

Improving 3D Facial Action Unit Detection with Intrinsic Normalization

Eric Yudin
eyudin@cs.technion.ac.il

Aaron Wetzler
twerd@cs.technion.ac.il

Matan Sela
matansel@cs.technion.ac.il

Ron Kimmel
ron@cs.technion.ac.il

Geometric Image Processing Lab
Technion - Israel Institute of Technology
Technion City, Haifa, Israel

Abstract

Data normalization techniques are commonly used to reduce intra-class variance while preserving inter-class differences that aid in classification and regression tasks. Such practices bring all data examples to similar scale and range, and help to decrease the dimensionality of categorization problems. In this paper we describe a novel use of a geometric framework to extend the concept of data normalization to the domain of functions that lie on surfaces. In this context, normalization is posed as an embedding of all example functions into Riemannian manifolds nearly isometric to one another. Using geometric tools, we propose an implementation for the case of discretized functions on triangulated meshes. To demonstrate the proposed framework we apply it as a preprocessing step to the task of automatic facial action unit detection for depth maps from a 3D scanner. We show how this subject-independent representation produces more accurate action unit classification results. We also empirically demonstrate that utilizing this intrinsic normalization technique indeed improves action unit recognition performance.

1 Introduction

Facial expressions constitute a significant component of human communication. They contain non-verbal elements that enhance and provide deeper context to words actually vocalized, and can even convey meaning in the absence of speech altogether. In addition, certain facial movements can correspond with various mental disorders and therefore decoding these correspondences can help simplify diagnostic processes that currently consist of lengthy and somewhat subjective psychological evaluations. The work of Ekman and Friesen [8] identified six universal expressions that do not vary across cultures. These are happiness, sadness, surprise, fear, disgust and anger. They also found that no other correspondence between facial configuration and communicative intent remained valid universally. This meant that such correspondences could be culturally dependent, or even individually dependent.

Ekman and Friesen recognized the need to codify all possible facial configurations and the Facial Action Coding System (FACS) [9] came from this insight. FACS considers the

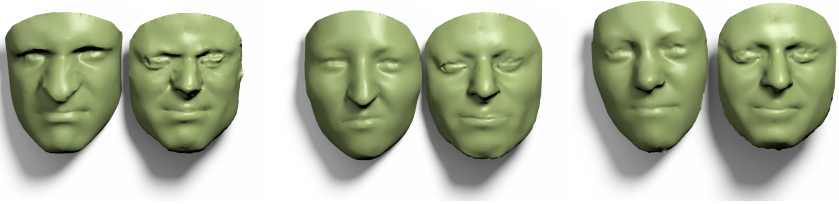


Figure 1: Illustration of three couples of faces undergoing intrinsic normalization, with the non-normalized face on the left, and normalized one on the right. All examples perform AU9 (nose wrinkler). Note that all of the normalized faces have similar physiognomies, yet, each bears the expression of its non-normalized counterpart.

face as groups of muscles, and modularly describes each local movement as one of 44 possible action units, or AUs. Together with a range of five possible intensities, AUs form the building blocks for any configuration expressible by the human face. Mathematically, action units can be seen to form a sparse basis that spans the space of facial expressions. With FACS, one can describe any facial expression as a sparse combination of AUs. Compared to a representation where each expression describes the movement of groups of 3D points, FACS significantly lowers the dimensionality of the space of possible expressions. FACS defines an AU in terms of how much effort a certain muscle group is making with respect to a given subject’s neutral face, which implies independence of subject *physiognomy*, the inherent structure and characteristics of an individual’s face. As an example, the same AUs and intensities will comprise a slight smile on both of, say, a long-faced and a heavily-jowled subject. This further implies that there is no guarantee of linearity. We therefore cannot simply model the effect of an AU as the addition of a field of deformation vectors; the required field would need to vary with physiognomy. In this paper, we directly address the difficulty posed by the influence of physiognomy-based variability on AUs. We try to answer the question: what if we could treat all examples as though they were performed by a single subject? Our primary contribution is therefore a framework that provides this capability and yields state-of-the-art results on the popular Bosphorus dataset [13]. In the next section we discuss previous approaches to the problem of AU detection. In Section 3 we introduce the concept of intrinsic normalization, giving motivating examples and describing the proposed algorithm. Section 4 discusses our experimental results. Finally, we conclude with Section 5.

2 Related Work

For decades, research in action unit detection was restricted primarily to analysis of two-dimensional RGB images. The advent of high-resolution, efficient, cheap 3D scanners, graphics processing hardware (GPUs) and accompanying high-level programming languages in the early- to mid-2000s brought interest and practicality to 3D data processing. While typically somewhat more time-consuming and complicated to process, and while 3D scanners do introduce their own errors and sensor noise, the advantages of this kind of data are significant. First, 3D depth scanning is illumination independent, allowing processing to occur in any kind of lighting conditions. Illumination dependence is a problem that plagues

traditional 2D color processing, and has generated much research on its own. Secondly, deformations are better described with the language of surfaces and geometry existing in three-dimensional ambient space. Further, true 3D data allows a facial analysis system to calculate much more accurate values for properties such as depth and curvature, instead of relying on rough approximations based on 2D RGB images. Much of the art in machine learning lies in the selection of features that represent the original data informatively and discriminatively, and a classifier that uses these features to build mathematical models to separate examples into the desired groups. Sun *et al.* [14] used an active appearance model to track 83 predefined positions and then a set of Hidden Markov Models to classify the displacements of these points across a video of 6 frames. Zhao *et al.* [15] used a patch-based ‘Statistical Facial feAture Model’ (SFAM) to classify features based on three types of data: landmark configuration, local texture, and local geometry. Savran & Sankur also primarily consider geometric features [16], but first align their data examples with a non-rigid registration algorithm, such that changes are properly measured with respect to consistent points on the face.

Finally, several works have devised a family of features based on Local Binary Patterns (LBP) [9]. In this paradigm, each pixel is assigned an N -bit value corresponding to the fluctuations of some property of the N surrounding pixels with respect to that center pixel. It has proven quite successful, generating impressive results with features that are relatively simple to compute. The LBP model was originally developed for use with 2D images, though since then, a wide array of variants have appeared in the literature. Huang *et al.* used 3D Local Binary Patterns, incorporating depth values into the original feature [9]. Sandbach *et al.* used Local Normal Binary Patterns, which measure fluctuations in normal angles instead of brightness, in addition to a wider array of variations on the LBP theme [17]. Recently, Bayramođlu *et al.* achieved state-of-the-art results using Center-Symmetric 3D Local Binary Patterns (CS-3DLBP) and Ratio-Based Geometry (RBG) features in conjunction with the Random Forests classifier [18]. CS-3DLBP itself measures fluctuations of normal angles, and ‘Center-Symmetric’ refers to its comparison of normals on opposing sides of the pixel of interest, reducing the length of the bit pattern to $N/2$. CS-3DLBP is defined as

$$\begin{aligned}
 \text{CS-3DLBP}_{R,N,\theta_t}(p) &= \sum_{i=0}^{N/2-1} s(\vec{n}_i, \vec{n}_{i+N/2}) 2^i \\
 s(\vec{n}_i, \vec{n}_j) &= \begin{cases} 1 & \angle(\vec{n}_i, \vec{n}_j) \geq \theta_t \\ 0 & \text{otherwise} \end{cases} \\
 \angle(\vec{n}_i, \vec{n}_j) &= \arccos \frac{\langle \vec{n}_i, \vec{n}_j \rangle}{\|\vec{n}_i\| \|\vec{n}_j\|}, \quad (1)
 \end{aligned}$$

where \vec{n}_i are surface normals at N equally-spaced points on a circle of radius R surrounding point p , and θ_t is a threshold angle. The result is an $N/2$ -bit binary number associated with p . The method does not use these values directly, however; the input depth map is instead broken into a 5×5 grid, and then histograms of the CS-3DLBP values in each grid location are concatenated to form the final feature vector. Since Equation (1) can generate $2^{N/2}$ unique bit patterns, a feature vector for the common case of $N = 8$ will be $16 \times 5 \times 5 = 400$ values long. The addition of the RBG feature set extends the vector by another 25 values.

Most of the above studies were performed using the Bosphorus database [13] as testing data. Bosphorus is a rich facial expression database comprised of 105 subjects each performing up to 24 AUs for a total of 4666 facial scans. Each example includes a depth map,

an RGB image, and up to 24 facial landmarks. In addition, each example has been labelled by a certified FACS coder, providing a ground truth for action unit detection research. Many of these methods contain some pre-processing step to bring example faces into a canonical form. This process typically involves fixing position and orientation, usually via Principal Component Analysis and/or Iterated Closest Point, and scaling along the x-, y-, and z-axes. The intent is to remove variation that is specific to the examples themselves but that is not characteristic of the class of interest.

However, especially in the case of facial analysis, each subject’s physiognomy imposes much more variability than can be removed with the above techniques. Savran and Sankur [10] attempt to reduce subject variability by mapping surface feature values to a plane and using a 2D non-rigid registration algorithm to achieve correspondence between points. The approach achieves good positional alignment across faces in spite of physiognomy. However, their method does not remove the geometric influence of physiognomy on the feature values. Consequently, while comparing the proper points improves effectiveness of the method, the numerical effect of physiognomy remains. Most previous attempts therefore allow the variability of subjects to affect the variance of observations of a given AU. In the case of the RGB feature set of [10], the goal is to introduce subject independence by using ratios instead of absolute measurements. Using RGB does improve their results, however, it still leaves the other 400 features unnormalized, indicating that there is still room for improvement. Ideally, analysis would involve a single subject performing all possible variations of a given expression. This essentially amounts to performing a scale normalization at each point. In geometric terms this means normalizing the metric at each point on the facial surface.

3 Intrinsic Normalization

Normalization is known to be an effective means of reducing the variance of observations of a class of phenomena [11]. It brings examples into a canonical form by removing variation which is not a manifestation of the class of interest. In doing so, it reduces the dimensionality and the number of degrees of freedom necessary for a mathematical model to successfully identify an instance of the class. 2D digit recognition can serve as a concrete motivating example. Suppose we are attempting to recognize the digit ‘8’. Examples of this digit will vary in many different ways, like shape, smoothness, and connectivity. With ever more examples our classifier builds a better picture of exactly how this class can vary. However, there is some variability that does not interest us. If two examples have been drawn identically but appear shifted within the image, we would still like to consider them the same version of the digit. We therefore center all input examples, as the shifting should not contribute to the classifier’s knowledge of what an ‘8’ looks like. In fact, considering the shift unnecessarily increases dimensionality of the space, requiring much more data for accurate classification. The same notion applies for rotation and scale, though only to an extent; over-normalization can cause false alarms by allowing the classifier to recognize members of different classes. For example, if we allowed rotational normalization of 90° , our classifier would now accept the infinity symbol ‘ ∞ ’ as part of our class. Good normalization attempts to reduce unnecessary variation as much as possible, but only as far as it can maintain the boundaries of the class of interest. For the application of facial expression recognition, it is prudent to center, align and scale along the axes. This removes dimensionality that does little to inform a classifier of a classifier’s inherent structure, and does not act to create false positives in the process. However, we can do better. As we are trying to isolate all manifestations of

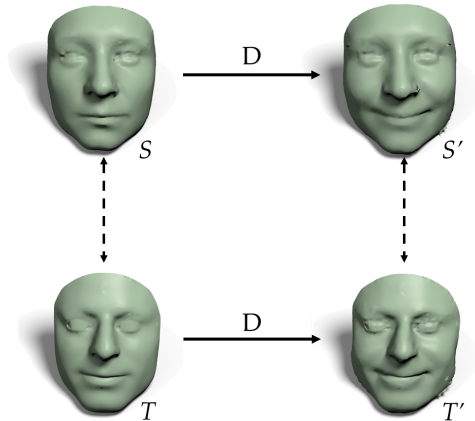


Figure 2: Flowchart for intrinsic normalization. Input meshes S and S' represent the subject's neutral and expression faces, respectively, and D is the deformation between them. These are registered with template mesh T and smoothed. The resulting meshes are then fed into the deformation transfer algorithm to produce T' , the normalized face. T' thereby combines the physiognomy of T and the AUs of S' .

an expression independent of subject, we would like to normalize facial geometry in such a way as to remove all variability related to an individual's physiognomy. To this end, we consider Bronstein *et al.*'s Isometric Model of facial expression [10]. Given a smooth, compact, connected two-dimensional Riemannian manifold \mathcal{S} , the authors assert that, since facial skin stretches only slightly during an expression, expressions of a single subject are approximately intrinsically equivalent. They go on to attribute the intrinsic geometry of the facial surface to the physiognomy, and the extrinsic geometry to the expression. As their goal is facial recognition, the authors use this model to transform faces to a representation that is invariant to facial expression. This allows them to distinguish variability in physiognomy. Here, we do exactly the opposite. We transform the face into a representation agnostic to physiognomy.

We can use deformation transfer [10] to achieve this objective. Given our source shape \mathcal{S} , a target shape \mathcal{T} , and a deformed source shape \mathcal{S}' , deformation transfer finds an analogously-deformed \mathcal{T}' . The process is based on the gradient-based deformation method of Yu *et al.* [14]. We begin by expressing \mathcal{S} as

$$\mathcal{S}(u, v) = \{x_{\mathcal{S}}(u, v), y_{\mathcal{S}}(u, v), z_{\mathcal{S}}(u, v)\} \quad (2)$$

where $x_{\mathcal{S}}$, $y_{\mathcal{S}}$, and $z_{\mathcal{S}}$ are coordinate functions on \mathcal{S} . The gradient fields of these functions are $\nabla x_{\mathcal{S}}$, $\nabla y_{\mathcal{S}}$, and $\nabla z_{\mathcal{S}}$, each of which is a vector field tangent to \mathcal{S} . The deformation D that brings $\mathcal{S} \rightarrow \mathcal{S}'$ transforms the gradient field such that at each point p , $\nabla x_{\mathcal{S}}(p) \rightarrow D_p(\nabla x_{\mathcal{S}}(p))$, and likewise for $y_{\mathcal{S}}$ and $z_{\mathcal{S}}$. We now use D_p to govern the analogous deformation of $\mathcal{T} \rightarrow \mathcal{T}'$. To do so, we seek $\mathcal{T}' = \{x_{\mathcal{T}'}, y_{\mathcal{T}'}, z_{\mathcal{T}'}\}$ whose coordinate functions agree with $\nabla x_{\mathcal{T}}(p) \rightarrow D_p(\nabla x_{\mathcal{T}}(p))$ in a least-squares sense. The gradient-based deformation

method accomplishes this for us by minimizing the energy

$$E = \int_{\mathcal{T}} \|\nabla x_{\mathcal{T}'} - D_p(\nabla x_{\mathcal{T}})\|^2 dA, \quad (3)$$

and likewise for $y_{\mathcal{T}'}$ and $z_{\mathcal{T}'}$. Here, dA is an area element about the point $p \in \mathcal{T}$. To minimize, we find the Euler-Lagrange equation for (3), given by

$$\Delta x_{\mathcal{T}'} = \text{div} D_p \nabla x_{\mathcal{T}}, \quad (4)$$

and likewise for $y_{\mathcal{T}'}$ and $z_{\mathcal{T}'}$. Here, Δ is the Laplace-Beltrami operator, the generalization of the Laplacian to non-orthogonal domains, and Equation (4) is the Poisson equation, whose discrete form is known to be numerically solvable. In our case, \mathcal{S} will be the neutral face of the example subject, \mathcal{S}' will be the same subject expressing the AU of interest and \mathcal{T} will be some chosen template face. With the deformation transfer algorithm we can obtain \mathcal{T}' , the template face deformed to perform the same AU. In doing so, we have normalized the physiognomic contribution to the facial geometry to that of our template face. Performing this operation for all input fixes this component across examples, thereby allowing the classifier to better learn the geometric contribution arising from the AU itself.

Deformation transfer requires dense correspondence between \mathcal{S} , \mathcal{T} and \mathcal{T}' . To achieve this, we utilize a non-rigid registration algorithm based on that of Weise *et al.* [15] to align all face instances. Non-rigid registration (NRR) operates by deforming one mesh into another, producing a dense point correspondence, or alignment, in the process. Here, we deform our chosen template face to align with all input examples. Roughly speaking, deformation transfer “copies” gradient deformations from one pair of meshes and “pastes” them onto a second. Any noise in either the source or target meshes will be compounded during deformation transfer. We must therefore perform smoothing on all surfaces. However, running NRR after doing so risks re-introducing noise into the mesh. We therefore smooth immediately before deformation transfer. One added constraint is that such a smoothing process should not significantly change the geometry, lest we distort the alignment obtained during the registration process. To this end we use Laplace Beltrami smoothing [9]. This method updates the N vertex positions $\{v_1, \dots, v_N\}$ via the algorithm

$$v_i = v_i + h\lambda \Delta v_i. \quad (5)$$

Here, $\Delta v_i = -2H\mathbf{n}$, where H is the mean curvature and \mathbf{n} is the normal to the surface at v_i . Lastly, we impose the constraint

$$\partial \mathcal{S}_{\text{smooth}} = \partial \mathcal{S} \quad (6)$$

Algorithm 1 Intrinsic Normalization

```

1: function INTNORMALIZE( $\mathcal{S}, \mathcal{S}', T$ )
2:    $\tilde{\mathcal{S}} \leftarrow \text{NRR}(T, \mathcal{S})$ 
3:    $\tilde{\mathcal{S}}' \leftarrow \text{NRR}(T, \mathcal{S}')$ 
4:    $\tilde{\mathcal{S}}_{\text{smooth}} \leftarrow \text{LBSmooth}(\tilde{\mathcal{S}})$ 
5:    $\tilde{\mathcal{S}}'_{\text{smooth}} \leftarrow \text{LBSmooth}(\tilde{\mathcal{S}}')$ 
6:    $T' \leftarrow \text{DefTrans}(\tilde{\mathcal{S}}_{\text{smooth}}, \tilde{\mathcal{S}}'_{\text{smooth}}, T)$ 
7:   return  $T'$ 
8: end function

```

on the smoothing process in order to maintain the boundary structure of the face. We have found that eight iterations works well for our purposes. With non-rigid registration, deformation transfer and Laplace-Beltrami smoothing, we can describe the intrinsic normalization algorithm as in Algorithm 1. Here, \tilde{S} and \tilde{S}' are the aligned versions (with respect to \mathcal{T}) of neutral input face S and expressive input face S' , respectively. \mathcal{T}' , as before, is the deformed version of the template face, which is the normalized version of the input and the result of our algorithm.

4 Experiments and Results

4.1 Classification Improvements

Currently, the results published in [10] are considered to be state-of-the-art in action unit detection performance. The authors classify Center-Symmetric 3D Local Binary Patterns (CS-3DLBP) together with Ratio-Based Geometry features using the Random Forests classifier. We show here that utilizing the proposed intrinsic normalization framework as a pre-processing step improves the results of this ensemble beyond the current state of the art. To compare, we implemented the algorithm described in [10] and recreated the setup and protocol. We set up the system as a set of binary classifiers, with one for each AU. For classification, we used Random Forests with 2000 trees, 20 features per node, and a maximum tree depth of 5. We used 10-fold subject-aware cross-validation, meaning that while the contents of each partition were chosen nearly at random, no subject could appear in both training and test sets. Each partition typically contained between 250 and 350 examples. We generated a normalized version of the Bosphorus database by executing Algorithm 1 on all examples using a manually-constructed template mesh. However, the protocol described in

Action Unit	Positive example count	CS-3DLBP+RBG Not normalized	CS-3DLBP+RBG Normalized	% Improvement
AU1	215	5.63	3.90	30.7
AU2	147	3.81	2.04	46.3
AU4	176	4.75	1.66	65.0
AU5	133	11.73	9.04	23.0
AU6	100	5.58	4.90	12.1
AU7	477	5.57	4.33	22.2
AU9	107	1.69	0.88	48.1
AU10	72	4.93	4.23	14.2
AU11	14	24.30	15.53	36.1
AU12	295	1.13	1.11	1.9
AU14	76	6.04	5.75	4.9
AU15	68	4.24	1.88	55.6

Action Unit	Positive example count	CS-3DLBP+RBG Not normalized	CS-3DLBP+RBG Normalized	% Improvement
AU16	102	5.79	5.94	-2.6
AU17	134	4.89	2.73	44.2
AU18	185	1.94	1.58	18.4
AU20	51	2.94	0.97	66.8
AU23	63	6.72	6.78	-1.0
AU24	153	3.85	2.96	23.2
AU25	568	0.80	0.68	15.3
AU26	156	10.12	8.43	16.8
AU28	43	0.40	0.18	54.2
AU34	74	0.41	0.36	12.1
AU43	137	2.32	2.39	-3.0

Table 1: ROC AuC Error. Column 1: Number of positive examples. Column 2: Our implementation of [10] run on non-normalized input. Column 3: Our implementation of [10] run on normalized input. Column 4: Improvement due to normalization.

Marker ID	AU	Variance	Variance
		Non-normalized	Normalized
11	AU9	2.02e5	5.50e3
12	AU9	1.89e5	4.16e3
17	AU18	5.53e5	1.64e4
17	AU22	2.50e5	9.71e3
17	AU23	9.64e4	5.34e4
17	AU24	1.71e5	1.86e4
21	AU16	2.48e6	7.86e5
21	AU18	2.40e6	5.21e5
21	AU23	4.26e5	2.14e5
21	AU24	1.93e6	1.52e5
16	AU15	1.53e5	2.01e5
18	AU15	8.89e5	1.35e6

Table 2: Each entry in the two right columns represents the variance of Gaussian curvature measurements taken at the specified landmark, across several faces performing the specified action unit. Column 3: variances of measurements taken from non-normalized data. Column 4: variances of measurements taken from normalized data.

[14] accepts depth maps as input, while the output of the intrinsic normalization as described in Algorithm 1 is a triangulated mesh. To overcome this incongruity, meshes were converted back to depth maps by fitting a surface of the form $Z = F(X, Y)$. With this, we could now run this system on non-normalized and normalized Bosphorus data. Table 1 compares results of these trials.

It should be noted that the values we obtained for the non-normalized cases do not exactly match those published in [14]. There can be many reasons for this, from the difficulty of replicating the distribution of examples in cross-validation partitions, to potential differences in the implementation itself, such as numerical inaccuracies, and choice of supporting libraries.

Also, for some action units, classification performance drops for the normalized case. We suspect that this is due to suboptimal non-rigid registration near difficult areas such as the lips (AU16 and AU23) and the eyes (AU43). In any case, running the same code on both non-normalized and normalized datasets shows that in the vast majority of cases this procedure leads to more accurate prediction for the normalized case.

4.2 Variance Reduction

In addition to showing that intrinsic normalization improves classification performance, we analyzed the behavior of the data itself. This allowed us to draw conclusions about the technique independent of any classification algorithm. Our goal here in particular was to confirm that for measurements taken at a specific point across several faces undergoing the same action unit, the variance of these measurements is reduced by normalization. Lower measurement variance implies higher feature consistency, which in turn helps classification. To test our hypothesis, we chose several of the landmarks provided by the Bosphorus database. For each landmark, we chose one or more action units whose activation affects the region in which the landmark resides. Note that since all faces before and after normalization are in dense correspondence with one another, the landmarks can be easily located on the nor-

malized faces. Finally, we chose Gaussian curvature for the feature measurement because of its intrinsic nature. We thus measured the curvature at the landmark on all given faces, for normalized and non-normalized values. The results are shown in Table 2.

5 Conclusion

We have introduced a natural framework for normalizing values of functions over a 2D manifold. We have shown its effectiveness in terms of improvement of classification performance, as well as intra-class variance reduction. The framework is posed in the context of facial action unit detection. However, its application can be generalized to action recognition on any set of surfaces with manifold structure. The method works with water-tight surfaces as well.

Although not directly addressed in this paper, implementations of NRR and deformation transfer in [15] indicate that these algorithms can be implemented for real-time applications. Similarly, our method requires accurate and automatic landmark detection. For faces, [15] have shown that this is indeed achievable for 3D data. Finally, capture of a neutral face, while a manual process, is a trivial procedure performed only once per subject.

FACS coding has traditionally been performed by hand by qualified FACS analysts. The process is time consuming and prone to error. Accurate automated systems that operate on thousands of frames at a time could be used for anything from diagnosing or predicting mental disorders to richer human-computer interfaces. By defining a strong normalization procedure we therefore believe that we are providing a solid foundation for future automated FACS systems to depend on.

6 Acknowledgments

This research was supported by European Community's FP7-ERC program grant agreement no. 267414.

References

- [1] Neslihan Bayramoğlu, Guoying Zhao, and Matti Pietikainen. CS-3DLBP and Geometry-based Person Independent 3D Facial Action Unit Detection. In *International Conference on Biometrics*, pages 1–6, 2013.
- [2] Mario Botsch, Robert W Sumner, Mark Pauly, and Markus Gross. Deformation Transfer for Detail-Preserving Surface Editing. In *Vision, modeling and visualization*, pages 357–364, 2006.
- [3] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Levy. *Polygon Mesh Processing*. AK Peters, 2010. ISBN 978-1-56881-426-1.
- [4] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly, 2008.
- [5] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16(1):188–197, 2006.

-
- [6] Paul Ekman. Universal facial expressions of emotion. *California Mental Health Research Digest*, 8(4):151–158, 1970.
- [7] Paul Ekman and Wallace Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, 1978.
- [8] Y Huang, Y Wang, and T Tan. Combining statistics of geometrical and correlative features for 3d face recognition. In *Proceedings of the British Machine Vision Conference*, pages 879–888, 2006.
- [9] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. In *IEEE Transactions on pattern analysis and machine intelligence*, pages 971–987. IEEE, 2002.
- [10] P. Perakis, G. Passalis, T. Theoharis, and I. A. Kakadiaris. 3d facial landmark detection under large yaw and expression variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1552–1564, 2013.
- [11] Georgia Sandbach, Stefanos Zafeiriou, and Maja Pantic. Binary Pattern Analysis for 3D Facial Action Unit Detection. In *Proceedings of the British Machine Vision Conference*, pages 119.1–119.12. BMVA Press, 2012.
- [12] Arman Savran and Bulent Sankur. Automatic detection of facial actions from 3D data. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1993–2000. IEEE, September 2009.
- [13] Arman Savran, Neşe Alyüz, Hamdi Dibeklioglu, Oya Çeliktutan, Berk Gökberk, Bulent Sankur, and Lale Akarun. Bosphorus Database for 3D Face Analysis. In *Biometrics and Identity Management*, pages 47–56. 2008.
- [14] Yi Sun, Michael Reale, and Lijun Yin. Recognizing partial facial action units based on 3D dynamic range data for facial expression recognition. *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–8, September 2008.
- [15] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. Face / Off : Live Facial Puppetry. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.
- [16] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. *ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04*, pages 644–651, 2004.
- [17] Xi Zhao, Emmanuel Dellandréa, Liming Chen, and Dimitris Samaras. AU Recognition on 3D Faces Based On An Extended Statistical Facial Feature Model. In *Fourth IEEE International Conference on Biometrics*, pages 1–6, 2010.