A Robust Multilinear Model Learning Framework for 3D Faces

Timo Bolkart, Stefanie Wuhrer

A Robust Multilinear Model Learning Framework for 3D Faces*

Timo Bolkart
Saarland University, Germany
tbolkart@mmci.uni-saarland.de

Stefanie Wuhrer
Inria Grenoble Rhône-Alpes, France
stefanie.wuhrer@inria.fr

Abstract

Multilinear models are widely used to represent the statistical variations of 3D human faces as they decouple shape changes due to identity and expression. Existing methods to learn a multilinear face model degrade if not every person is captured in every expression, if face scans are noisy or partially occluded, if expressions are erroneously labeled, or if the vertex correspondence is inaccurate. These limitations impose requirements on the training data that disqualify large amounts of available 3D face data from being usable to learn a multilinear model. To overcome this, we introduce the first framework to robustly learn a multilinear model from 3D face databases with missing data, corrupt data, wrong semantic correspondence, and inaccurate vertex correspondence. To achieve this robustness to erroneous training data, our framework jointly learns a multilinear model and fixes the data. We evaluate our framework on two publicly available 3D face databases, and show that our framework achieves a data completion accuracy that is comparable to state-of-the-art tensor completion methods. Our method reconstructs corrupt data more accurately than state-of-the-art methods, and improves the quality of the learned model significantly for erroneously labeled expressions.

1. Introduction

The human face plays an essential role in all kinds of social interactions as it provides a rich source of nonverbal communication. Within the last 20 years there has been a great increase in investigating the facial shape with its richness of variations within multiple fields ranging from computer vision and graphics to psychology, medicine and ergonomics. Depending on the application this requires a model that precisely describes the facial variations and achieves a high level of realism.

The facial shape is highly variable as it is affected by *e.g.* ethnicity, sex, age or facial expression. This makes it difficult to model the human face by hand; instead data driven methods are applied to learn a model. For 3D facial shape, data driven methods are supported by the increasing number of publicly available 3D face databases that were acquired within the last decade (*e.g.* [35, 31]).

Multilinear models are widely used to represent the statistical variations of 3D faces as they successfully decouple shape changes due to identity and expression (*e.g.* [33, 14, 34]). We focus on these two types of shape changes here, but multilinear models allow other factors to be included.

To compute statistics of a class of shapes requires all shapes to be in correspondence [15, Chapter 1]. Multilinear models further require the full Cartesian product of all facial attributes (*i.e.* all identities need to be present in all expressions), and for the data to be in semantic correspondence specified by labels for the different expressions. While multilinear face models have been shown to be a powerful tool, acquiring a 3D face database that suits the needs of a multilinear model is difficult.

There is a wealth of static 3D face data that has been captured, where problems in a small percentage of the data prevent learning an effective multilinear model. For instance, the Bosphorus database [31] would allow to learn information on action units, but not every person was captured in every action unit and some scans are corrupt. New tools are needed to leverage such data.

In particular, tools are needed to cope with missing data, corrupt data, or wrong semantic correspondences. Missing data occur if not all available identities are present in all expressions, *i.e.* some identities are only captured in a subset of the expressions. Missing data are caused if some subjects are unable to perform certain expressions spontaneously, or if an existing database should be extended by additional expressions with some subjects being unavailable for further scanning. Corrupt data arise if the facial geometry is noisy or partially occluded. If the data are corrupt, frequently used registration methods (*e.g.* [29, 27, 17]) fail, and establishing a full vertex correspondence without prior knowledge becomes infeasible. Wrong semantic correspondences arise if a subject has difficulties in performing specific expres-

*This work has been partially funded by the German Research Foundation (WU 786/1-1, Cluster of Excellence MMCI, Saarbrücken Graduate School of Computer Science).

sions correctly and mixes up certain expressions, or due to erroneous classifications of the performed expressions.

Overall, building a multilinear model is a chicken-and-egg problem. Given a multilinear face model, it is able to complete missing data (*e.g*. [11]), reconstruct corrupt data (*e.g*. [6]), or label expressions (*e.g*. [27]), all of which is necessary to build up a database that fulfills the needs of a multilinear model. This motivates us to formulate the multilinear model learning as a groupwise optimization framework that aims to learn a multilinear face model while at the same time correcting the data.

In this work we introduce the first groupwise robust multilinear model (RMM) learning framework that is robust to missing data, corrupt data caused by noise and partial occlusions, wrong semantic correspondence, and inaccurate vertex correspondence caused by drift within the surface. The novelties of our framework are

- a data completion technique with similar performance as state-of-the-art tensor completion methods
- a data reconstruction technique of corrupt data that outperforms the state-of-the-art, and
- a re-labeling technique to improve semantic correspondence.

2. Related work

Multilinear face models: Multilinear face models have been used in a variety of applications. Vlasic *et al*. [33] and Dale *et al*. [14] use a multilinear face model to reconstruct 3D faces from 2D video and to transfer expressions between 2D videos. Mpiperis *et al*. [27] use the model for identity and expression recognition of 3D face scans. Yang et al. [34] and Bolkart and Wuhrer [3] exploit the decoupling of identity and expression variations to obtain a compact representation for facial motion sequences. Cao *et al*. [11] generate user specific blendshapes that are used to track the facial performance in 2D videos. Brunton *et al*. [6] use multiple localized multilinear models to reconstruct 3D faces from noisy and partially occluded face scans.

To learn a multilinear face model, all these methods require a fully registered face database where each identity is present in each expression, and the expressions are correctly labeled. To complete missing data, Vlasic *et al*. [33] fill in missing data in a preprocessing step. None of these methods aim to learn a multilinear face model while at the same time correcting and completing the data.

Completing missing data: To estimate missing data, matrix factorization and low rank methods have been proposed. Tipping and Bishop [32] introduce a probabilistic principal component analysis (PCA) that jointly learns a PCA model and completes missing data. Candes *et al*. [10] use a convex rank approximation to complete matrices with missing data. With further sparsity constraints, this convex matrix rank approximation forms a robust PCA approach [9] that allows to learn a PCA model from missing and noisy data. Liu *et al*. [26] extend the matrix rank approximation to tensors and propose the algorithm HaLRTC that is a state-of-the-art algorithm to complete missing data in tensors of 2D images. Chen *et al*. [13] use a similar approach that imposes rank constraints on the factor matrices of the tensor to complete 2D image tensors. Zhang *et al*. [36] complete 2D image tensors in the Fourier domain; hence this algorithm is not directly applicable to 3D data.

In contrast to these methods, RMM is more general as it further handles erroneous vertex correspondence, corrupt data, and wrong semantic correspondences in one common framework.

Once a good face model is given, *e.g*. a multilinear model learned from training data, it can synthesize new faces to complete missing data. For instance, Cao *et al*. [11] complete expressions by synthesizing user specific blendshapes. This method requires an initially registered database with each identity present in each expression.

Cleaning corrupt data: Parametric models such as blendshape models [23], morphable models [2] or multilinear models have been shown to be robust to noise and partial occlusions, and hence can be used to clean up corrupt data. Li *et al*. [24] use a blendshape model to reconstruct facial performance from noisy RGBD data. Hsieh *et al*. [19] use personalized blendshapes for RGBD tracking that is robust to various partial occlusions and noise. Blanz *et al*. [1] use a PCA model to reconstruct faces from potentially noisy face scans. Brunton *et al*. [7] give a comprehensive overview of statistical face models and shows their ability to reconstruct noisy and partially occluded data. Further, Brunton *et al*. [6] show that global and localized multilinear face models are able to reconstruct noisy and partially occluded data.

All these parametric models are robust to corrupt data due to their prior knowledge of the possible variations. But all these models require data for training or manually designed basis deformations. In contrast to RMM, none of these methods aim to learn a statistical model while at the same time correcting corrupt data.

Semantic correspondence optimization: Expression recognition methods can be applied to classify expressions and then exchange the labels accordingly. For a comprehensive overview of expression recognition methods, we refer to the survey by Sandbach *et al*. [30]. Note that once a multilinear face model is learned, it can be used to classify facial expressions [27]. Expression recognition methods require prior knowledge of the expression classes to classify expressions into a correctly labeled expression set. To run standard expression recognition methods for semantic correspondence optimization would therefore require a manual selection of a correctly labeled subset. RMM in contrast fully automatically adjusts the expression labeling with a

groupwise optimization.

Groupwise optimization: RMM is related to methods that jointly learn a model and compute correspondence in a groupwise fashion. Kotcheff and Taylor [21] and Davies *et al*. [15] jointly learn a linear model and optimize vertex correspondence between 3D surfaces. Burghard *et al*. [8] use a part-based linear model for the correspondence optimization. Chen *et al*. [12] use a kernel PCA to model the data and the correspondences therein nonlinearly. Hirshberg *et al*. [18] jointly learn a model and optimize correspondence for human bodies with a skeleton-based approach. Bolkart and Wuhrer [4] use a multilinear correspondence optimization method that results in better vertex correspondences and is computationally more efficient than linear methods. Inspired by this, we demonstrate that a groupwise multilinear model optimization framework can be employed to complete missing data, clean up corrupt data, and correct wrong semantic correspondences.

3. Multilinear face model

This section introduces the multilinear model applied to a database of registered and rigidly aligned 3D faces of d_2 identities performing d_3 expressions each. Let $\mathbf{x} = (x_1, y_1, z_1, \cdots, x_n, y_n, z_n)^T$ denote one face with n vertices (x_1, y_1, z_1), and let $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ denote a three dimensional array, also called 3-mode tensor. Here, each mode describes one axis of the tensor. We center each $\mathbf{x}_i$ by subtracting the mean $\bar{\mathbf{x}} = \frac{1}{d_2 d_3} \sum_{i=1}^{d_2 d_3} \mathbf{x}_i$ over all shapes and arrange the centered data in $\mathcal{X}$ such that the coordinates of each $\mathbf{x}_i$ align with the first mode. Based on the semantic correspondence, the different identities are associated with the second mode of $\mathcal{X}$, and the different expressions with the third mode. A higher order singular value decomposition (HOSVD) [22] decomposes $\mathcal{X}$ into a multilinear model tensor $\mathcal{M} \in \mathbb{R}^{3n \times m_2 \times m_3}$ and orthogonal factor matrices $\mathbf{U}_2 \in \mathbb{R}^{d_2 \times m_2}$ and $\mathbf{U}_3 \in \mathbb{R}^{d_3 \times m_3}$ as

$$\mathcal{X} = \mathcal{M} \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{1}$$

where $\mathcal{M} \times_n \mathbf{U}_n$ denotes the n-th mode product of tensor $\mathcal{M}$ and a matrix $\mathbf{U}_n$ that replaces each vector $\mathbf{v} \in \mathbb{R}^{m_n}$ in $\mathcal{M}$ aligned with the n-th mode by $\mathbf{U}_n \mathbf{v}$. To compute $\mathbf{U}_n$, HOSVD unfolds $\mathcal{X}$ along the n-th mode to a matrix $\mathbf{X}_{(n)}$ (the vectors of $\mathcal{X}$ aligned with the n-th mode form the columns of $\mathbf{X}_{(n)}$) and matrix SVD is performed as $\mathbf{X}_{(n)} = \mathbf{U}_n \mathbf{S}_n \mathbf{V}_n^T$, where $\mathbf{U}_n \in \mathbb{R}^{d_n \times d_n}$. The multilinear model is then computed as $\mathcal{M} = \mathcal{X} \times_2 \mathbf{U}_2^T \times_3 \mathbf{U}_3^T$. Truncating columns of $\mathbf{U}_n$ reduces the dimensionality of $\mathcal{M}$, where $m_n \leq d_n$ defines the number of remaining columns of $\mathbf{U}_n$.

The multilinear model allows to reconstruct a registered 3D face $\mathbf{f} \in \mathbb{R}^{3n}$ given coefficients for identity $\mathbf{w}_2 \in \mathbb{R}^{m_2}$ and expression $\mathbf{w}_3 \in \mathbb{R}^{m_3}$ as

$$\mathbf{f} = \bar{\mathbf{x}} + \mathcal{M} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T. \tag{2}$$

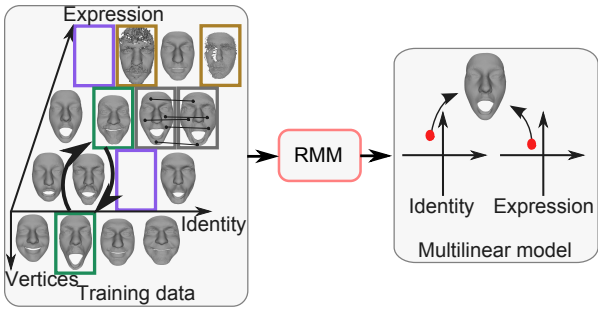


Figure 1. Overview of our robust multilinear model (RMM) learning framework that is robust to missing data (purple), corrupt data (brown), wrong semantic correspondence (green), and inaccurate vertex correspondence (gray).

4. Robust multilinear model learning

This section describes our RMM framework as outlined in Figure 1 that is robust to missing data, corrupt data, wrong semantic correspondence and erroneous vertex correspondence. To achieve this robustness to erroneous training data, RMM jointly learns a multilinear model and corrects the data. First, we describe the groupwise multilinear objective function that minimizes multilinear compactness. Second, we describe how to optimize the objective function to complete and clean up an incomplete database and improve wrong semantic correspondence that allows to build a multilinear model using Eq. 1.

4.1. Multilinear objective function

Our objective function consists of a compactness energy E_C, a data energy E_D, and a regularization energy E_R^μ as

$$E(\mathcal{X}, w_D, w_R, \mu) = E_C + w_D E_D + w_R E_R^\mu, \tag{3}$$

where the weights w_D and w_R control the influence of the data and regularization terms, respectively. The parameter μ specifies the influence of the regularization target. We now describe all terms in more detail.

Compactness: The recently introduced multilinear compactness term [4] aims to minimize the mode-ranks of $\mathcal{X}$ by minimizing the ranks of $\mathbf{X}_{(2)}$ and $\mathbf{X}_{(3)}$. Minimizing E_C implicitly favors compact multilinear models as

$$E_C = \frac{1}{d_2} \ln(det(\mathbf{D}_2 + \delta_2 \mathbf{I}_{d_2})) + \frac{1}{d_3} \ln(det(\mathbf{D}_3 + \delta_3 \mathbf{I}_{d_3})), \tag{4}$$

where $\mathbf{D}_2 = \frac{1}{d_3} \mathbf{X}_{(2)} \mathbf{X}_{(2)}^T$ and $\mathbf{D}_3 = \frac{1}{d_2} \mathbf{X}_{(3)} \mathbf{X}_{(3)}^T$ are the mode-2 and mode-3 covariance matrices, and $\mathbf{I}_{d_i} \in \mathbb{R}^{d_i \times d_i}$ is the identity matrix. The small regularization constant δ_n avoids singularities of E_C for mode covariance matrices without full rank.

Data: The data term measures the distance of a corrupt shape $\mathbf{x}$ in $\mathcal{X}$ (aligned with the first mode of $\mathcal{X}$) to a corresponding unregistered face scan $\mathbf{s}$. The data energy is

$$E_D = \frac{1}{n} \sum_{k=1}^{n} \min(\|\mathbf{v}_k(\mathbf{x}) - \mathbf{n}\mathbf{n}_k\|^2, \rho), \tag{5}$$

where $\mathbf{nn}_k$ denotes the nearest neighbor of $\mathbf{v}_k(\mathbf{x})$ in $\mathbf{s}$ computed by a point-to-plane distance measure, and ρ is a truncation threshold to be robust to outliers.

Regularization: The regularization term for each shape $\mathbf{x}$ in $\mathcal{X}$ is a bi-Laplacian of the form

$$E_R^\mu = \frac{1}{n}\sum_{k=1}^{n} \left\| U^2(\mathbf{v}_k(\mathbf{x})) - \mu U^2(\mathbf{v}_k(\widetilde{\mathbf{x}})) \right\|^2, \tag{6}$$

where $\mathbf{v}_k(\mathbf{x})$ and $\mathbf{v}_k(\widetilde{\mathbf{x}})$ denote the k-th vertex of shape $\mathbf{x}$ and the fixed reference shape $\widetilde{\mathbf{x}}$, respectively. The energy E_R^μ measures the deformation energy of $\mathbf{x}$ relative to $\widetilde{\mathbf{x}}$. The parameter $\mu \in [0,1]$ controls the regularization influence of $\widetilde{\mathbf{x}}$. Minimizing E_R^μ forces $\mathbf{x}$ to be locally smooth, and the local geometry of $\mathbf{x}$ to be similar to $\widetilde{\mathbf{x}}$. The operator $U^2(\mathbf{p})$ approximates the discrete bi-Laplacian [20] as

$$U^2(\mathbf{p}) = \frac{1}{|N(\mathbf{p})|} \sum_{\mathbf{p}_r \in N(\mathbf{p})} U(\mathbf{p}_r) - U(\mathbf{p}), \tag{7}$$

where $N(\mathbf{p})$ denotes the set of neighbors of vertex $\mathbf{p}$ within the mesh, and $U(\mathbf{p}) = \frac{1}{|N(\mathbf{p})|}\sum_{\mathbf{p}_r \in N(\mathbf{p})} \mathbf{p}_r - \mathbf{p}$.

4.2. Optimization

RMM minimizes E (Eq. 3) to jointly learn a compact multilinear model, complete and clean up an incomplete database, and improve semantic correspondence, as outlined in Algorithm 1. The input of RMM is a set of $k \leq d_2 d_3$ shapes $\Omega_X = \{\mathbf{x}_{ie}\}$ with $i \in \{1, \cdots, d_2\}$ and $e \in \{1, \cdots, d_3\}$. All shapes in Ω_X are required to be in full per-vertex correspondence that is possibly inaccurate due to drift. The remaining $d_2 d_3 - k$ shapes $\mathbf{x}_{ie} \notin \Omega_X$ are either corrupt or missing. In contrast to the registered shapes (in Ω_X), for corrupt shapes only partial, possibly noisy data are available that cannot be registered easily. For each corrupt $\mathbf{x}_{ie}$, we require as input an unregistered face scan $\mathbf{s}_{ie} \in \Omega_S$ that is rigidly aligned with the $\mathbf{x}_{ie} \in \Omega_X$. The indices (ie) of $\mathbf{x}_{ie} \in \Omega_X$ and $\mathbf{s}_{ie} \in \Omega_S$ define the initial semantic correspondence. For the remaining shapes (not given in $\Omega_X \cup \Omega_S$) no further information are provided. These shapes are called missing shapes.

After initialization, RMM first optimizes the semantic correspondence as described in Alg. 2. Then, RMM optimizes E for each shape in $\mathcal{X}$ individually. That is, each iteration of the optimization processes all shapes of the database in random order to avoid bias towards specific shapes [15, Chapter 7.1.1]. This shape-wise optimization of E allows to independently handle missing data, corrupt data, and inaccurate vertex correspondence as shown in Alg. 1. Finally, the multilinear model $\mathcal{M}$ is built from $\mathcal{X}$ after all shapes in $\mathcal{X}$ are fixed.

Initialization: For each registered shape $\mathbf{x}_{ie} \in \Omega_X$ a thin-plate spline [16] defines a continuous mapping from $2D$ parameter space to the surface of $\mathbf{x}_{ie}$. The thin-plate spline is computed from a discrete mapping between parameters $\boldsymbol{\alpha}_k \in \mathbb{R}^2$ and vertices $\mathbf{v}_k(\mathbf{x}_{ie})$ of $\mathbf{x}_{ie}$ [4]. Let $\Phi_{ie}(\boldsymbol{\alpha}) = \mathbf{x}_{ie}$ denote the mapping of $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n)^T$ to $\mathbf{x}_{ie}$.

Each missing and corrupt shape $\mathbf{x}_{ie} \notin \Omega_X$ is initialized by the mean over the registered shapes of the same identity i and expression e. Specifically, let

Algorithm 1: RMM

Data: Ω_X; Ω_S
Result: $\mathcal{M}$
Initialization;
for *M iterations* **do**
  `/* Opt. semantic corr. (Alg. 2) */`
  $\min_{\pi} E(\mathcal{X}, 0, 0, 0)$
  `/* Shape-wise optimization */`
  for *each shape* **do**
    if ***x** is missing* **then**
      `/* Estimate missing shape */`
      $\min_{\mathbf{x}} E(\mathcal{X}, 0, w_R, 1)$
    else if ***x** is corrupt* **then**
      `/* Reconstruct corrupt shape */`
      $\min_{\mathbf{x}} E(\mathcal{X}, w_D, w_R, 1)$
    else
      `/* Vertex corr. opt. */`
      $\Phi(\min_{\boldsymbol{\alpha}} E(\mathcal{X}, 0, w_R, 0))$
    end
  end
end
Compute $\mathcal{M}$ (Eq. 1)

Algorithm 2: Semantic correspondence opt.

Data: $\mathcal{X}$; threshold τ
Result: $\mathcal{X}$ relabeled
for *each identity i* **do**
  $\tau_i = \tau$
  $\pi_i := \{\pi_i(1), \ldots, \pi_i(d_3)\} = \{1, \ldots, d_3\}$
  $\pi_{best} = \pi_i$; $E_{best} = E_i = E(\mathcal{X}, 0, 0, 0)$
  for *N_t iterations* **do**
    for *N_s iterations* **do**
      Locally change π_i randomly to π_*
      $\mathcal{X}^* = \mathcal{X}$
      $\mathbf{x}^*_{ie} = \mathbf{x}_{i\pi_*(e)} \;\; \forall e \in \{1, \ldots, d_3\}$
      $E^* = E(\mathcal{X}^*, 0, 0, 0)$
      if $E^* < E_i + \tau_i$ **then**
        $\pi_i = \pi^*$; $E_i = E^*$
      end
      if $E^* < E_{best}$ **then**
        $\pi_{best} = \pi^*$; $E_{best} = E^*$
      end
    end
    $\tau_i = 0.5 \cdot \tau_i$
  end
  $\mathbf{x}_{ie} = \mathbf{x}_{i\pi_{best}(e)} \;\; \forall e \in \{1, \ldots, d_3\}$
end

$\Omega_i := \{\mathbf{x}_{ie} | \forall e \in \{1,\dots,d_3\} : \mathbf{x}_{ie} \in \Omega_X\}$ and $\Omega_e := \{\mathbf{x}_{ie} | \forall i \in \{1,\dots,d_2\} : \mathbf{x}_{ie} \in \Omega_X\}$ denote the set of registered shapes of identity i, and expression e, respectively. The shape $\mathbf{x}_{ie}$ is initialized as

$$\mathbf{x}_{ie} = 0.5 \left(\frac{1}{|\Omega_i|} \sum_{\mathbf{x} \in \Omega_i} \mathbf{x} + \frac{1}{|\Omega_e|} \sum_{\mathbf{x} \in \Omega_e} \mathbf{x} \right), \tag{8}$$

where $|\Omega_i|$ and $|\Omega_e|$ denote the cardinality of Ω_i and Ω_e, respectively. We call this initialization technique averaging scheme (AVS) in the following. We use the result of AVS as reference shape $\widetilde{\mathbf{x}}$ in E_R.

Semantic correspondence optimization: To optimize semantic correspondence, RMM minimizes $E(\mathcal{X}, 0, 0, 0) = E_C$. As joint optimization of the semantic correspondence over all data is infeasible, we optimize E for each identity individually. Note that as for the shape-wise optimization, E still depends on all shapes, and hence the method remains a groupwise optimization. To avoid any bias towards specific identities, the order of the processed identities in each iteration is chosen randomly.

For each identity i we search for the permutation $\pi_i = \{\pi_i(1), \dots, \pi_i(d_3)\}$ with $\pi_i(e) \in \{1,\dots,d_3\}$ of the expressions of i that minimizes E. Note that π_i only changes the labeling of the expressions for each identity; the geometry of the shapes remains unchanged. Due to the domain of π_i, this is an integer problem.

Integer problems are often solved by discretization, *i.e.* instead of the integer problem $\pi \subseteq \mathbb{Z}$ a discretized problem $\pi \subseteq \mathbb{R}$ is optimized. The optimization of the discretization of E with a local method such as L-BFGS like in the other RMM optimization steps fails due to many local minima.

Instead, we directly solve the integer problem. We optimize E with a threshold accepting (TA) method [28] as outlined in Algorithm 2. Given an initial threshold τ, the iteratively decreasing τ equates to the cooling schedule of simulated annealing. TA uses two iterations, one to lower the threshold, and one for optimization for a certain threshold. TA stores the minimum E_{best} of E together with the corresponding best permutation π_{best}. In one optimization iteration, π_i is randomly altered to π_* by permuting 10% of the elements of π_i, the expressions of i in $\mathcal{X}$ are permuted accordingly to $\mathcal{X}^*$, and E is evaluated for $\mathcal{X}^*$. Depending on τ_i, π^* is used as starting point for the next iteration. If a new minimum is found, E_{best} and π_{best} are updated. Finally, the expressions of i in $\mathcal{X}$ are permuted by π_{best}. The threshold τ can be chosen automatically.

Vertex correspondence optimization: To optimize the vertex correspondence of $\mathbf{x}_{ie} \in \Omega_X$, RMM minimizes $E(\mathcal{X}, 0, w_R, 0) = E_C + w_R E_R^0$ by reparametrizing $\mathbf{x}_{ie}$ [4]. As the energy E is analytically differentiable with respect to the parameters $\boldsymbol{\alpha}$ of $\mathbf{x}_{ie}$, E is minimized in parameter space using L-BFGS [25]. The optimized shape $\mathbf{x}_{ie}$ is updated as $\mathbf{x}_{ie} = \Phi_{ie}(\boldsymbol{\alpha})$.

Missing data estimation: To estimate a missing shape, RMM minimizes $E(\mathcal{X}, 0, w_R, 1) = E_C + w_R E_R^1$. In contrast to the vertex correspondence optimization, E is minimized in Euclidean vertex space using L-BFGS rather than in parameter space. That is, during optimization each vertex of the missing shape moves in $\mathbb{R}^3$ to minimize E. This is required as the geometry of the missing shape is unknown.

Corrupt data estimation: To estimate the shape from a corrupt face scan $\mathbf{s} \in \Omega_S$, RMM minimizes $E(\mathcal{X}, w_D, w_R, 1) = E_C + w_D E_D + w_R E_R^1$. To be robust to erroneous initial alignments, the alignment of $\mathbf{s}$ is refined using an iterative closest point algorithm. As for the missing data estimation, E is minimized in Euclidean vertex space using L-BFGS.

5. Evaluation

This section evaluates the robustness of RMM to missing data, to corrupt data, and to wrong semantic correspondence. The supplementary video shows further results.

Data: We evaluate RMM on two publicly available 3D face databases, the BU-3DFE database [35] and the Bosphorus database [31]. The BU-3DFE database contains scans of 100 identities each in neutral expression and the six prototypic expressions anger, disgust, happiness, sadness and surprise. The Bosphorus database contains scans of 105 identities in up to 35 expressions, 4 variants of facial occlusions, and up to 13 head poses. Both databases are initially registered with an automatic template fitting method [29] that uses the landmarks provided with the databases.

For BU-3DFE we randomly choose 50 identities and use 7 expressions, the neutral expression and the highest level of each prototypic expression. For Bosphorus we randomly choose 30 identities and use 17 action units. We call these subsets BU-3DFE set and Bosphorus set, respectively.

The robustness of RMM to missing data is evaluated on the BU-3DFE set and the Bosphorus set, each with randomly removed shapes. For evaluation, we use for both datasets configurations with 1%, 5%, 10%, 25%, and 50% of the shapes missing.

The robustness of RMM to corrupt data is evaluated on the BU-3DFE set and the Bosphorus set, each with subsets of corrupt data due to simulated and real partial occlusions. While the BU-3DFE set is only corrupted by simulated occlusions, the Bosphorus set contains noisy and partially occluded face scans, which we use to substitute the complete scans in our experiments. The occlusions are selected to affect the facial regions shown in the top row of Figure 4. We use for both datasets configurations with 1%, 5%, 10%, 25%, and 50% of corrupt shapes during evaluation.

The robustness of RMM to wrong semantic correspondence is evaluated on the BU-3DFE set and the Bosphorus set, each with a subset of randomly generated erroneously labeled expressions. To simulate erroneously labeled ex-

GT AVS HaLRTC RMM

Figure 2. Comparison of robustness to missing data. From left to right: Ground truth (GT). Averaging scheme (AVS). HaLRTC [26].

pressions, the wrong semantic correspondence subsets consist of randomly chosen identities, where the expressions are randomly permuted. We use for both datasets configurations with randomly permuted expression labelings of 5%, 10%, 25%, 50%, and 100% of the identities.

Parameter settings: For all evaluations we fix all of the parameters heuristically. The parameters w_D and w_R (Eq. 3) control the influence of the data and regularization terms, respectively. We choose $w_D = 1e-3$ and $w_R = 20$ to reconstruct missing and corrupt data, and $w_R = 0.5$ to optimize vertex correspondence. For databases that contain less corrupt data than in our experiments, w_D could be set higher and w_R could be set lower to allow the recovery of more facial detail. The parameters δ_2 and δ_3 are used to avoid singularities of E_C (Eq. 4), and we choose them as $\sigma_2 = \sigma_3 = 0.01$ as in previous work [4]. The parameter ρ (Eq. 5) relates directly to the size of the face, and can be fixed at 5 mm. The parameters M (Alg. 1), N_t, and N_s (Alg. 2) control the number of iterations performed, and allow to tradeoff running time and accuracy. We choose them as $M = 15$, $N_t = 10$, and $N_s = 200$.

Reproducibility: To facilitate the use of our framework on new databases, we make our code available [5]. Further, we publish a multilinear model learned using RMM from the combination of all 100 identities in 7 expressions of the BU-3DFE database and all 105 identities in 23 expressions of the Bosphorus database. The different expression sets of both databases and the missing shapes of the Bosphorus database cause a large portion of the joint database to be missing (2205 shapes of 4715 missing). RMM successfully learns a model for these data by estimating the missing data.

5.1. Robustness to missing data

Objective function: To study the influence of E_R on E for missing data completion, we optimize E with ($w_D = 1e-3$) and without ($w_D = 0$) regularization. During optimization, each shape has only limited influence on E. We

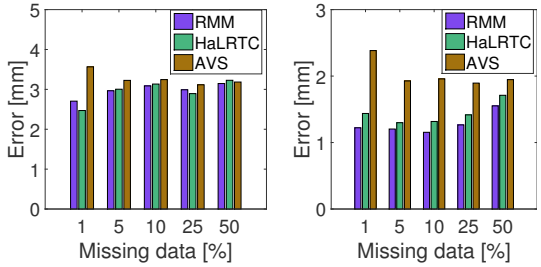


Figure 3. Median error of HaLRTC [26] and AVS for different missing data configurations compared to RMM. Left: BU-3DFE set. Right: Bosphorus set.

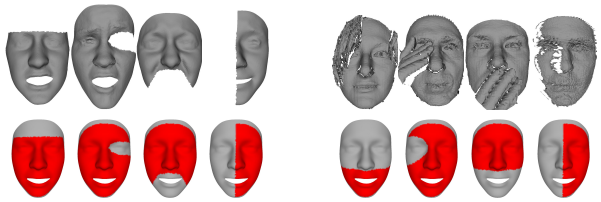


Figure 4. Samples of corrupt data (top) and corresponding valid regions (red) for each type of occlusion used for error measure (bottom). Left: Simulated occlusions. Right: Real occlusions in the Bosphorus database.

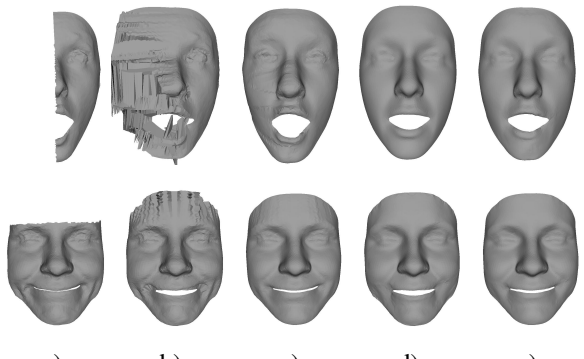
observed that the shape-wise optimization of E_C overcompensates for the limited influence of few shapes and may produce unlikely shapes. The regularization successfully prevents this overcompensation as it penalizes strong local distortions.

Comparison: We compare RMM to the ground truth shape, to AVS, and to the result of the state-of-the-art tensor completion method HaLRTC [26]. Figure 2 visually compares the completed shapes. While HaLRTC and RMM result in a better estimation of the missing shape than AVS, they perform rather similarly. Figure 3 shows the median error, measured as the distance of all completed shapes to the ground truth for all configurations. HaLRTC and RMM perform better than AVS if up to 10% of the data are missing. While for the Bosphorus set RMM performs slightly better than HaLRTC, both methods perform overall similar.

Summing up, given a dataset with missing data, RMM reconstructs the missing data well.

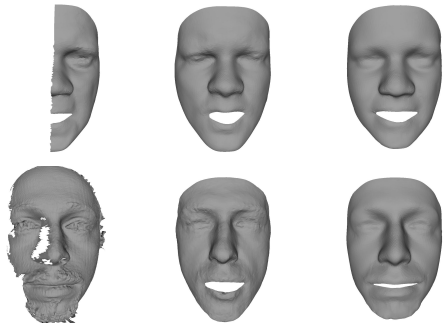
5.2. Robustness to corrupt data

Objective function: To show the individual influence of each term of E to reconstruct corrupt data, we optimize E with different combinations of energy terms. Figure 5 visually compares the results for the different combinations. The optimization of E_D closely reconstructs $\mathbf{s}$ in non-corrupt regions, but corrupt regions produce strong artifacts, and the expressions are not always well reconstructed. The optimization of $E_C + w_D E_D$ reconstructs the shape and the expression of $\mathbf{s}$ well in non-corrupt regions, and gives a



a) b) c) d) e)

Figure 5. Influence of each term in E (Eq. 3) to reconstruct corrupt data (10% corrupt). From left to right: a) Corrupt scan $\mathbf{s}$. Optimization of: b) E_D. c) $E_C + w_D E_D$. d) $w_D E_D + w_R E_R$. e) RMM. Top: BU-3DFE set. Bottom: Bosphorus set.



s HaLRTC + MM RMM

Figure 6. Comparison with combination of HaLRTC [26] and multilinear model (MM) [3] to reconstruct corrupt data (10% corrupt). Top: BU-3DFE set. Bottom: Bosphorus set.

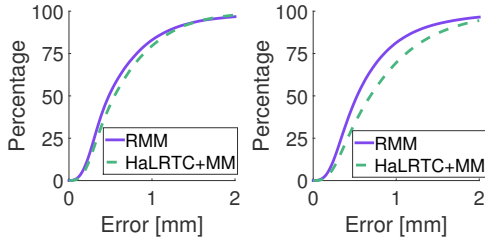


Figure 7. Cumulative error of combination of HaLRTC [26] and multilinear model [3] for 10% corrupt data compared to RMM. Left: BU-3DFE set. Right: Bosphorus set.

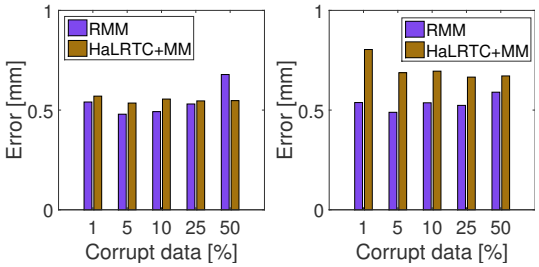


Figure 8. Median error of combination of HaLRTC [26] and multilinear model [3] for different corrupt data configurations compared to RMM. Left: BU-3DFE set. Right: Bosphorus set.

reasonable prediction of the shape for corrupt regions, but corrupt regions contain artifacts. Note that E_C is unable to regularize E_D sufficiently as (even strong) local distortions in the reconstruction only have a negligible influence on E_C. The optimization of $w_D E_D + w_R E_R$ avoids the artifacts in corrupt regions, but the facial expression is not reconstructed well. The full optimization of E reconstructs the facial expression well and is robust to corrupt data.

Comparison: As statistical face models are known to be robust to partial occlusions and noise (*e.g.* [1, 7]), we compare RMM to a multilinear model reconstruction of the corrupt data. Since the multilinear face model requires a complete data tensor for training, the data tensor is completed using HaLRTC [26]. A multilinear face model is trained that keeps 95% of the identity and expression variations on the completed data, and all corrupt shapes of the dataset are reconstructed. We call this combination of existing methods HaLRTC+MM in the following. In contrast to RMM, HaLRTC+MM gets facial landmarks for fitting to initialize the expression.

Figure 6 visually compares HaLRTC+MM and RMM for 10% corrupt data. While both methods are robust to corrupt data, RMM better reconstructs the facial expression. Further, RMM better reconstructs the facial shape, *e.g.* at the nose. Since the distance-to-data measure is only a valid error measure in non-occluded regions, we define for each type of occlusion a valid region as visualized in the bottom of Figure 4. The error measure then only uses vertices within the valid regions. Figure 7 shows the cumulative error plots for both datasets with 10% corrupt data. For both datasets RMM performs better than HaLRTC+MM. For most other configurations RMM performs better than HaLRTC+MM as shown in Figure 8. For the BU-3DFE set with 50% corrupt data RMM reconstructs a few expressions incorrectly due to the sparse sampling of the data, while HaLRTC+MM better reconstructs the expression thanks to the additionally provided landmarks. To reconstruct corrupt data, RMM assumes AVS to give a reasonable initialization of the expression of $\mathbf{s}$ as the iterative nearest neighbor terms E_D is known to only converge locally. This requires the expression of $\mathbf{s}$ to be similar to the expressions in Ω_X. Using landmarks for initialization could help RMM to reconstruct extreme expressions more reliably.

Summing up, given a dataset with corrupt data, RMM provides a reconstruction that preserves facial details while being robust to partial occlusions and noise.

5.3. Robustness to wrong semantic correspondence

We evaluate the optimized semantic correspondence with the measures compactness, generalization, and specificity [15, Chapter 9.2] that are widely used to quantify the quality of statistical models.

Compactness measures the amount of variability the

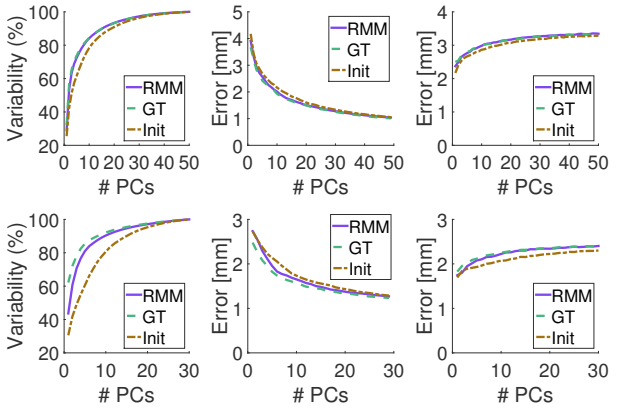


Figure 9. Comparison to ground truth (GT) for randomly permuted labeling of 50% of the identities before (Init) and after optimization (RMM). Left: Compactness. Middle: Generalization: Right: Specificity. Top: BU-3DFE set. Bottom: Bosphorus set.

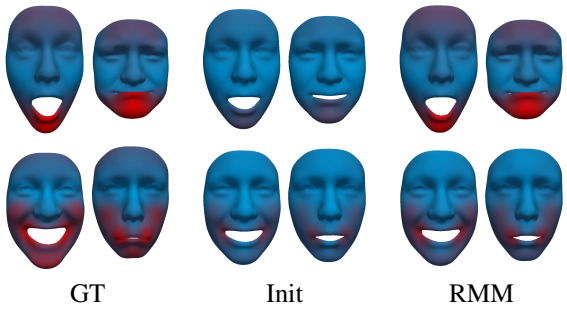


Figure 10. Expression variations of two expression components (rows) for randomly permuted labeling of 50% of the identities for the BU-3DFE set. The magnitude of the vertex displacement is color coded from blue (zero) to red (maximum). Left: ground truth (GT). Middle: Erroneously labeled data (Init). Right: RMM.

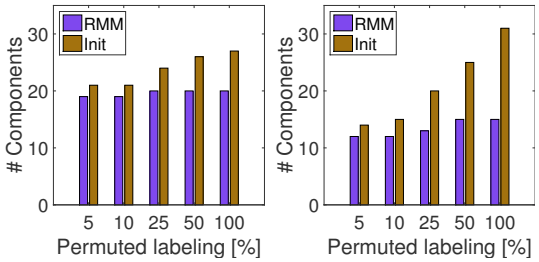


Figure 11. Number of components needed to keep 90% of the data variability before (Init) and after optimization (RMM). Left: BU-3DFE set. Right: Bosphorus set.

model explains. A model with high compactness requires fewer components to describe the same amount of variability than a less compact model. The compactness for k components is computed as $C(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d_n} \lambda_i}$, where λ_i denotes the i-th eigenvalue of $\mathbf{D}_2$ for identity mode, and $\mathbf{D}_3$ for expression mode, respectively.

Generalization measures the ability of a model to describe unseen data that are not part of the training. The generalization error is computed with a leave-one-out reconstruction. For identity mode, all but one identity of the training data are used to build a multilinear model and all excluded shapes are then reconstructed. This is repeated for all identities. The generalization error then measures the average per-vertex errors of all reconstructions.

Specificity measures the ability of the model to only produce valid instances of the object class. To compute the specificity error, we randomly choose 10000 samples from identity and expression space, reconstruct each sample using Eq. 2, and measure for each reconstructed sample the per-vertex distance to the closest training shape. The specificity error then measures the average error over all samples.

Figure 9 shows the influence of wrong semantic correspondence on compactness, generalization and specificity (identity mode) for BU-3DFE set (top) and the Bosphorus set (bottom) for randomly distorted expression labelings of 50% of the identities. Compared to the ground truth (GT), the model with wrong semantic correspondence (Init) is less compact, less general, and more specific. After optimization (RMM) the model becomes significantly more compact, more general, and less specific, comparable to GT. Hence, after optimizing the semantic correspondence, the model requires less components to capture the same variability of the data.

When 50% of the data are permuted, to keep 90% of the data variability before optimization, a total of 26 and 25 components are necessary for the BU-3DFE and Bosphorus sets, respectively, while after optimization 20 and 15 components suffice for the BU-3DFE and Bosphorus sets, respectively. Figure 10 shows the variations of two expression components. The variations of the model increase significantly after optimization. For the other configurations RMM also gives significant improvements (see Figure 11).

Summing up, given a dataset with wrong semantic correspondence, RMM improves the semantic correspondence, and results in a more compact model.

6. Conclusion

We have presented the first groupwise multilinear model learning framework that is robust to missing data, corrupt data, wrong semantic correspondence, and inaccurate vertex correspondence. This allows to build highly accurate multilinear face models from existing 3D face databases. We have evaluated our framework on two databases with multiple levels of missing data, corrupt data caused by noise and partial occlusions, and erroneously labeled expressions. We have shown that our framework completes data comparable to state-of-the-art tensor completion methods, that it reconstructs corrupt data better than state-of-the-art methods, and that the quality of the learned model increases significantly for erroneously labeled expressions.

References

[1] V. Blanz, K. Scherbaum, and H.-P. Seidel. Fitting a morphable model to 3D scans of faces. In *ICCV*, 2007.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

[3] T. Bolkart and S. Wuhrer. 3D faces in motion: Fully automatic registration and statistical analysis. *CVIU*, 131:100–115, 2015.

[4] T. Bolkart and S. Wuhrer. A groupwise multilinear correspondence optimization for 3D faces. In *ICCV*, pages 3604–3612, 2015.

[5] T. Bolkart and S. Wuhrer. Robust multilinear model framework, 2016. http://rmm-framework.gforge.inria.fr/.

[6] A. Brunton, T. Bolkart, and S. Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *ECCV*, pages 297–312, 2014.

[7] A. Brunton, A. Salazar, T. Bolkart, and S. Wuhrer. Review of statistical shape spaces for 3D data with comparative analysis for human faces. *CVIU*, 128(0):1 – 17, 2014.

[8] O. Burghard, A. Berner, M. Wand, N. J. Mitra, H.-P. Seidel, and R. Klein. Compact part-based shape spaces for dense correspondences. *CoRR*, abs/1311.7535, 2013.

[9] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011.

[10] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[11] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D shape regression for real-time facial animation. *TOG (Proc. SIGGRAPH)*, 32(4):41:1–41:10, 2013.

[12] J.-H. Chen, K. C. Zheng, and L. G. Shapiro. 3D point correspondence by minimum description length in feature space. In *ECCV*, pages 621–634, 2010.

[13] Y.-L. Chen, C.-T. Hsu, and H.-Y. Liao. Simultaneous tensor decomposition and completion using factor priors. *PAMI*, 36(3):577–591, 2014.

[14] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *TOG (Proc. SIGGRAPH Asia)*, 30(6):130:1–10, 2011.

[15] R. Davies, C. Twining, and C. Taylor. *Statistical Models of Shape: Optimisation and Evaluation*. Springer, 2008.

[16] I. Dryden and K. Mardia. *Statistical shape analysis*. Wiley, 1998.

[17] J. Guo, X. Mei, and K. Tang. Automatic landmark annotation and dense correspondence registration for 3D human facial images. *BMC Bioinf.*, 14(1), 2013.

[18] D. Hirshberg, M. Loper, E. Rachlin, and M. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *ECCV*, pages 242–255, 2012.

[19] P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *CVPR*, pages 1675–1683, 2015.

[20] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH*, pages 105–114, 1998.

[21] A. C. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by direct optimization. *Med. Image Anal.*, 2(4):303 – 314, 1998.

[22] L. D. Lathauwer. *Signal processing based on multilinear algebra*. PhD thesis, K.U. Leuven, Belgium, 1997.

[23] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng. Practice and theory of blendshape facial models. In *EG - STARs*, 2014.

[24] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *TOG (Proc. SIGGRAPH)*, 32(4):42:1–42:10, 2013.

[25] D. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Math. Prog.: Series A and B*, 45(3):503–528, 1989.

[26] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *PAMI*, 35(1):208–220, 2013.

[27] I. Mpiperis, S. Malassiotis, and M. G. Strintzis. Bilinear models for 3-D face and facial expression recognition. *IFS*, 3:498–511, 2008.

[28] V. Nissen and H. Paul. A modification of threshold accepting and its application to the quadratic assignment problem. *OR Spektrum*, 17(2-3):205–210, 1995.

[29] A. Salazar, S. Wuhrer, C. Shu, and F. Prieto. Fully automatic expression-invariant face correspondence. *MVAP*, 25(4):859–879, 2014.

[30] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image Vision Comput.*, 30:683–697, 2012.

[31] A. Savran, N. Alyuöz, H. Dibeklioglu, O. Celiktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In *BIOID*, pages 47–56, 2008.

[32] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 61:611–622, 1999.

[33] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *TOG (Proc. SIGGRAPH)*, 24(3):426–433, 2005.

[34] F. Yang, L. Bourdev, J. Wang, E. Shechtman, and D. Metaxas. Facial expression editing in video using a temporally-smooth factorization. In *CVPR*, pages 861–868, 2012.

[35] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3D facial expression database for facial behavior research. In *FG*, pages 211–216, 2006.

[36] Z. Zhang and S. Aeron. Exact tensor completion using t-svd. *CoRR*, abs/1502.04689, 2015.