@sofienbouaziz.com, `http://sofienbouaziz.com`**)**
obtained his PhD degree in 2015 in the Computer Graphics and Geometry Laboratory (LGG) at EPFL. He received his MSc degree in Computer Science from EPFL in 2009. His research interests include computer graphics, computer vision, and machine learning. In 2012, he co-founded faceshift, an EPFL spin-off that brings high-quality markerless facial motion capture to the consumer market.

Dr. Andrea Tagliasacchi (ataiya@uvic.ca, `http://gfx.uvic.ca`**)**
is an assistant professor at the University of Victoria and PI on the NSERC Discovery grant "Real-Time Modeling and Registration of Dynamic Geometry". He was a post doctoral scholar at EPFL, and obtained his PhD from Simon Fraser University as an NSERC Alexander Graham Bell scholar. He received his MSc from Politecnico di Milano (cum laude, faculty gold medalist).

Dr. Hao Li (hao@hao-li.com, `http://hao.li`**)**
is an assistant professor at the University of Southern California, Director of the Vision and Graphics Lab at the USC Institute for Creative Technologies, and CEO of Pinscreen. He was a postdoctoral researcher at Columbia and Princeton and a research lead at Industrial Light&Magic. He obtained his PhD from ETH Zurich and his MSc degree from the University of Karlsruhe.

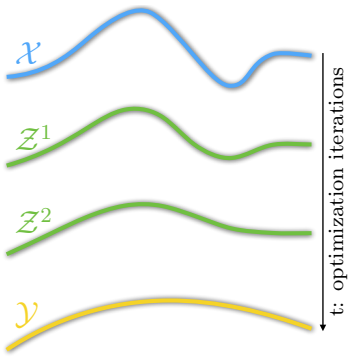
Dr. Mark Pauly (mark.pauly@epfl.ch, `http://lgg.epfl.ch`**)**
is a professor of computer science at EPFL. Prior to joining EPFL he was an assistant professor at ETH Zurich and a post doctoral scholar at Stanford. He received his Ph.D. degree in 2003 from ETH Zurich. His research interests include computer graphics and animation, geometry processing, and architectural design.

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 [44] and human motion tracking [39]. 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 geometry (3D) and image (2D) registration is one important aspect in the design of robust applications based on RGB-D devices. This course introduces the main concepts of 2D and 3D registration and explains how to combine them efficiently. To enable dense correspondence computation and non-rigid registration between shapes of significant deformations and shape variations, we present a deep learning framework based on convolutional neural networks.

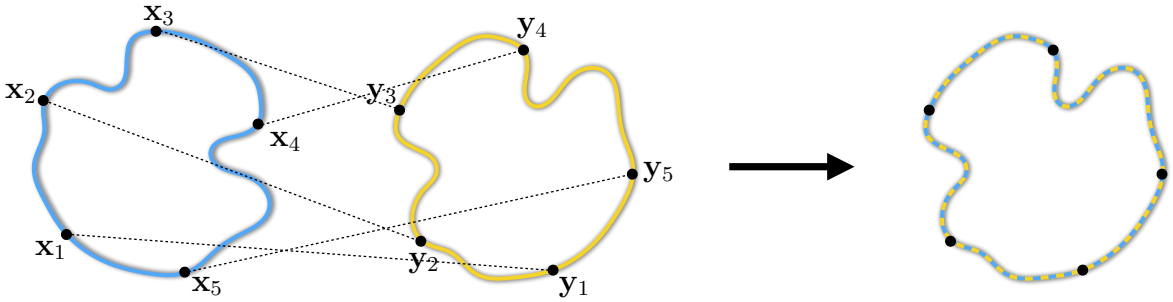
1 Fundamentals of Registration

In pairwise registration we desire to align a source surface $\mathcal{X}$ to a target surface $\mathcal{Y}$. 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 $X = \{\mathbf{x}_n \in \mathcal{X}, n = 1 \dots N\}$; different sampling strategies can be found in [32]. The matrix X, of dimensions $D \times N$, will represent a set of points, the point $\mathbf{x}_n$ is contained in its n-th column.



We start by describing a technique capable to estimate the (rigid) transformation between $\mathcal{X}$ and $\mathcal{Y}$ as far as a few correct correspondences are given in input. We then introducine the ICP algorithm that leverages object proximity to automatically compute registration correspondences.

Shape Matching Problem Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ and $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ be two set of *corresponding* points in $\mathbb{R}^D$. The set X is rigidly transformed into $Z = \mathbf{R}X + \mathbf{t}$ by a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$.



The rigid transformation that optimally aligns the two point sets is the solution of the least-squares optimization problem:

$$(\mathbf{R}, \mathbf{t}) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{n=1}^{N} \|(\mathbf{R}\mathbf{x}_n + \mathbf{t}) - \mathbf{y}_n\|_2^2 \tag{1}$$

As derived in [35], the optimal solution is computed by arranging the points in two $3 \times N$ matrices $\bar{X}$ and $\bar{Y}$ whose columns $\bar{x}_n$ and $\bar{y}_n$ are:

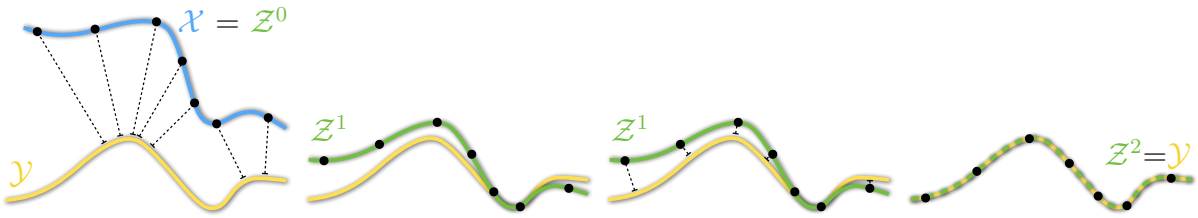
$$\bar{\mathbf{x}}_n = \mathbf{x}_n - \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n, \qquad \bar{\mathbf{y}}_n = \mathbf{y}_n - \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n \tag{2}$$

By computing the singular value decomposition of the $D \times D$ covariance matrix $\bar{X}\bar{Y}^T$, i.e. $U\Sigma V^T = \text{SVD}(\bar{X}\bar{Y}^T)$, the optimal rotation and translation are given respectively by

$$\mathbf{R} = V \begin{pmatrix} 1 & & \\ & 1 & \\ & & \det(VU^T) \end{pmatrix} U^T, \qquad \mathbf{t} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n - \mathbf{R}\frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \tag{3}$$

As we are only interested in rotations, the term $\det(VU^T)$ factors out reflections as potential solutions of our optimization [35, Sec.3].

Iterative Closest Point (ICP) Given correct pairwise correspondences, the shape matching problem computes the optimal (rigid) transformation between two object. However, in most situations ground truth correspondences are not available. The Iterative Closest Point algorithm (ICP) addresses this problem through local optimization. In the *first* step, given $\mathcal{Z}$ and samples $\mathbf{z}_i \in \mathcal{Z}$, we compute $\mathbf{y}_i = \Pi_{\mathcal{Y}}(\mathbf{z}_i)$, the *closest point* correspondence of $\mathbf{z}_i$ onto $\mathcal{Y}$. In the *second* step, these correspondences are used to solve the shape matching problem; see Eq. 1. This process is iterated until the optimization converges to a *local* minima. The fundamental assumption made by ICP is that the surfaces are in rough initial alignment, therefore closest point correspondences approximate ground truth correspondences.



Derivation of ICP Given a source surface $\mathcal{Z}$, a target surface $\mathcal{Y}$, we introduce a matching energy measuring the proximity of $\mathcal{Z}$ to $\mathcal{Y}$. The metric $\varphi(\mathbf{z}, \mathcal{Y})$ measures the distance between a point $\mathbf{z}$ and the surface $\mathcal{Y}$. Also, as we want to numerically optimize this energy, the integral is discretized by sampling $\mathcal{Z}$:

$$E_{\text{match}}(\mathcal{Z}) = \int_{\mathcal{Z}} \varphi(\mathbf{z}, \mathcal{Y}) d\mathbf{z} \approx \sum_{n=1}^{N} \varphi(\mathbf{z}_n, \mathcal{Y}) \tag{4}$$

We then re-write the metric φ by expressing it as the solution of an optimization problem measuring the distance between $\mathbf{z}_n$ and the *closest* point $\mathbf{y}_n$ on the surface $\mathcal{Y}$:

$$\varphi(\mathbf{z}_n, \mathcal{Y}) = \min_{\mathbf{y} \in \mathcal{Y}} \varphi(\mathbf{z}_n, \mathbf{y}), \quad \mathbf{y}_n = \Pi_{\mathcal{Y}}(\mathbf{z}_n) = \arg\min_{\mathbf{y} \in \mathcal{Y}} \varphi(\mathbf{z}_n, \mathbf{y}) \tag{5}$$

For simplicity, we use the squared Euclidian distance as our metric $\varphi(\mathbf{z}, \mathbf{y}) = \|\mathbf{z} - \mathbf{y}\|_2^2$; see Sec. 3 for other metrics. By introducing of a set of auxiliary variables Y, and remembering how $\mathcal{Z} = \mathbf{R}\mathcal{X} + \mathbf{t}$, we can rewrite our rigid registration problem as:

$$\arg\min_{\mathbf{R},\mathbf{t},Y} \sum_{n=1}^{N} \|(\mathbf{R}\mathbf{z}_n + \mathbf{t}) - \mathbf{y}_n\|_2^2 \tag{6}$$

Our problem can then be solved by alternating optimization:

$$\arg\min_{Y} \sum_{n=1}^{N} \|(\mathbf{R}\mathbf{x}_n + \mathbf{t}) - \mathbf{y}_n\|_2^2, \qquad \arg\min_{\mathbf{R},\mathbf{t}} \sum_{n=1}^{N} \|(\mathbf{R}\mathbf{x}_n + \mathbf{t}) - \mathbf{y}_n\|_2^2 \tag{7}$$

In the first step, we optimize for closest point correspondences (Eq. 5), while in the second step we optimize for the optimal transformation (Eq. 1). This alternating optimization is iterated until convergence to a local minima.

2 Image/Geometry Registration Framework

In registration, the alignment of a source model onto a target model can be rigid or non-rigid depending on the type of object being scanned. We formulate the registration as the minimization of a compound energy:

$$E_{\text{reg}} = E_{\text{match}} + E_{\text{prior}}. \tag{8}$$

The matching energy E_{match} measures the proximity of source to 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 8 provides a unified way to formulate both image and geometry registration, which simplifies their integration.

2.1 Geometry 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 $\mathbb{R}^3$. We recall how $\mathcal{Z}$ that is a transformed or deformed version of $\mathcal{X}$ that eventually aligns with $\mathcal{Y}$.

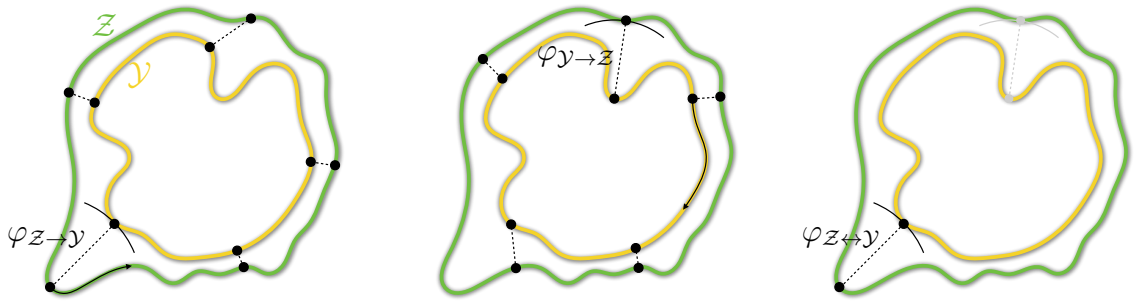
2.1.1 Matching energy

The matching energy measures how close the surface $\mathcal{Z}$ is to the surface $\mathcal{Y}$. The similarity of two geometric models is measured by the (symmetric) Hausdorff distance/metric, $d_{\mathcal{Z}\leftrightarrow\mathcal{Y}}$ obtained as the maximum of two asymmetric terms:

$$\varphi_{\mathcal{Z}\to\mathcal{Y}} = \max_{\mathbf{z}\in\mathcal{Z}} [\min_{\mathbf{y}\in\mathcal{Y}} \varphi(\mathbf{z},\mathbf{y})] \tag{9}$$

$$\varphi_{\mathcal{Y}\to\mathcal{Z}} = \max_{\mathbf{y}\in\mathcal{Y}} [\min_{\mathbf{z}\in\mathcal{Z}} \varphi(\mathbf{z},\mathbf{y})] \tag{10}$$

$$\varphi_{\mathcal{Z}\leftrightarrow\mathcal{Y}} = \max\{\varphi_{\mathcal{Z}\to\mathcal{Y}}, \varphi_{\mathcal{Y}\to\mathcal{Z}}\} \tag{11}$$



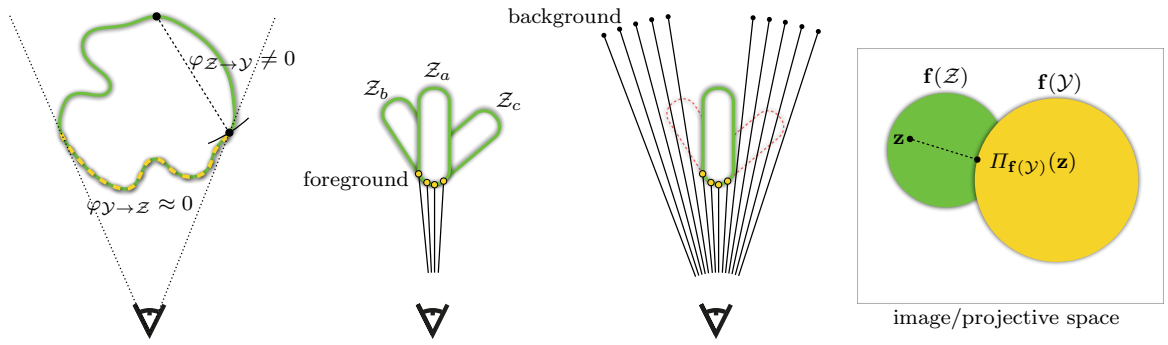


Figure 1: (a) The Hausdorff distance is non-zero in monocular acquisition, even when $\mathcal{Z}$ and $\mathcal{Y}$ are in perfect alignment. (b) Three different configurations of $\mathcal{Z}$ equivalently minimize the data-to-model residuals. (c) Only $\mathcal{Z}_a$ is optimal as it does not cross the background view rays. (d) These constraints can be encoded by minimizing model-to-data correspondences in 2D projective space space.

In computer graphics, the Hausdorff distance is often used in multi-resolution modeling to measure the difference between two different representations of the same 3D object [14]. Typically, squared euclidean distances are employed as metrics $\varphi(\mathbf{z}, \mathbf{y}) = \|\mathbf{z} - \mathbf{y}\|_2^2$.

Differentiable Hausdorff To derive a differentiable Hausdorff metric, we replace the max operator in Eq. 9 and Eq. 10 by an integral, and the one in Eq. 11 by a sum:

$$\varphi_{\mathcal{Z}\leftrightarrow\mathcal{Y}} \approx \int_{\mathbf{z}\in\mathcal{Z}} \min_{\mathbf{y}\in\mathcal{Y}} \|\mathbf{z} - \mathbf{y}\|_2^2 \, d\mathbf{z} \; + \; \int_{\mathbf{y}\in\mathcal{Y}} \min_{\mathbf{z}\in\mathcal{Z}} \|\mathbf{z} - \mathbf{y}\|_2^2 \, d\mathbf{y} \tag{12}$$

We can further simplify this expression by leveraging the projection operator introduced in Eq. 5, and replace integrals by discrete sums, reducing our Hausdorff metric to an ICP-like problem with two-way correspondences [42]:

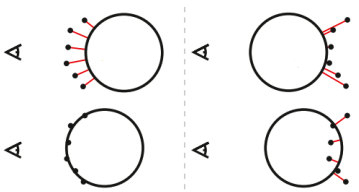
$$\varphi_{\mathcal{Z}\leftrightarrow\mathcal{Y}} \approx \underbrace{\sum_{\mathbf{z}\in Z} \|\mathbf{z} - \Pi_{\mathcal{Y}}(\mathbf{z})\|_2^2}_{\text{model-to-data}} + \underbrace{\sum_{\mathbf{y}\in Y} \|\mathbf{y} - \Pi_{\mathcal{Z}}(\mathbf{y})\|_2^2}_{\text{data-to-model}} \tag{13}$$

Monocular Hausdorff When the sensor data $\mathcal{Y}$ has been measured by a monocular acquisition system, the metric in Eq. 13 suffers a fundamental limitation: even if the digital model is in perfect alignment with the data, that is when $\varphi_{\mathcal{Y}\to\mathcal{Z}} = 0$, overall our metric might be non-zero as $\varphi_{\mathcal{Z}\to\mathcal{Y}} \neq 0$; see Fig. 1a We can resolve this problem by computing the $\varphi_{\mathcal{Z}\to\mathcal{Y}}$ in projective/camera space (2D) rather than in 3D. To achieve this, we first rasterize our digital model in the camera coordinate system and generate a *silhouette* image $\mathbf{f}(Z)$; see Fig. 1d. We then compare this silhouette to the silhouette from the sensor $\mathbf{f}(Y)$. Our $\varphi_{\mathcal{Z}\to\mathcal{Y}}$ is then effectively re-written as a 2D ICP registration energy:

$$\varphi_{\mathcal{Z}\leftrightarrow\mathcal{Y}} \approx \underbrace{\sum_{\mathbf{z}\in\mathbf{f}(Z)} \|\mathbf{z} - \Pi_{\mathbf{f}(Y)}(\mathbf{z})\|_2^2}_{\text{model-to-data (2D)}} + \underbrace{\sum_{\mathbf{y}\in Y} \|\mathbf{y} - \Pi_{\mathcal{Z}}(\mathbf{y})\|_2^2}_{\text{data-to-model (3D)}} \tag{14}$$

The function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^2$ projects a 3D point to the sensor's image plane.

Monocular Correspondences Fast motion and monocular acquisition pose a particular challenge to iterative registration algorithms. In the inset figure we illustrate two scenarios. In the first, motion is small and closest-point correspondences register our model to input data effectively. In the second, fast motion results in correspondences that map to the back of the model, resulting in a local minima of registration. Note that to solve this problem it is not sufficient to discard correspondences whose normals are back facing [32]. Instead, we can overcome this problem by computing closest point correspondences to a model that has been *backface culled* [39]. Alternatively, the normals of $\mathbf{z}_n$ can be included in the registration energy to *penalize* correspondences mapping to the back of the model; however, such an approach is only suitable for algorithms performing a *joint* optimization over $\mathcal{Z}$ and its correspondences to $\mathcal{Y}$ [40].



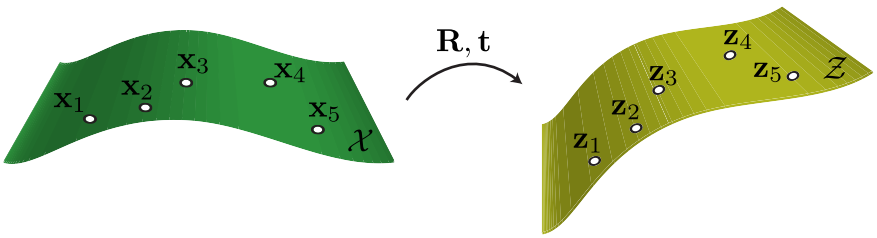
2.1.2 Prior energy

We now discuss 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

$$E_{\text{rigid}}(Z, \mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2, \tag{15}$$

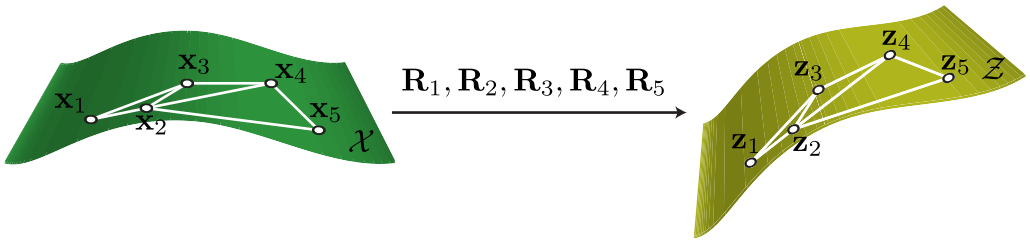
where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ a translation vector. In this case, the deformed surface $\mathcal{Z}$ tries to follow a rigid transformation of the original surface $\mathcal{X}$.



Local rigidity. The local rigidity energy, following [36, 8], can be expressed as:

$$E_{\text{arap}}(Z, \mathbf{R}_{\{1..n\}}) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j - \mathbf{z}_i) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_i)\|_2^2, \tag{16}$$

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 [7, 37].



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 $\mathbf{m}$ [15, 49, 39]. A shape $\mathbf{s}$ can be defined as:

$$\mathbf{s} = \mathbf{P}\mathbf{d} + \mathbf{m}, \tag{17}$$

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

$$E_{\text{prior}}(Z, \mathbf{d}) = \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i)\|_2^2, \tag{18}$$

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.

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 [50, 51, 27]. 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

$$\begin{aligned} E(Z, \mathbf{R}, \mathbf{t}) &= w_1 E_{\text{match}} + w_2 E_{\text{rigid}} \\ E_{\text{match}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - \Pi_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 \\ E_{\text{rigid}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2 \end{aligned} \tag{19}$$

where the matching energy is combined with a global rigidity prior. To optimize $E(Z, \mathbf{R}, \mathbf{t})$ we linearize the rotation matrix approximating $\cos\theta \approx 1$ and $\sin\theta \approx \theta$ [30]:

$$\mathbf{R} \approx \tilde{\mathbf{R}} = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix}. \tag{20}$$

The alignment is computed by solving iteratively

$$\underset{Z^{t+1}, \tilde{\mathbf{R}}, \tilde{\mathbf{t}}}{\arg\min} \sum_{i=1}^{n} w_1 \|\mathbf{z}_i^{t+1} - \Pi_{\mathcal{Y}}(\mathbf{z}_i^t)\|_2^2 + w_2 \|\mathbf{z}_i^{t+1} - (\tilde{\mathbf{R}}(\mathbf{R}^t\mathbf{x}_i + \mathbf{t}^t) + \tilde{\mathbf{t}})\|_2^2, \tag{21}$$

where t is the iteration number and $\mathbf{z}_i^0 = \mathbf{x}_i$. As $\Pi_{\mathcal{Y}}(.)$ is a non linear function that is difficult to optimize with, we use in the optimization the previous estimate $\Pi_{\mathcal{Y}}(\mathbf{z}_i^t)$. This correspond to the *point-to-point* matching error [3]. To speed up the convergence of the optimization [28] one can linearize $\|\mathbf{z}_i^{t+1} - \Pi_{\mathcal{Y}}(\mathbf{z}_i^t)\|_2$ at $\Pi_{\mathcal{Y}}(\mathbf{z}_i^t)$ which gives $\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - \Pi_{\mathcal{Y}}(\mathbf{z}_i^t))$, where $\mathbf{n}_i$ is the normal of the surface $\mathcal{Y}$ at $\Pi_{\mathcal{Y}}(\mathbf{z}_i^t)$. This leads to the *point-to-plane* matching error [12]. The optimization can be reformulated as

$$\underset{Z^{t+1}, \tilde{\mathbf{R}}, \tilde{\mathbf{t}}}{\arg\min} \sum_{i=1}^{n} w_1 \left[\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - \Pi_{\mathcal{Y}}(\mathbf{z}_i^t))\right]^2 + w_2 \|\mathbf{z}_i^{t+1} - (\tilde{\mathbf{R}}(\mathbf{R}^t\mathbf{x}_i + \mathbf{t}^t) + \tilde{\mathbf{t}})\|_2^2. \tag{22}$$

Both Equation 21 and Equation 22 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.

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

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - \Pi_{\mathcal{Y}}(\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2. \tag{23}$$

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 [16].

Non-rigid registration Registering a shape template towards a scanned 3D object allows to obtain a complete and clean 3D mesh [22]. An example is given below in the context of face modeling. In this case, the morphable model of Blanz and Vetter [4] that

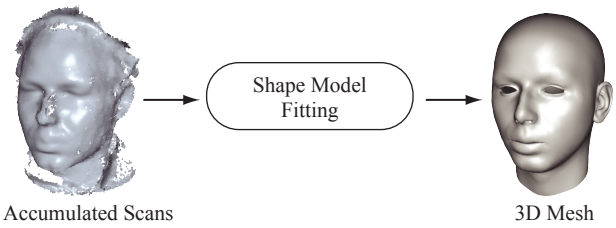


Figure 2: Registration of a morphable model towards the scanned face.

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

$$E(Z, \mathbf{d}, \mathbf{R}_i|_{i=1}^n, \mathbf{R}, \mathbf{t}) = w_1 E_{\text{match}} + w_2 E_{\text{rigid}} + w_3 E_{\text{model}} + w_4 E_{\text{arap}} \tag{24}$$

$$\begin{aligned} E_{\text{match}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - \Pi_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 \\ E_{\text{rigid}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}\mathbf{x}_i + \mathbf{t})\|_2^2 \\ E_{\text{model}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i)\|_2^2 \\ E_{\text{arap}} &= \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j - \mathbf{z}_i) - \mathbf{R}_i(\mathbf{x}_j - \mathbf{x}_i)\|_2^2 \end{aligned} \tag{25}$$

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

$$\begin{aligned} \underset{Z^{t+1}, \mathbf{d}, \tilde{\mathbf{R}}_i|_{i=1}^n, \tilde{\mathbf{R}}, \tilde{\mathbf{t}}}{\arg\min} \sum_{i=1}^{n} w_1(\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - \Pi_{\mathcal{Y}}(\mathbf{z}_i^t)))^2 + w_2\|\mathbf{z}_i^{t+1} - (\tilde{\mathbf{R}}(\mathbf{R}^t\mathbf{x}_i + \mathbf{t}^t) + \tilde{\mathbf{t}})\|_2^2 + \\ w_3\|\mathbf{z}_i^{t+1} - (\mathbf{P}_i\mathbf{d} + \mathbf{m}_i)\|_2^2 + w_4 \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j^{t+1} - \mathbf{z}_i^{t+1}) - \tilde{\mathbf{R}}_i\mathbf{R}_i^t(\mathbf{x}_j - \mathbf{x}_i)\|_2^2, \end{aligned} \tag{26}$$

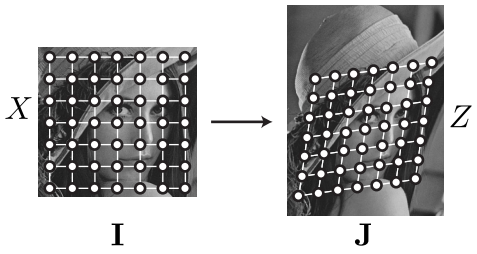
which corresponds to solving a linear system.

2.2 Image Registration

In image 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

$$E_{\text{match}}(Z) = \sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{z}_i)\|_2^2. \tag{27}$$

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 [25] the deformation is assumed to be constant within a patch around each pixel. This corresponds to the prior energy

$$E_{\text{LK}}(Z) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \|(\mathbf{z}_j - \mathbf{x}_j) - (\mathbf{z}_i - \mathbf{x}_i)\|_2^2, \tag{28}$$

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

Horn-Schunck. In the Horn-Schunck algorithm [19] the smoothness of the flow is defined using a Laplacian operator

$$E_{\text{HK}}(Z) = \sum_{i=1}^{n} \|(\mathbf{z}_i - \mathbf{x}_i) - |\mathcal{N}_i|^{-1} \sum_{j \in \mathcal{N}_i} (\mathbf{z}_j - \mathbf{x}_j)\|_2^2, \tag{29}$$

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 [19]. Our optimization energy is:

$$\begin{aligned} E(Z) &= w_1 E_{\text{match}} + w_2 E_{\text{HK}} \\ E_{\text{match}} &= \sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{z}_i)\|_2^2 \\ E_{\text{HK}} &= \sum_{i=1}^{n} \|(\mathbf{z}_i - \mathbf{x}_i) - |\mathcal{N}_i|^{-1} \sum_{j \in \mathcal{N}_i} (\mathbf{z}_j - \mathbf{x}_j)\|_2^2 \end{aligned} \tag{30}$$

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

$$\begin{aligned}\arg\min_{Z^{t+1}} \sum_{i=1}^{n} w_1 \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{z}_i^t) - \nabla\mathbf{J}(\mathbf{z}_i^t)^T(\mathbf{z}_i^{t+1} - \mathbf{z}_i^t)\|_2^2 + \\ w_2 \|(\mathbf{z}_i^{t+1} - \mathbf{x}_i) - |\mathcal{N}_i|^{-1} \sum_{j\in\mathcal{N}_i} (\mathbf{z}_j^{t+1} - \mathbf{x}_j)\|_2^2. \end{aligned} \tag{31}$$

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.

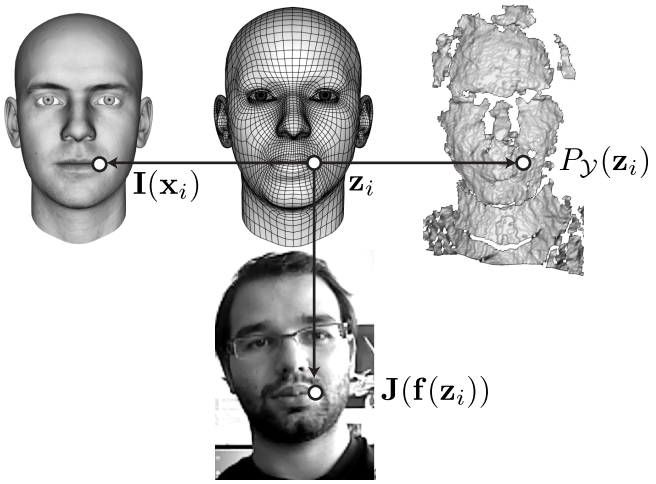
2.3 2D/3D Registration

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

$$E_{\text{match}}(Z) = \sum_{i=1}^{n} w_1 \|\mathbf{z}_i - \Pi_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 + w_2 \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i))\|_2^2. \tag{32}$$



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 $\mathbf{f}(\mathbf{z}_i) = \begin{bmatrix} f\mathbf{z}_{i,x}/\mathbf{z}_{i,z} & f\mathbf{z}_{i,y}/\mathbf{z}_{i,z}\end{bmatrix}^T$.

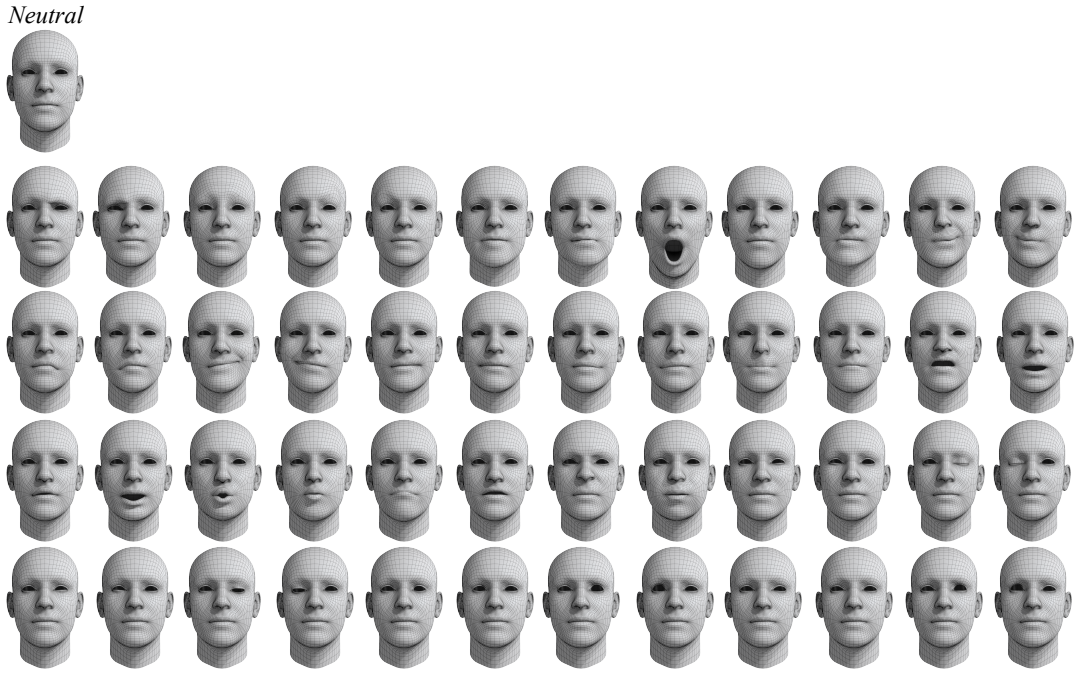


Figure 3: A blendshape model composed of 48 expressions.

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 [17]. Realtime face tracking using an RGB-D device can be formulated as a 2D/3D registration of the blendshape model to the 2D and 3D data [49]. The registration energy can be formulated as

$$\begin{aligned} E(Z, \mathbf{d}, \mathbf{R}, \mathbf{t}) &= w_1 E_{\text{match geometry}} + w_2 E_{\text{match color}} + w_3 E_{\text{model+rigid}} \qquad (33) \\ E_{\text{match geometry}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - \Pi_{\mathcal{Y}}(\mathbf{z}_i)\|_2^2 \\ E_{\text{match color}} &= \sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i))\|_2^2 \\ E_{\text{model+rigid}} &= \sum_{i=1}^{n} \|\mathbf{z}_i - (\mathbf{R}(\mathbf{B}_i\mathbf{d} + \mathbf{b}_i^0) + \mathbf{t})\|_2^2 \end{aligned}$$

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

$$\sum_{i=1}^{n} \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i^{t+1})\| \approx \|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i^t)) - \nabla\mathbf{J}(\mathbf{f}(\mathbf{z}_i^t))^T \frac{\partial f(\mathbf{z}_i^t)}{\partial \mathbf{z}_i}(\mathbf{z}_i^{t+1} - \mathbf{z}_i^t)\|_2^2. \qquad (34)$$

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

$$\frac{\partial f(\mathbf{z}_i)}{\partial \mathbf{z}_i} = \begin{bmatrix} \frac{f}{\mathbf{z}_{i,z}} & 0 & -\frac{f\mathbf{z}_{i,x}}{\mathbf{z}_{i,z}^2} \\ 0 & \frac{f}{\mathbf{z}_{i,z}} & -\frac{f\mathbf{z}_{i,y}}{\mathbf{z}_{i,z}^2} \end{bmatrix}. \tag{35}$$

In [49], 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

$$\begin{aligned} \underset{Z^{t+1},\mathbf{d}^{t+1}}{\arg\min} \sum_{i=1}^{n} w_1(\mathbf{n}_i^T(\mathbf{z}_i^{t+1} - \Pi_{\mathcal{Y}}(\mathbf{z}_i^t)))^2 + \\ w_2\|\mathbf{I}(\mathbf{x}_i) - \mathbf{J}(\mathbf{f}(\mathbf{z}_i^t)) - \nabla\mathbf{J}(\mathbf{f}(\mathbf{z}_i^t))^T \frac{\partial f(\mathbf{z}_i^t)}{\partial \mathbf{z}_i}(\mathbf{z}_i^{t+1} - \mathbf{z}_i^t)\|_2^2 + \\ w_3\|\mathbf{z}_i^{t+1} - (\mathbf{R}^t(\mathbf{B}_i\mathbf{d}^{t+1} + \mathbf{b}_i^0) + \mathbf{t}^t)\|_2^2, \end{aligned} \tag{36}$$

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

$$\underset{\mathbf{R}^{t+1},\mathbf{t}^{t+1}}{\arg\min} \sum_{i=1}^{n} \|\mathbf{z}_i^{t+1} - (\mathbf{R}^{t+1}(\mathbf{B}_i\mathbf{d}^{t+1} + \mathbf{b}_i^0) + \mathbf{t}^{t+1})\|_2^2. \tag{37}$$

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 [16] or by linearizing the rotation matrix. For tracking, another 2D matching energy can be added to the system:

$$E_{\text{match}}(Z^{t+1}) = \sum_{i=1}^{n} \|\mathbf{J}_t(\mathbf{f}(\mathbf{z}_i^t)) - \mathbf{J}_{t+1}(\mathbf{f}(\mathbf{z}_i^{t+1}))\|_2^2. \tag{38}$$

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 [32] 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 [10, Sec. 7.1.1]. However, it is not robust to outliers which are common in real world data acquired by RGB-D devices.

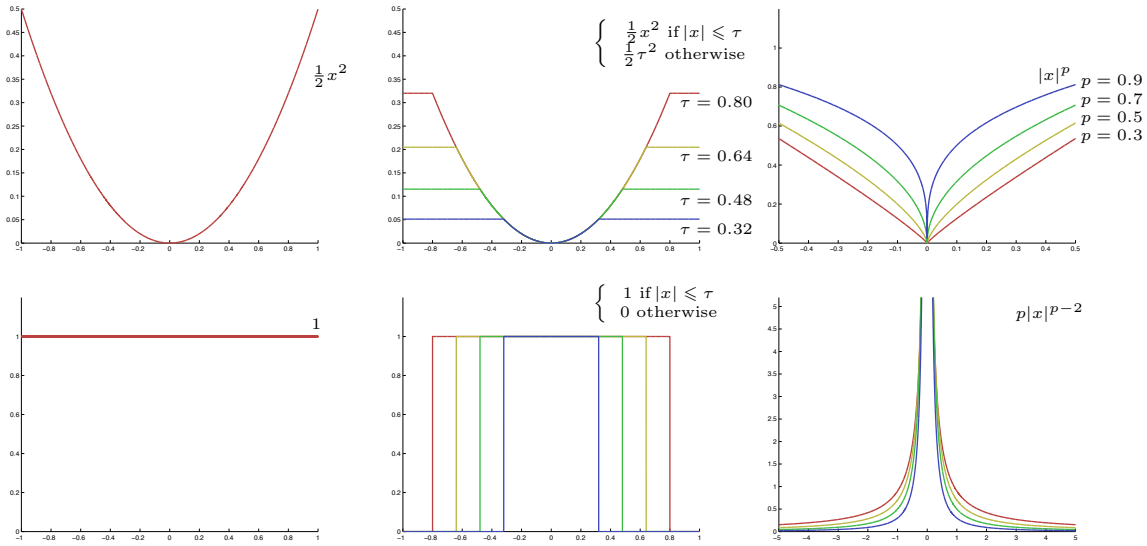


Figure 4: (top) The robust norms φ. (bottom) The associated weight functions w.

In registration, robustness can be obtained by exploiting robust functions [26]. 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

$$\arg\min_{\mathbf{p}} \quad \sum_{i=1}^{n} \varphi(\epsilon_i(\mathbf{p})). \tag{39}$$

In Fig. 4 we show a few examplar commonly used penalty functions, note how these all posses properties like radial monotonicity and symmetry [18]. This optimization problem in Equation 39 can be solved using *Iteratively Re-Weighted Least Squares (IRLS)* by solving a sequence of problems of the form

$$\arg\min_{\mathbf{p}} \quad \sum_{i=1}^{n} \alpha_i \, \epsilon_i(\mathbf{p})^2. \tag{40}$$

To understand how to compute the weights α_i first notice that the optima of Eq. 39 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)

$$\frac{\partial \varphi(\epsilon(\mathbf{p}))}{\partial \mathbf{p}} = \psi(\epsilon(\mathbf{p})) \frac{\partial \epsilon(\mathbf{p})}{\partial \mathbf{p}} = w(\epsilon(\mathbf{p}))\epsilon(\mathbf{p}) \frac{\partial \epsilon(\mathbf{p})}{\partial \mathbf{p}}, \tag{41}$$

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. 40 is

$$\frac{\partial \alpha \, \epsilon(\mathbf{p})^2}{\partial \mathbf{p}} = \alpha \, \epsilon(\mathbf{p}) \frac{\partial \epsilon(\mathbf{p})}{\partial \mathbf{p}}. \tag{42}$$

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

$$\arg\min_{\mathbf{p}^{t+1}} \sum_{i=1}^{n} w(\epsilon_i(\mathbf{p}^t))\epsilon_i(\mathbf{p}^{t+1})^2. \tag{43}$$

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 [45].

Trimmed Metrics. Discarding unreliable correspondences is undoubtedly the simplest and most common way of dealing with outliers [32]. This can as well be formulated by Eq. 39, as it corresponds to a weight function like the one in Fig. 4 (bottom-middle) whose corresponding penalty function is a truncated squared euclidean norm Fig. 4 (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 [13].

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. 4 (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 [9].

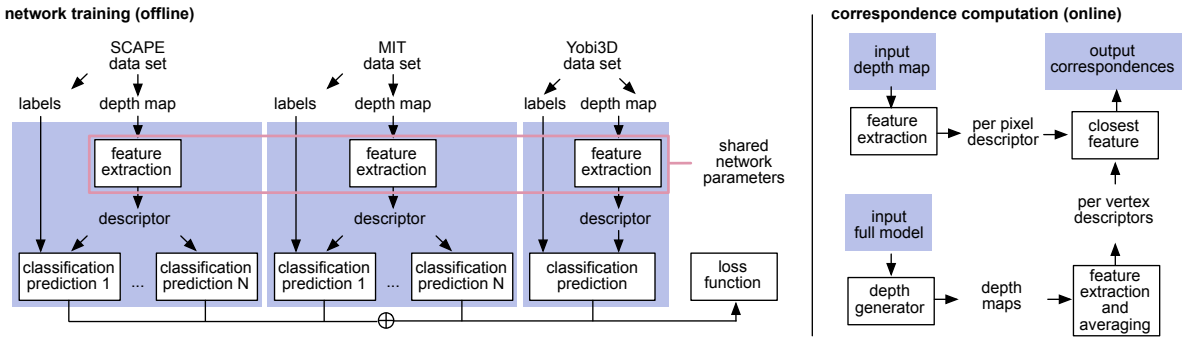


Figure 5: A neural network can be trained to extract a feature descriptor and predict the corresponding segmentation label on the human body surface for each point in the input depth maps. Per-vertex descriptors are generated for 3D models by averaging the feature descriptors in their rendered depth maps. The extracted features can then be used for example to compute dense correspondences.

4 Correspondence computation with Deep Learning

The success of registration techniques relying on closest-point correspondences generally relies on the deformation between source and target shapes being reasonably small, with sufficient overlap. While local shape descriptors [31] can be used to determine correspondences between surfaces that are far apart, they are typically sparse and prone to false matches and require manual clean-up. Dense correspondences between shapes with larger deformations can be obtained reliably using statistical models of human shapes [2, 6], but the subject has to be naked [5]. For clothed bodies, the automatic computation of dense mappings [20, 24, 29, 11] have been demonstrated on full surfaces with significant shape variations, but are limited to compatible or zero-genus surface topologies. In this section, we present a deep neural network technique to compute dense correspondences between shapes of clothed subjects in arbitrary complex poses [48]. The input surfaces can be a full model, a partial scan, or a depth map, maximizing the range of possible applications.

The system is trained with a large dataset of depth maps generated from the human bodies of the SCAPE database [2], as well as from clothed subjects of the Yobi3D [1] and MIT [46] dataset. Note all meshes in the SCAPE database are in full correspondence, while the clothed 3D body models are manually labeled. Similar to the unified embedding approach of FaceNet [33], the AlexNet [21] classification network can be used to learn distinctive feature vectors for different subregions of the human body. While the performance of this dense correspondence computation is comparable to state of the art techniques between two full models, learning shape priors of clothed subjects can yield highly accurate matches between partial-to-full and partial-to-partial shapes. The effectiveness of these correspondences is demonstrated in a template based performance capture application that uses a single RGB-D camera as input.

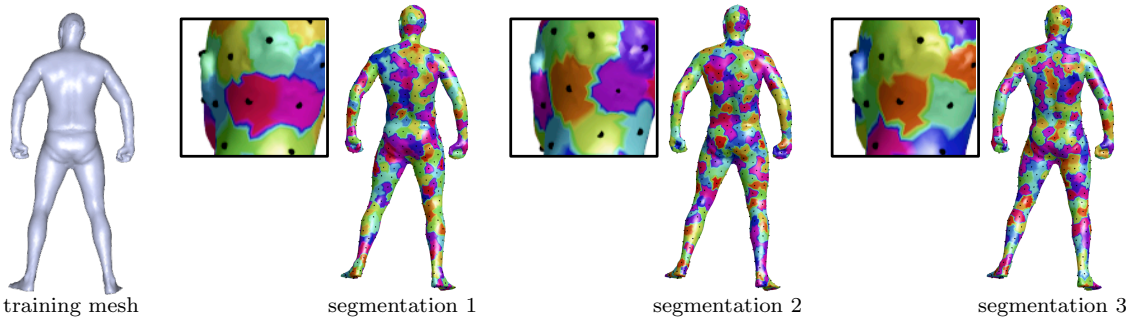


Figure 6: To ensure smooth descriptors, we define a classification problem for multiple segmentations of the human body. Nearby points on the body are likely to be assigned the samal label in at least one segmentation.

4.1 Classification Network

We now describe a deep learning framework to compute high-dimensional feature descriptors for classification tasks. Traditional classification neural networks tend to separate the embedding of surface points lying in different but nearby classes. Thus, using such learned feature descriptors for correspondence matching between deformed surfaces often results in significant outliers at the segmentation boundaries. Learning on repeated mesh segmentations can overcome this issue [48]. As a result, shape points that are geodesically close on the surface of their corresponding 3D model to nearby points in the feature space. Further, with this approach outliers are considerably reduced and the amount of necessary training data is much smaller compared to conventional learning methods. We can formulate the correspondence problem as a *classification* problem: first, a feature descriptor $\mathbf{f} : I \rightarrow R^d$ is learned. This descriptor maps each pixel in a *single* depth image I to a feature vector. These feature descriptors are then used to establish correspondences across depth maps; see Figure 5. The feature vector should satisfy two properties:

Inter-Subject: $\mathbf{f}$ depends only on the pixel's location on the human body, so that if two pixels are sampled from the same anatomical location on depth scans of two *different* humans, their feature vector should be nearly identical, irrespective of pose, clothing, body shape, and angle from which the depth image was captured.

Intra-Subject: $\|\mathbf{f}(p) - \mathbf{f}(q)\|$ is small when p and q represent nearby points on the *same* human body, and large for distant points.

The literature takes two different approaches to enforcing these properties when learning descriptors using convolutional neural networks. *Direct* methods promote these properties in the loss function (by using e.g. siamese or triplet-loss); however, it is not trivial how to sample a dense set of training pairs or triplets that can all contribute to training [33]. *Indirect* methods instead optimize the network architecture to perform classification. The indirect approach is effective since classification networks tend to assign similar (dissimilar) descriptors to the input points belonging to the same (different) class, and thus satisfy the above properties implicitly. The experiments in [48] suggest that an indirect method that uses an ensemble of classification tasks has better performance and computational efficiency.

4.2 Descriptor Learning as Ensemble Classification

There are two challenges to learning a feature descriptor for depth images of human models with an indirect approach. First, the training data is heterogenous: between different human models, only sparse set of key point correspondences are available, while for different poses of the same person, we have dense pixel-wise correspondences [2]. Second, smoothness of descriptors learned through classification is not explicitly enforced. Even though some classes tend to be closer to each other than the others in reality, the network treats all classes equally. To address both challenges, per-pixel descriptors can be learned for depth images by first training a network to solve a *group* of classification problems, using a *shared feature extraction tower*.

Formally, suppose there are M classification problems $C_i, 1 \leq i \leq M$. Denote the parameters to be learned in classification problem C_i as $(\mathbf{w}_i, \mathbf{w})$, where $\mathbf{w}_i$ and $\mathbf{w}$ are the parameters corresponding to the classification layer and descriptor extraction tower, respectively. The descriptor learning is defined as minimizing a combination of loss functions of all classification problems:

$$\{\mathbf{w}_i^\star\}, \mathbf{w}^\star = \arg\min_{\{\mathbf{w}_i\}, \mathbf{w}} \sum_{i=1}^{M} l(\mathbf{w}_i, \mathbf{w}). \tag{44}$$

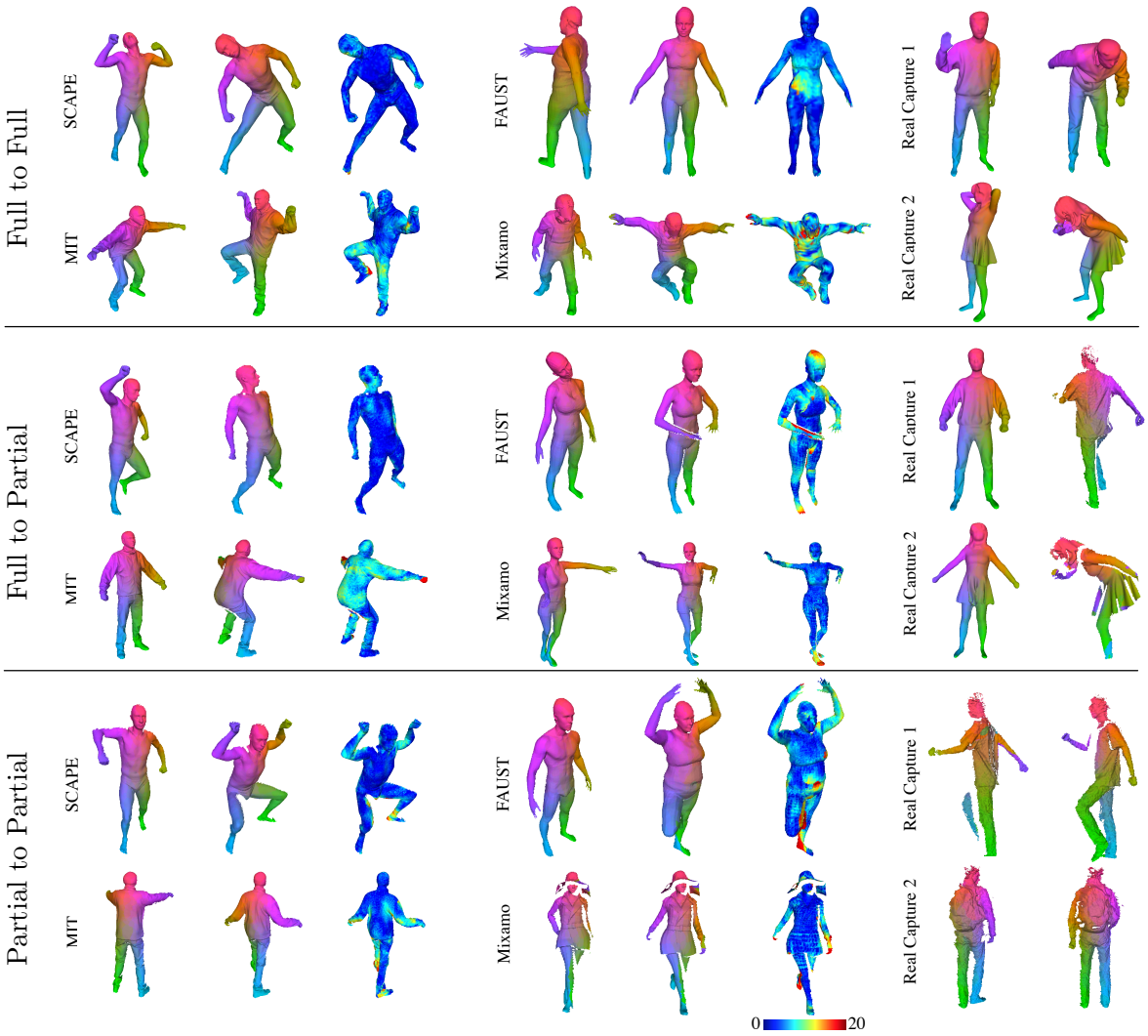
After training, the optimized descriptor extraction tower becomes the output. When $\mathbf{w}_i, \mathbf{w}$ are given by convolutional neural networks, Eq. 44 can be effectively optimized using stochastic gradient descent through back-propagation.

Heterogenous Training Datasets To overcome the challenge of heterogenous training data, two types of classification tasks are included in this ensemble: one for classifying key points, used for iter-subject training where only sparse ground-truth correspondences are available, and one for classifying dense pixel-wise labels, e.g., by segmenting models into patches used for intra-subject training; see Fig. 6. Both contribute to the learning of the descriptor extraction tower.

Descriptor Smoothness Instead of introducing additional terms in the loss function, a simple yet effective strategy is to randomize the dense-label generation procedure. As shown in Figure 6, multiple segmentations of the same person are considered, and a classification problem for each is introduced. Identical points will always be associated with the same label and far-apart points will be associated with different labels. Yet for other points, the number of times that they are associated with the same label is related to the distance between them. Consequently, the similarity of the feature descriptors are correlated to the distance between them on the human body resulting in a smooth embedding satisfying the desired properties discussed earlier.

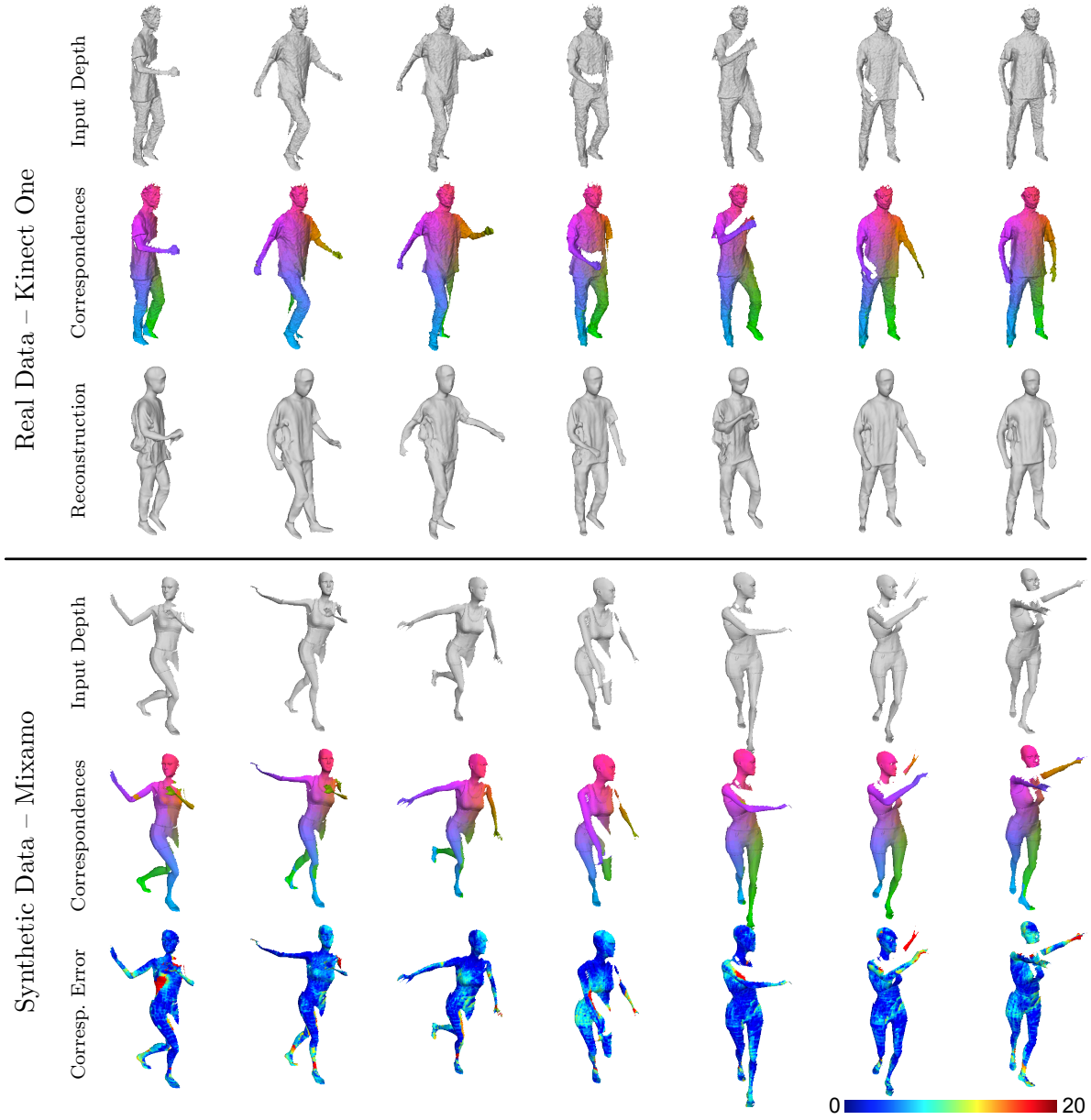
4.3 Correspondence Computation

The classification network can be used to extract per-pixel feature descriptors for depth maps. For full or partial 3D scans, we first render depth maps from multiple viewpoints and compute a per-vertex feature descriptor by averaging the per-pixel descriptors of the depth maps. Then, we use these descriptors to establish correspondences simply by a nearest neighbor search in feature space; see Figure 5. For applications that require deforming one surface to align with the other, we can use these correspondences into any non-rigid deformation technique; see Sec. 2. The performance of deep learning-based correspondences on various real and synthetic datasets, naked and clothed subjects, as well as full and partial matching for challenging examples is illustrated in Figure 4.3. The colorizations validate the accuracy, smoothness, and consistency of dense correspondences, including topological variations between source and target. Correspondences between front and back views are correctly identified for the full-to-partial matchings; see *Real Capture 1*. Popular skeleton tracking methods such as [34, 43] often have difficulties resolving this ambiguity. Note that for poses and clothing that are significantly different than those in the training data, learned correspondences will be erroneous.



4.4 Application: Performance Capture

The learned correspondences can be used for template-based performance capture using a depth map sequence captured from a single RGB-D sensor. The complete geometry and motion is reconstructed in every sequence by deforming a given template model to match the partial scans at each incoming frame of the performance. Unlike existing methods [38, 23, 47, 41, 39, 42] which track a template using the previous frame, in order to avoid potential drifts, the template model is deformed from its canonical rest pose using the computed full-to-partial correspondences. Even though the correspondences are computed independently in every frame, we observe a temporally consistent matching during smooth motions without enforcing temporal coherency as with existing performance capture techniques. Since our deep learning framework does not require source and target shapes to be close, we can effectively handle large and instantaneous motions. For real data, we visualize the reconstructed template model at every frame; for synthetic data we show the error (in cm) to the ground truth.



5 Conclusion

In this course, we introduced 2D/3D registration algorithms and show their applications for data captured with RGB-D devices, such as the Microsoft Kinect or the Intel RealSense. 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. We also presented a deep learning framework that can infer accurate dense correspondences between partial shapes of objects with extremely large intra-class shape variations or deformations. 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, realtime face tracking, and human performance capture.

References

[1] Yobi3d - free 3d model search engine. `https://www.yobi3d.com`. Accessed: 2015-11-03.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. In *ACM TOG (Siggraph)*, pages 408–416, 2005.

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

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

[5] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *IEEE ICCV*, Dec. 2015.

[6] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *IEEE CVPR*, 2014.

[7] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. Primo: coupled prisms for intuitive surface modeling. Computer Graphics Forum (Proc. of SGP), 2006.

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

[9] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. *Computer Graphics Forum (Proc. of SGP)*, 2013.

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

[11] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *IEEE ICCV*, 2015.

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

[13] 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.

[14] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. 1998.

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

[16] 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.

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

[18] 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.

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

[20] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended Intrinsic Maps. In *ACM TOG (Siggraph)*, volume 30, 2011.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. 2012.

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

[23] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. In *ACM TOG (Siggraph Asia)*, 2009.

[24] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. In *ACM TOG (Siggraph)*, pages 72:1–72:12, 2009.

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

[26] 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.

[27] 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.

[28] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3d shapes. *Inter. Journal of Computer Vision*, 2006.

[29] E. Rodola, S. Rota Bulo, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. 2014.

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

[31] S. Rusinkiewicz, B. Brown, and M. Kazhdan. 3d scan matching and registration. In *ICCV 2005 Short Course*, 2005.

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

[33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[34] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE PAMI*, 2012.

[35] O. Sorkine. Least-squares rigid motion using SVD. *Technical Notes*, 2009.

[36] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. Computer Graphics Forum (Proc. of SGP), 2007.

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

[38] J. Süssmuth, M. Winter, and G. Greiner. Reconstructing animated meshes from time-varying point clouds. Number 5, pages 1469–1476, 2008.

[39] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust real-time hand tracking using a single depth camera. *Computer Graphics Forum (Proc. of SGP)*, 2015.

[40] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, T. Sharp, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 2016.

[41] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-P. Seidel. Animation cartography – intrinsic reconstruction of shape and motion. *ACM TOG*, 31(2):12:1–12:15, Apr. 2012.

[42] A. Tkach, M. Pauly, and A. Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 2016.

[43] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169, 2014.

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

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

[46] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. 2008.

[47] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM TOG*, 28(2), 2009.

[48] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li. Dense human body correspondences using convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, 2011.

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

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