Spatiotemporal Modeling for Efficient Registration of Dynamic 3D Faces

Victoria Fernández Abrevaya, Stefanie Wuhrer, Edmond Boyer

Spatiotemporal Modeling for Efficient Registration of Dynamic 3D Faces

Victoria Fernández Abrevaya Stefanie Wuhrer
Edmond Boyer
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP*, LJK, 38000 Grenoble, France
* Institute of Engineering Univ. Grenoble Alpes
{victoria.fernandez-abrevaya, stefanie.wuhrer, edmond.boyer}@inria.fr

Abstract

We consider the registration of temporal sequences of 3D face scans. Face registration plays a central role in face analysis applications, for instance recognition or transfer tasks, among others. We propose an automatic approach that can register large sets of dynamic face scans without the need for landmarks or highly specialized acquisition setups. This allows for extended versatility among registered face shapes and deformations by enabling to leverage multiple datasets, a fundamental property when e.g. *building statistical face models. Our approach is built upon a regression-based static registration method, which is improved by spatiotemporal modeling to exploit redundancies over both space and time. We experimentally demonstrate that accurate registrations can be obtained for varying data robustly and efficiently by applying our method to three standard dynamic face datasets.*

1. Introduction

Facial motion provides significant non-verbal signals that give cues for communication and social interaction. Considering such dynamic face information expands the scope and performance of automatic facial analysis systems, for instance with expression or identitiy recognition [13, 3], pain detection [35] or realistic expression synthesis [33]. To recover coherent face information from inconsistent visual observations, the data must generally be first registered to ensure that anatomically corresponding points are consistently identified. Such registration can be purely spatial, yielding a *static* face pose parametrization that is agnostic to time information. Yet, in the case of faces in motion, the registration can account for the temporal aspects through a spatiotemporal face pose parametrization. The interest is to better capture face deformations with a temporal tracking, where static registrations provide only coarse and noisy motion information. Spatiotemporal, or dynamic, registration is however more complex than its static counterpart since tracking robustly and reliably is still challenging in practice.

Face registration is also confronted with the issue of possibly large and heterogeneous datasets. This is particularly true with recently released large 4D face databases, *e.g.* [11, 36]. While providing a rich and diverse source of information for the study of both spatial and temporal patterns in facial motion, the body of available datasets raises difficulties for global registration, such as dealing with data coming from different capturing systems and different acquisition protocols. Face registration methods in this context should be able to handle datasets in a fully automatic way, as manual intervention is not feasible for thousands of frames, in addition to being robust to different types of noise resulting from different acquisition scenarios.

This work addresses these objectives by proposing an automatic method to register dynamic 3D face data with the longer-term motivation of analyzing large datasets of facial motion. The method is purely geometry-based to allow leveraging any available temporal 3D face scan, even when RGB data is not provided *e.g.* for privacy reasons. We do not require pre-determined landmarks as input, which in the case of 3D data is more challenging to obtain, thus removing a possible source of error. The main innovation is to extend a recent efficient and robust static 3D face registration method [1] to dynamic data by exploiting the spatiotemporal coherence [2] of the data. This enables registrations that both fix identities over temporal sequences and regularize observed motions to prevent high-frequency flickering. The approach presents the following advantages: it can register multiple datasets into a single representation; it does not require color information, as in *e.g.* [11, 10, 14], and is robust to occlusions by construction; it runs an order of magnitude faster than recently proposed methods based on parametric face models [7, 20] while achieving comparable accuracy; and provides compact representations of the results.

We evaluate our method qualitatively and quantitatively on the three publicly available datasets D3DFACS [11], BU-4DFE [31] and BP4D-Spontaneous [36], and show that we can efficiently obtain accurate representations. We provide comparisons to three recent methods and show that we achieve similar or better results in terms of vertex-to-scan error and in the semantic parametrization, while remaining either more general in terms of requirements of the datasets, or more efficient in terms of computational times.

2. Related Work

Numerous works have studied the registration of static 3D face scans [6, 22]. While a static method can be applied independently to each frame of a motion sequence, this is known to lead to artifacts including high-frequency jitter. We consider here 4D face data in the form of sequences of 3D face scans and therefore focus our review on methods that take advantage of the temporal redundancy captured by 4D data. Another line of research that has recently received considerable attention is the reconstruction of 4D facial motion based on monocular 2D video. These works solve an underconstrained reconstruction instead of a 3D registration as addressed in this work; the interested reader is anyway referred to the survey of Zollhöfer *et al.* [38]. In the following we discuss strategies for the registration of 4D face data.

Registration of 4D face data

Sun *et al.* [24] propose a geometry-based coarse-to-fine approach to register planar meshes using conformal mapping, aided by automatically computed feature points. For expression recognition, Fang *et al.* [13] perform pairwise registration of consecutive frames using an Annotated Face Model (AFM) [17]. Temporal information is exploited by initialization using the result of the previous frame.

For real-time expression transfer, Weise *et al.* [30] introduce a system based on a non-rigid Iterative Closest Point (ICP) method, from which a person-specific blendshape model is built and used to track sequences of the same actor. Follow-up work [29] improve on this by using color cues and a probabilistic animation prior to handle noise. The methods of [19, 8] further remove the need for calibration by updating an initial blendshape model on-the-fly. Recent improvements on this line of work include robustness to occlusions and pose [16] and more detailed blendshape models by the use of displacement maps [26].

More recent methods follow two main lines. The first performs registration in texture space by computing correspondences between sparse landmarks predicted using an Active Appearance Model (AAM), which are densified using thin-plate spline deformations [11, 10]. The method of Cosker *et al.* [11] achieves inter-sequence correspondence by registering each frame towards a manually selected neutral expression, and intra-sequence correspondence by registering these neutral frames to a template. To better handle texture variations, Cheng *et al.* [10] extend the previous by using session-and-subject specific AAM, and non-rigid ICP [4] between manually selected neutral frames. Since these methods operate on color information, they require careful acquisition setups with controlled lighting conditions, as *e.g.* moving shadows can lead to inaccuracies.

The second line of work takes advantage of low dimensional parametric shape spaces learned from large databases of static 3D face scans and used as prior during registration, *e.g.* [25, 37, 7, 20]. Most related to our work, multilinear models of identity and expression [7] and a linear articulated model with expressions [20] have been used for this purpose. These works achieve registrations of relatively high accuracy and report running times of $30s$ to $2min$ per frame. Although the overall facial shape is recovered, fine-scale details such as wrinkles are not modeled. Our work shares this property, as well as the robustness and accuracy of these methods while allowing for a gain in efficiency.

Joint registration and reconstruction

In the context of multi-view reconstruction, several works use optical flow to recover a temporally consistent geometry from synchronized multi-view 2D videos, *e.g.* [34, 9, 5]. Bradley *et al.* [9] jointly solve for registration and reconstruction by sequentially tracking the initial frame. Beeler *et al.* [5] improve on this method by using a set of “anchor frames”, which are automatically selected key-frames that define subsequences in which optical flow is performed; results are of very high quality and achieve pore-level details. Both these methods require a dense setup of synchronized video cameras. Valgaerts *et al.* [27] simplify these requirements by introducing a method that achieves results of similar quality from a single pair of stereo cameras, combining scene flow to compute the global registration with shading-based refinement to compute fine-scale details. More recently, Fyffe *et al.* [14] proposed a method to reconstruct a registered model that includes the full head, eye balls, and the interior cavities of mouth and nose. All of these methods achieve temporally coherent results which are of high quality and include fine-scale details. However, they are limited to specific acquisition setups as the input to the methods are synchronized and calibrated 2D videos. In this work, we consider the more general problem of registering the geometry of 4D face scans without the need for reliable color information.

3. Background

This section recalls the two components our method builds upon: a multilinear autoencoder for static registration and a spatiotemporal model to handle time aspects.

3.1. Multilinear autoencoder

Multilinear models are suitable for 3D face modeling and have been shown to effectively separate identity and expression [28]. These models are based on tensor algebra, and obtain decoupled representations by parameterizing a face using a set of weights for each modeled factor of variation, *e.g.* identity and expression, whose relationship is encoded using a *core tensor* $\mathcal{M}$. More formally, multilinear models allow to approximate a registered face $\mathbf{x} \in \mathbb{R}^{3n}$, stored as vector of coordinates associated with n vertices of a mesh, using identity and expression coefficient vectors $\mathbf{w}_{id} \in \mathbb{R}^{d_{id}}$ and $\mathbf{w}_{exp} \in \mathbb{R}^{d_{exp}}$ by

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathcal{M} \times_2 \mathbf{w}_{id} \times_3 \mathbf{w}_{exp}, \tag{1}$$

where $\mathcal{M} \in \mathbb{R}^{3n \times d_{id} \times d_{exp}}$ is the core tensor of the model, $\bar{\mathbf{x}}$ is the mean face over the model's training data, and $\times_i$ denotes mode-i multiplication (we refer to [18] for an overview of tensor operations). Typically, the dimensions d_{id} and d_{exp} are chosen to obtain a compact representation that yields a low approximation error of the training data.

Abrevaya *et al.* [1] recently proposed an autoencoder architecture that allows to robustly and efficiently regress raw 3D face scans on identity and expression using a multilinear model. This method can be used to register an arbitrary 3D facial scan by representing it as a heightmap, which is processed by a convolutional neural network in order to get multilinear model parameters. We leverage this to obtain initial registrations for each frame of an input sequence.

3.2. Bilinear spatiotemporal model

Akhter *et al.* [2] propose the use of a bilinear model to leverage both spatial and temporal redundancies of a single 3D motion sequence. The model requires that the sequence is registered, and we denote its F frames by $\mathbf{x}^1, \dots, \mathbf{x}^F \in \mathbb{R}^{3n}$ in the following. The temporal sequence is organized into a matrix $\mathbf{S} \in \mathbb{R}^{F \times 3n}$ containing each frame in a row, $\mathbf{S} = [\mathbf{x}^1, \dots, \mathbf{x}^F]^T$, which implies that the columns of $\mathbf{S}$ describe vertex trajectories. Let $\mathbf{B} \in \mathbb{R}^{3n \times d_s}$ contain a set of shape basis vectors that encode each frame into a space of dimension d_s, and similarly, let $\mathbf{\Theta} \in \mathbb{R}^{F \times d_t}$ contain a set of temporal basis vectors that encode the trajectory of each vertex into a space of dimension d_t. Then the sequence matrix $\mathbf{S}$ can be decomposed as

$$\mathbf{S} \approx \mathbf{\Theta} \mathbf{C} \mathbf{B}^T, \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{d_t \times d_s}$ is a matrix of spatiotemporal coefficients that compactly encode $\mathbf{S}$. The dimensions d_s and d_t allow to trade off the compactness of the representation and the approximation error of the input sequence.

In the original formulation, both shape basis $\mathbf{B}$ and coefficients $\mathbf{C}$ need to be optimized for each individual sequence. However, they show that the temporal basis $\mathbf{\Theta}$ can be fixed to the Discrete Cosine Transform (DCT), as this approaches the optimal PCA-learned basis when the data is generated from a stationary first-order Markov process. Akhter *et al.* [2] empirically demonstrate this to hold for sparse facial data. We leverage this model to build a compact spatiotemporal space that allows to efficiently register 4D faces in an iterative way.

4. Method

Our goal is to register a large number of sequences of 3D face scans. Each of these sequences may contain many frames, and each frame a large number of vertices, making the problem high-dimensional and difficult to optimize. Furthermore, datasets captured with different acquisition setups present different levels of noise, missing data and occlusions, which plagues naive template fitting. To keep the method as general as possible, we do not assume availability of either landmark or color information.

To tackle this challenging problem, we proceed in two major steps. First, we perform an initialization independently on each frame that robustly and efficiently regresses each scan against identity and expression coefficients of the multilinear model. The resulting approximations correctly capture the general structure of the motion and shape but are still overly smooth and, because each frame is treated independently, exhibit high-frequency jitter. To remedy this, in a second step, we use the multilinear face model to build a bilinear spatiotemporal model, which iteratively improves the initial approximation. This regularizes the motion of the vertices, and turns the problem into a much lower-dimensional one compared to a frame-wise formulation. After a few iterations we obtain registered data which is regularized both in space and time and fits more closely to the scans. Figure 1 summarizes this process.

4.1. Frame-wise initialization

Given a sequence of observations $[\mathbf{o}^1, \dots, \mathbf{o}^F]$ consisting of F frames, we register each of the frames independently by regressing identity and expression coefficients using the multilinear autoencoder, as outlined in Section 3.1. We thus obtain a sequence of identity and expression weights that represent the face motion:

$$W_{id} = [\mathbf{w}_{id}^1, \dots, \mathbf{w}_{id}^F], \text{and} \tag{3}$$

$$W_{exp} = [\mathbf{w}_{exp}^1, \dots, \mathbf{w}_{exp}^F]. \tag{4}$$

Pre-processing Although the neural network we use [1] was trained with raw scans presenting different types of noise, the quality of the registration degrades when the input data differs in form and orientation from the original training data. To ensure results of high quality, we therefore pre-process the face as follows. We first detect the nose tip by training a neural network for this task on depth data with the same architecture and training data as [1]. We next crop the face a radius of 100mm around this point, and perform a coarse frontalization step so that it approximately looks

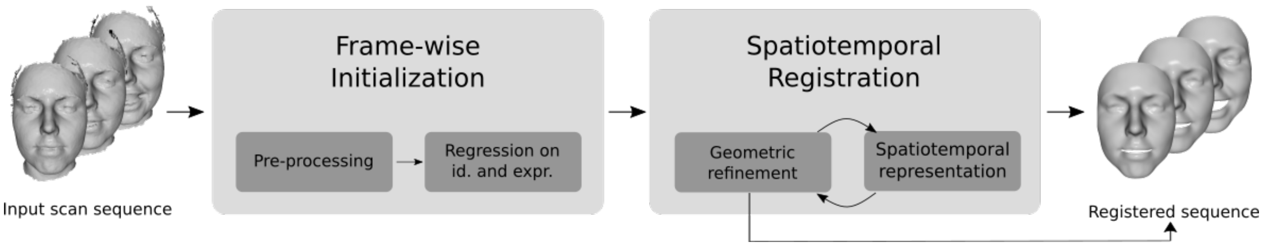

Input scan sequence
Frame-wise Initialization
Pre-processing
Regression on id. and expr.
Spatiotemporal Registration
Geometric refinement
Spatiotemporal representation
Registered sequence

Figure 1: Registration pipeline.

towards the z-direction. For frontalization, we consider the direction of the normals of each vertex, which gives a distribution of orientations over the sphere that sample a semisphere for a cropped face. The directional mean of this distribution gives a coarse approximation of where the face is "looking at", and we align this to the directional mean of the model's mean face. This makes the weak assumption that the face is not upside-down and works well as long as the cropped face does not contain too many holes or extra parts. Note that only a coarse alignment is required here since the autoencoder was trained to have some degree of robustness to pose variation in the depth image. As a result, this preprocessing step can be replaced with any other method that will produce a cropped face and a rough frontalization.

The resulting multilinear model representations $[\mathbf{x}^1, \ldots, \mathbf{x}^F]$, obtained by reconstructing the faces using Equation 1 with the coefficients from Equations 3 and 4, are in the coordinate system in which the multilinear model was learned. For further refinement of these approximations they need to be compared to the original scans. To align the observations $[\mathbf{o}^1, \ldots, \mathbf{o}^F]$ to the model coordinate system, we take advantage of the height-map images generated during regression to find initial correspondences. In particular, we consider the depth image of the cropped and frontalized scan and the depth image of the registered mesh, and establish preliminary correspondences by assigning pixels at the same location. This correspondence is used to rigidly transform $\mathbf{o}^i$ to $\mathbf{x}^i$ $(i = 1, \ldots, F)$. Subsequently, a few iterations of regular iterative closest point alignment are performed. Once each frame is aligned, we discard the cropped version and go back to the original raw scan; this allows to remove the quality of the crop as possible source of error in the subsequent steps.

4.2. Iterative spatiotemporal registration

The previous section computes, for each frame i, spatially aligned observations $\mathbf{o}^i$ along with registered faces $\mathbf{x}^i$ which approximate the geometry of $\mathbf{o}^i$. We leverage spatiotemporal cues to refine $\mathbf{x}^i$ jointly over all frames, thereby obtaining temporally smooth results that can be represented compactly. By iterating between refining the geometry of $\mathbf{x}^i$ to match $\mathbf{o}^i$, and building a spatiotemporal sequence representation, we improve the geometric approximation of the registrations.

Geometric refinement To improve the quality of the approximation $\mathbf{x}^i$ of $\mathbf{o}^i$, we non-rigidly deform the registrations to the scans. The following discussion omits the frame index i to simplify notation. The registration $\mathbf{x}$ is warped to $\mathbf{o}$ by optimizing for displacements $\delta^{\mathbf{x}}$ of the vertices of $\mathbf{x}$ along their normal directions with Laplacian regularization. In particular, we solve for

$$\min_{\{\delta_1^{\mathbf{x}}, \ldots, \delta_n^{\mathbf{x}}\}} \alpha \sum_{j=1}^{n} w_j ||\mathbf{v}_j^{\mathbf{x}} + \delta_j^{\mathbf{x}} \mathbf{n}_j^{\mathbf{x}} - \mathbf{p}_j^{\mathbf{x}}||^2 + \beta \sum_{j=1}^{n} ||\mathcal{L}(\delta_j^{\mathbf{x}})||^2, \quad (5)$$

where $\mathbf{v}_j^{\mathbf{x}} \in \mathbb{R}^3$ is a vertex in frame $\mathbf{x}$, $\mathbf{p}_j^{\mathbf{x}} \in \mathbb{R}^3$ the closest point to $\mathbf{v}_j^{\mathbf{x}}$ in $\mathbf{o}$, $\mathbf{n}_j^{\mathbf{x}} \in \mathbb{R}^3$ the normal vector of $\mathbf{v}_j^{\mathbf{x}}$, $\mathcal{L}$ the cotangent discretization of the Laplace-Beltrami operator [21], and w_j, α, β are scalar weights. We discard closest points whose Euclidean distance is greater than 5mm and whose deviation in normal vector is greater than 45° by setting w_j to zero. This formulation can be efficiently minimized by solving a linear system of equations.

Spatiotemporal sequence projection We build upon the model explained in Section 3.2 to compute a spatiotemporal representation of the input sequence for given $[\mathbf{x}^1, \ldots, \mathbf{x}^F]$. Unlike the original formulation of [2], in which both the shape basis $\mathbf{B}$ and the model coefficients $\mathbf{C}$ need to optimized, we leverage the multilinear model to obtain a person-specific spatiotemporal representation. Specifically, we summarize the regressed identity coefficients W_{id} into a unique coefficient for the entire sequence, given that the identity of the subject is fixed for any given motion. Let $\mathbf{w}_{id}$ denote this unique coefficient vector. We compute $\mathbf{w}_{id}$ as a mean over the regressed results, $\mathbf{w}_{id} = \overline{W_{id}}$. Given $\mathbf{w}_{id}$ we can create a shape basis $\mathbf{B}$ by multiplying the core tensor with $\mathbf{w}_{id}$ as $\mathbf{B}^T = \mathcal{M} \times_2 \mathbf{w}_{id}$. For the temporal basis, we follow Akhter *et al.* and fix $\mathbf{\Theta}$ to the DCT basis.

For each iteration, we gather the approximations $\mathbf{x}^i$ obtained in the previous step into a sequence matrix $\mathbf{S}$, and use

Equation 2 to compute $\mathbf{C}$ by solving

$$\mathbf{B}\mathbf{C}^T = \mathbf{S}^T\mathbf{\Theta}. \tag{6}$$

This can be performed efficiently, as $\mathbf{B}$ is fixed and can be factorized once for all the iterations.

4.3. Sequence representation

To obtain more detailed results we complete the iteration with a final geometric refinement step. This allows the results to leave the bounds of the multilinear model, thereby providing more accurate approximations of $\mathbf{o}^i$. To prevent artifacts, a stronger regularization weight β is used in the last geometric refinement. This allows closer fits towards the data, but in turn looses both the compactness of the representation and the motion regularization. To rectify this, we project the trajectory of the *displacements* $\delta_i^{\mathbf{x}}$ into a second DCT basis (of possibly different dimensionality), obtaining a displacement coefficient vector $\mathbf{d}_i \in \mathbb{R}^{d'_t}$ for each of the n vertices, with $d'_t << F$. We thus retain compactness while allowing for more detailed registrations, as well as preventing flickering in the final trajectory of the vertices.

After this process, we can store the results compactly given the multilinear model, which is stored once for all sequences. For each registered sequence the following information suffices to reconstruct $\mathbf{x}^i$: (1) the identity coefficient $\mathbf{w}_{id} \in \mathbb{R}^{d_{id}}$ that compactly encodes the shape matrix $\mathbf{B}$; (2) the spatiotemporal coefficients $\mathbf{C} \in \mathbb{R}^{d_t \times d_s}$; (3) the dimensions of the temporal basis d_t and d'_t; and (4) the displacement coefficients $\{\mathbf{d}_1, \ldots, \mathbf{d}_n\}, \mathbf{d}_i \in \mathbb{R}^{d'_t}$. This significantly reduces the storage requirements of large datasets (*e.g.* from 9.1GB to less than 1MB in the example from Figure 4), while still retaining a reasonable level of detail.

5. Evaluation

We validate our method by registering D3DFACS [11], BU-4DFE [31] and BP4D-Spontaneous [36] datasets. D3DFACS contains 519 sequences of 10 subjects performing different types of facial action units, while BU-4DFE presents 101 subjects with 6 sequences each performing the six prototypical emotions; in both cases the average sequence length is around 100 frames and the meshes contain around 30K and 35K vertices respectively. BP4D-Spontaneous is made of 328 sequences with 41 subjects performing 8 tasks each, which were designed to elicit spontaneous emotions. The average sequence length is around 1100 frames and the average number of vertices is around 37K. In the following we provide both qualitative and quantitative evaluations over these.

Implementation details The code was implemented in C++ using the Eigen3 library [15]. We use the publicly available autoencoder of [1] that was trained on BU-3DFE [32] and Bosphorus [23] datasets. The dimensions of identity and expression spaces are set to 65 and 20. For the dimension of the temporal basis, we observe that the impact depends on the length of the sequence. We thus set it to a factor of the sequence length; in particular $F/5$. We set $d'_t = 5$, and unless otherwise specified, we fix the number of iterations to 5. For Equation 5 we set $\beta = 1$, $\alpha = 0.9$ during iterations and $\alpha = 0.8$ for the final step. Since BU-4DFE presents noisier scans, we set $\alpha = 0.5$ during iterations and $\alpha = 0.2$ for the final step to avoid overfitting. The mesh used to register all face scans contains 5996 vertices.

5.1. Qualitative results

Figure 3 shows an example of the results obtained on each of the steps of the method: regression, spatiotemporal registration, and final refinement. Figure 2 shows a few more examples of registrations obtained on D3DFACS, BU-4DFE and BP4D-Spontaneous. They illustrate that accurate cross-dataset registrations can be obtained, while still being robust to different types of noise in the data.

5.2. Quantitative results

We evaluate the quality of the registrations with two commonly used metrics: median *per-vertex error* towards the input scan and *landmark distances*. The median per-vertex error is taken across all registered frames in the dataset, and shows how close the registrations are to the real scans. We also evaluate semantic accuracy by manually placing landmarks on 5 key-frames over 10 randomly selected sequences of D3DFACS, and measuring the Euclidean distances between these and the landmarks defined over the template. We use in particular 11 landmarks: nose tip, inner and outer eye corners, left and right mouth corners, and upper/lower outer and inner center of lips. The chosen key-frames sample the sequence by taking the first and last frame, the peak frame, and two intermediate ones.

We also evaluate the stability of the motion by using a *compactness* measure [12] as follows. For each sequence, we align the frames using generalized Procrustes analysis, perform PCA, and measure the amount of variabilty captured by each principal component. If the registrations exhibit high-frequency jitter, we expect to see less variability retained by the first principal components, as the variations coming from flickering vertices would have to be encoded by higher-order principal components. To be able to summarize this over the entire dataset, we determine the mean variability obtained as a function of the percentage of principal components considered.

Figure 5 shows a cumulative plot of the median per-vertex error on D3DFACS obtained after the initial regression, after 1, 3, 5 and 10 iterations of spatiotemporal registration, and for the final result. We can see that the iterative process improves the initial regressions in terms of surface

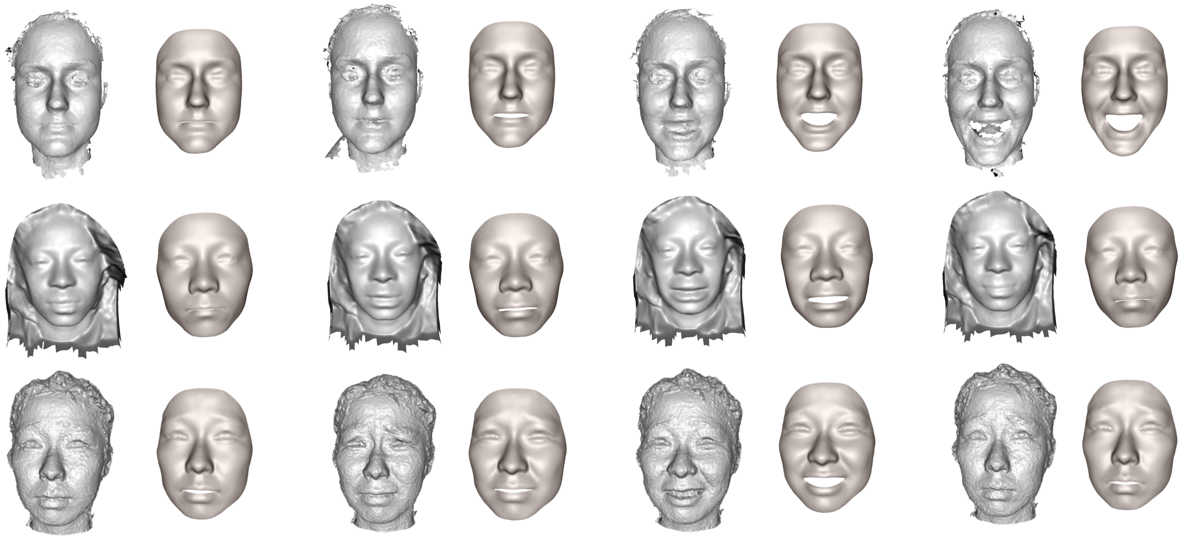


Figure 2: Registration examples on (from top to bottom): D3DFACS, BU-4DFE and BP4D datasets.

fit. Furthermore, Table 1 shows the mean landmark error over the 11 landmarks for the final results obtained after 1, 3, 5 and 10 iterations. Despite being a landmark-free registration method, the method obtains a good semantic accuracy that is improved with each iteration.

We evaluate the benefits of the temporal regularization by comparing the full model with a static version of our framework. For this, instead of projecting onto the spatiotemporal model we independently project each frame onto the shape basis $\mathbf{B}$, and measure the results in terms of vertex error and compactness. Figure 6 shows cumulative plots obtained for these registrations. Note that while using a spatiotemporal model achieves similar accuracies in terms of vertex error, the compactness of each sequence improves with the spatiotemporal model, implying less high-frequency jitter with the latter. This results can also be qualitatively assessed in the accompanying video.

Finally, we show the ability of the method to track long videos by registering the sequences from BP4D-Sponteanous, many of which consist of more than 1000 frames. We obtain a mean vertex error of 0.33mm over all registered sequences and frames. Figure 4 further shows the median per-vertex error for each frame on one example.

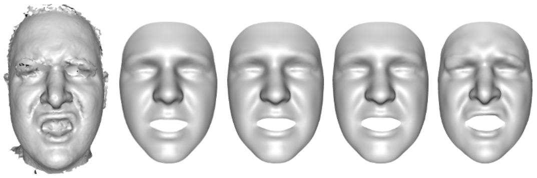


Figure 3: Results for the successive approach steps (left to right): raw scan, regression, 1 iteration, 5 iterations, final.

	It. 1	It. 3	It. 5	It. 10	Li *et al.* [20]
Mean error	2.92	2.77	2.69	**2.65**	3.13

Table 1: Mean landmark error in mm., for 1, 3, 5 and 10 iterations, and comparison with Li *et al.* [20], over 10 selected sequences of the D3DFACS dataset.

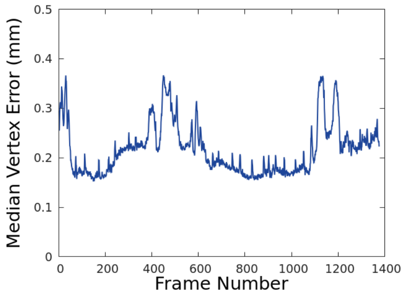


Figure 4: Median per-vertex error for each frame of a long sequence in BP4D.

This error stays between 0.1 and 0.4mm and does not increase with time, suggesting that no drift is occuring. This is expected, as the regression step is performed independently on each frame. A textured visualization of this sequence can be found in the accompanying video.

With respect to the running times, we report a mean per-frame processing time of 578ms on the D3DFACS dataset, 637ms on BU-4DFE and 399ms for BP4D, for 5 iterations in all cases. Computation times were measured on an Intel Xeon 3.30GHz with NVidia GeForce GTX 1080 GPU.

5.3. Comparisons

We compare our method to the previous works of Bolkart and Wuhrer [7] , Li *et al*. [20] and Cosker *et al*. [11] using registrations provided by the authors.

Bolkart and Wuhrer This work [7] also registers motion sequences in a fully-automatic manner by using a multilinear model and geometric information only. We compare to this method on 497 sequences from BU-4DFE, which are the sequences that were correctly registered by [7]. Figure 7a shows cumulative plots of the median per-vertex error on all the registered sequences in BU-4DFE, comparing [7] to our registration without and with the final refinement step, since [7] has no refinement step. Figure 8a further shows a qualitative comparison over a challenging example. Results reveal similar accuracy for both methods without the refinement step, whereas [7] requires around 30 seconds per-frame to process.

Li et al. The method of [20] was used to register D3DFACS, and thus we compare our results over this dataset. For a fair comparison we crop their full-head model so that it contains only the face, to be similar to our registrations. We obtain a mean vertex error of 0.13 mm for our method, and 0.33mm for [20]. In Figure 7b we show the cumulative plots for the median per-vertex distance for both methods. They demonstrate that our approach achieves higher accuracy while reducing both the running time ([20] reported 155 seconds per-frame) and the requirements on the dataset. We also compare with respect to landmark errors; results can be found in Table 1. Our approach achieves better results with a single iteration, even though it requires no pre-determined landmarks to guide the process. This also confirms that our registrations faithfully preserve the anatomic semantics. Figure 8b shows a qualitative comparison.

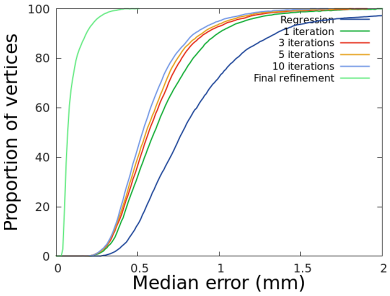
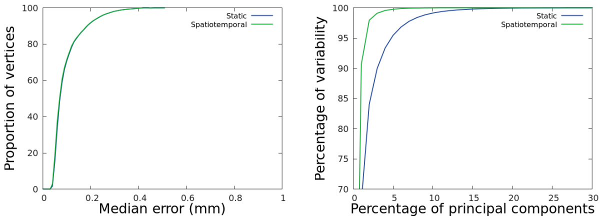


Figure 5: Cumulative plot of median per-vertex error over D3DFACS (46028 frames) for: regression results, spatiotemporal registration (1, 3, 5 and 10 iterations) and final refinement.



(a) Median per-vertex error (b) Mean compactness

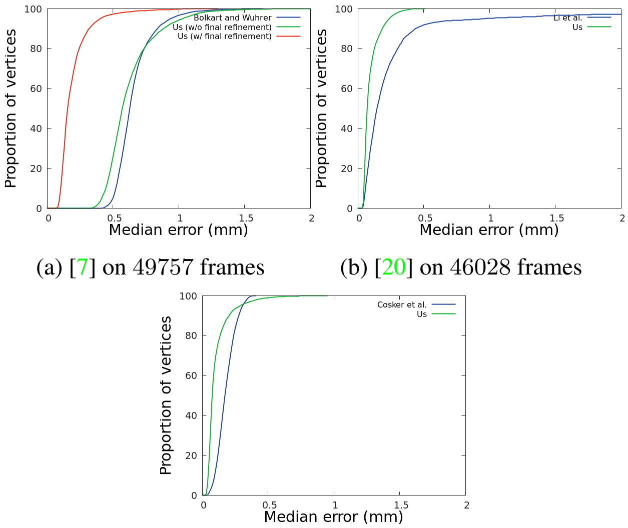
Figure 6: Comparison between our method and a static version, in terms of vertex error and sequence compactness.

Cosker et al. Finally, we compare to the work of Cosker *et al*. [11]. Comparisons are done over 3 sequences of D3DFACS that were provided by the authors. We obtain a mean error of 0.13mm for our method and 0.18 mm for [11]. Figure 7c shows the results in terms of median per-vertex error. They demonstrate similar accuracy although our method is more general as it does not require a controlled capture setup. Figure 8c shows a qualitative comparison.

Comparisons on efficiency The efficiency of our method comes from both the regression step and the spatiotemporal model optimization. The regression is essential to get a good starting point that is already close to a local minimum, and it can be performed efficiently on the GPU thanks to the heightmap representation. Furthermore, thanks to the spatiotemporal model we need to optimize for much less parameters, while still remaining in a global sequence formulation. In particular, on each iteration we need to solve for the matrix $\mathbf{C}$ which is of size $d_s \times d_t$, with $d_s = d_{exp}$. In our implementation $d_t = F/5$ and thus we optimize for $d_{exp}F/5$ parameters, reducing by a factor of 5 compared to a frame-by-frame formulation. The method of [7] optimizes for the parameters of each frame, which amounts to $d_{id} + Fd_{exp}$ variables to be solved. The main data term on the method of [20] optimizes for shape and expression parameters plus per-joint pose parameters of an articulated model on a frame-by-frame basis, increasing the complexity. Moreover, each frame is initialized from the previous one and thus it cannot be parallelized. As for UV-based methods such as [11], the computational complexity depends on the number of pixels of the image; while these are usually more efficient than 3D-based ones, we have shown that we can achieve similar accuracy, while remaining more general with respect to the acquisition setup.

5.4. Limitations

The use of regression for initialization allows the method to be robust to noise in the input data, but it comes with drawbacks. In particular, the use of a heightmap implies

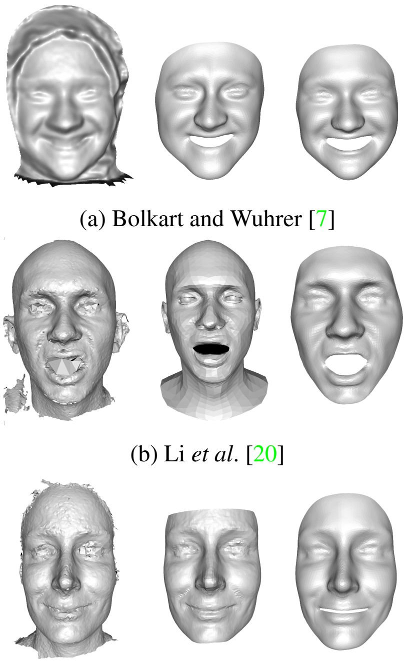


(a) [7] on 49757 frames

(b) [20] on 46028 frames

(c) [11] on 292 frames

Figure 7: Comparisons to Bolkart and Wuhrer [7] on BU-4DFE, Li *et al.* [20] on D3DFACS, and Cosker *et al.* [11] on a subset of D3DFACS. Cumulative plots for median per-vertex error.



(a) Bolkart and Wuhrer [7]

(b) Li *et al.* [20]

(c) Cosker *et al.* [11]

Figure 8: Qualitative comparisons. From left to right: original scan, compared method, our result.

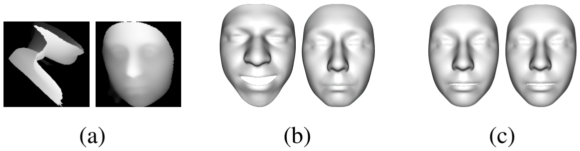


Figure 9: Failure example. (a) Heightmap obtained after bad nose tip detection (left) and the following frame (right); (b) Regression results; (c) Recovery by interpolation.

that the method is not rotation-invariant, and thus a proper pre-processing is needed to ensure that the face is “looking front”. Although this does not require accurate pose detection (the network was trained with data showing ±30° of pose variation), the output will be more accurate the closer the input is to a frontal pose, and this in turn affects the final result. Furthermore, our choice of initialization can sometimes be a source of failure, particularly with the nose tip detection; see *e.g.* Figure 9. When this step fails all subsequent steps fail too, as regressions are inaccurate and initial correspondences cannot be found. In our experiments, this resulted in failure of some of the frames in BU-4DFE and BP4D datasets. On the other hand, dealing with motion data means that isolated failed frames can be ignored without resulting in failure of the entire sequence. In our implementation we automatically detect failed frames after ICP diverges, and this is fixed by interpolating pose and shape parameters using correct neighbouring frames. With this simple approach all sequences from BU-4DFE and 95% from BP4D were correctly registered (no errors were found during registration of D3DFACS).

Another intrinsic limitation comes from the limited scope of our trained model. Particularly for expressions that are far from this scope, our framework will provide only a coarse approximation and even the final refinement step can fail to compensate. Finally, the approach does not yet perform explicit tracking of subtle movements such as eyelids or the mouth during speech, which we leave for future work.

6. Conclusions

We presented a method that automatically registers large sets of dynamic 3D face scans. The main novelty is to combine a robust and efficient static registration approach with a spatiotemporal model, allowing to exploit redundancies over space and time and enabling compact representations for temporal face sequences. Remarkably, we successfully registered over three standard datasets using our method, without loosing accuracy and with significantly better time performances than related works, demonstrating therefore the efficiency of our model. In the future, we plan to use this method to build versatile statistical face models that cover large ranges of identities and expressions.

Acknowledgements We would like to thank the authors of [20], [7] and [11] for providing comparison data.

References

[1] V. F. Abrevaya, S. Wuhrer, and E. Boyer. Multilinear autoencoder for 3d face model learning. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. 1, 3, 5

[2] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics*, 31(2):17, 2012. 1, 3, 4

[3] T. Alashkar, B. B. Amor, M. Daoudi, and S. Berretti. A grassmann framework for 4d facial shape analysis. *Pattern Recognition*, 57:21–30, 2016. 1

[4] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[5] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics*, 30(4):75:1–75:10, 2011. 2

[6] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 2

[7] T. Bolkart and S. Wuhrer. 3d faces in motion: Fully automatic registration and statistical analysis. *Computer Vision and Image Understanding*, 131:100–115, 2015. 1, 2, 7, 8

[8] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics*, 32(4):40:1–9, 2013. 2

[9] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Transactions on Graphics*, 29(4):41:1–41:10, 2010. 2

[10] S. Cheng, I. Kotsia, M. Pantic, and S. Zafeiriou. 4dfab: A large scale 4d facial expression database for biometric applications. *arXiv preprint arXiv:1712.01443*, 2017. 1, 2

[11] D. Cosker, E. Krumhuber, and A. Hilton. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 1, 2, 5, 7, 8

[12] R. Davies, C. Twining, and C. Taylor. *Statistical models of shape: Optimisation and evaluation*. Springer Science & Business Media, 2008. 5

[13] T. Fang, X. Zhao, O. Ocegueda, S. K. Shah, and I. A. Kakadiaris. 3d/4d facial expression analysis: An advanced annotated face model approach. *Image and Vision Computing*, 30(10):738–749, 2012. 1, 2

[14] G. Fyffe, K. Nagano, L. Huynh, J. Busch, A. Jones, and H. L. H. L. Debevec. Multi-view stereo on consistent face topology. In *Eurographics*, 2017. 1, 2

[15] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 5

[16] P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[17] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):640–649, 2007. 2

[18] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. 3

[19] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics*, 32(4):42:1–10, 2013. 2

[20] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36(6):194, 2017. 1, 2, 6, 7, 8

[21] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 4

[22] A. Patel and W. A. Smith. 3d morphable face models revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[23] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. In *European Workshop on Biometrics and Identity Management*. Springer, 2008. 5

[24] Y. Sun, X. Chen, M. Rosato, and L. Yin. Tracking vertex flow and model adaptation for three-dimensional spatiotemporal face analysis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(3):461–474, 2010. 2

[25] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183–1, 2015. 2

[26] D. Thomas and R.-I. Taniguchi. Augmented blendshapes for real-time simultaneous 3d head modeling and facial motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[27] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics*, 31(6):187:1–11, 2012. 2

[28] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM transactions on Graphics*, 24(3):426–433, 2005. 2

[29] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (SIGGRAPH)*, 30(4):77:1–77:10, 2011. 2

[30] T. Weise, H. Li, L. V. Gool, and M. Pauly. Face/off: Live facial puppetry. In *Symposium on Computer Animation*, 2009. 2

[31] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2008. 1, 5

[32] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *International Conference on Automatic Face and Gesture Recognition (FG)*, 2006. 5

[33] H. Yu, O. G. Garrod, and P. G. Schyns. Perception-driven facial expression synthesis. *Computers & Graphics*, 36(3):152–162, 2012. 1

[34] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. In *ACM Transactions on Graphics (TOG)*, pages 548–558. ACM, 2004. 2

[35] X. Zhang, L. Yin, and J. F. Cohn. Three dimensional binary edge feature representation for pain expression analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2015. 1

[36] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014. 1, 5

[37] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth. Automatic reconstruction of personalized avatars from 3d face scans. *Computer Animation and Virtual Worlds*, 22(2-3):195–202, 2011. 2

[38] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)*, 2018. 2