# 3D Face Pose Tracking from Monocular Camera via Sparse Representation of Synthesized Faces

Ngoc-Trung Tran[1], Jacques Feldmar[2], Maurice Charbit[2]
Dijana Petrovska-Delacrétaz[1] and Gérard Chollet[2]
[1]*Telecom Sudparis, Paris, France*
[2]*Telecom ParisTech, Paris, France*

Keywords: Pose Tracking, Pose Estimation, Sparse Representation.

Abstract: This paper presents a new method to track head pose efficiently from monocular camera via sparse representation of synthesized faces. In our framework, the appearance model is trained using a database of synthesized face generated from the first video frame. The pose estimation is based on the similarity distance between the observations of landmarks and their reconstructions. The reconstruction is the texture extracted around the landmark, represented as a sparse linear combination of positive training samples after solving $\ell_1$-norm problem. The approach finds the position of new landmarks and face pose by minimizing an energy function as the sum of these distances while simultaneously constraining the shape by a 3D face. Our framework gives encouraging pose estimation results on the Boston University Face Tracking (BUFT) dataset.



## 1 INTRODUCTION

Head pose tracking is an important issue and has received much attention in the last decade because of the multiple applications involved such as video surveillance, human computer interface, biometrics, etc. The difficulties come from a number of factors such as projection, multi-source lighting as well as biological appearance variations, facial expressions and occlusion with accessories, e.g. glasses, hats... and especially the self-occlusion of the face appearance depending on head pose.

Since the pioneer work of (Cootes and Taylor, 1992; Cootes et al., 1998), it is well-known nowadays that the Active Shape Model (ASM) and Active Appearance Model (AAM) provided an efficient approach for face pose estimation and tracking frontal or near-frontal faces. Then some extensions (Xiao et al., 2004; Gross et al., 2006) have been developed. More recently, some works tackled local appearances changes by exhaustive local search around landmarks constrained by a 3D shape model, called deformable model fitting. This method can track single non-frontal face of large Pan angle well in well-controlled environment (Saragih et al., 2011). However, it requires a lot of training data to learn 3d shape and local appearance distributions. It is a limitation which makes them costly in unconstrained environments.

Another approach to track faces and estimate pose uses 3d rigid models such as semi-spherical or semi-cylindrical (Cascia et al., 2000; Xiao et al., 2003), ellipsoid (Morency et al., 2008) or mesh (Vacchetti et al., 2004). These methods do not need a lot training data and can estimate three rotations. However, these models assume constant distances between points and only rigid transformation can be applied on. This hypothesis is efficient when the face is far from camera and the image resolution is low. The low number of degree of freedom of these models facilitates the alignment process since there is not many parameters to optimize. The bias introduced by these strong constraints on the model can be restrictive particularly when the morphology of facial expressions are complicated to align.

For a robust tracking, head pose and facial actions should be taken into account. The early proposal (DeCarlo and Metaxas, 2000) to do so involves optical flow and and updates continuously during tracking to be adaptable to environmental changes. Optical flow can be very accurate but is not robust to fast movements. Moreover, this approach accumulates errors and drift away which is not easy to recover in long video sequences. With the help of local features which provides descriptors invariant to non-rigid motions, Chen and Davoine (Chen and Davoine, 2006) took advantages of local features constrained by a 3d-face paramerized model, called Candide-3, to capture both rigid and non-rigid head motions. This method does not need a huge-size pool of training data ei-


Tran N., Feldmar J., Charbit M., Petrovska-Delacrétaz D. and Chollet G..
3D Face Pose Tracking from Monocular Camera via Sparse Representation of Synthesized Faces.
DOI: 10.5220/0004345003280333
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2013), pages 328-333
ISBN: 978-989-8565-48-8

ther: it generates learning data from synthesized faces which is rendered from the first video frame. However, this approach suffers the following problem: the learning model assumes Gaussian distributions of local appearance and the candidate is chosen depends on the minimum distance to mean vector; hence, it is not realistic and possible to render more training faces to be robust to profile-view. Ybanez et al (Ybáñez-Zepeda et al., 2007) and Lefevre et al (Lefevre and Odobez, 2009) have adopted the same approach using synthesized faces as training data. (Ybáñez-Zepeda et al., 2007) finds the linear correlation between 3d model parameters and global appearance of stabilized face images. This method is robust for face and landmark tracking but limited to frontal and near-frontal faces. (Lefevre and Odobez, 2009) extended Candide by collecting more appearance information from head profiles by randomly choosing more points to represent facial appearance. Their error function consists of structure and appearance features combined with dynamic modeling. The minimization problem of this function is of large dimension and is likely to fall into local minimum.

In this paper, we adopt the sparse presentation which is well-known in many applications (Wright et al., 2009; Elad and Aharon, 2006; Mei and Ling, 2011), to build the tracking framework. We also take advantage of a synthesized database (Chen and Davoine, 2006; Ybáñez-Zepeda et al., 2007; Lefevre and Odobez, 2009) to circumvent the huge-size data problem and adopt local features to be robust to rigid and non-rigid changes. Why is sparse representation useful in our context? (1) the database of synthesized faces is not huge which is suitable to build dictionaries or codebooks. (2) The codebooks are able to be built by collecting not only positive samples but also negative samples. (3) The method could search and choose only the nearest neighbors from training data to represent the observation by solving $\ell_1$-norm problem (4) and the most important is that this method can reconstruct the observation using training samples that is likely a way to realize whether the observation is good or not; moreover, the noise is probably removed during reconstruction.

The remaining sections of this paper are organized as follow: Section 2 gives some background on the 3d face model and the sparse representation. Section 3 shows the proposed framework for tracking using sparse representation. Experimental results and analysis are presented in Section 4. Finally, we draw conclusions in Section 5.

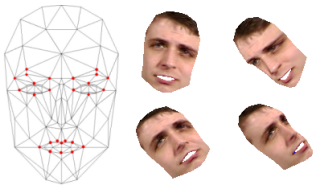

Figure 1: Candide-3 model and 26 selected points for tracking in our framework and some examples of rendered images from frontal face using Candide-3.

## 2 RELATED WORKS

In this work, Candide model (Ahlberg, 2001) is used to represent the 3d face model and create synthesized images as learning data and sparse representation (Wright et al., 2009) which models a new face as a sparse linear combination of learning faces.

### 2.1 3d Geometric Model

Candide-3 (Ahlberg, 2001) is a very commonly used face shape model. It consists of 113 vertices and 168 surfaces. See Fig. 1 represents the frontal view of the model. It is controlled both in translation, rotation, shape and animation:

$$g(\sigma,\alpha) = Rs(\bar{g} + S\sigma + A\alpha) + t \quad (1)$$

where $\bar{g}$ is 3N-dimensional mean shape (N = 113 is the number of vertices) containing the 3d coordinates of the vertices. The matrices S and A control respectively shape and animation through $\sigma$ and $\alpha$ parameters. And R is a rotation matrix, s is the scale, and t is the translation vector. The model makes a weak perspective assumption to project 3d face onto 2d image. Like in (Chen and Davoine, 2006; Lefevre and Odobez, 2009; Ybáñez-Zepeda et al., 2007), only 6 dimensions $r_a$ of the animation parameter are used to track eyebrows, eyes and lips. Therefore,the full model parameter is in our framework has 12 dimensions, consists of 3 dimensions of rotation $(r_x, r_y, r_z)$, 3 dimensions of translation $(t_x, t_y, t_z)$ and 6 dimensions of animation $r_a$:

$$b = [r_x, r_y, r_z, t_x, t_y, t_z, r_a] \quad (2)$$

**Texture Model:** In Candide model, appearance or texture parameters are not available. Usually, we warp and map the image texture onto the triangles of the 3d mesh by the image projection.

### 2.2 Sparse Representation

In many applications based on a linear model, we have to deal with a set of coefficients which almost

all of them are equal to zero. This is particularly evident when the problem is highly under-determined. In such cases we speak of sparsity representation. Sparsity representation has received a very large attention during the last two decades in many data processing applications (Tibshirani, 1996). Nowadays it is well-known that introducing a $\ell_1$-norm constraint on the optimization problem, based for example on the quadratic minimization, is able to force the sparsity of the solution.

More recently for classification problem in image processing, sparsity has been considered by (Wright et al., 2009). Given a set $A$ (training database) of $K$ different dictionaries, associated respectively to $K$ object classes, and let $y$ a vector under testing, we may expect that, if $y$ belongs to the class $i$, it could be explained by only the dictionary $A_i$. More specifically, if we write that $y \approx A\alpha$ that $\alpha$ must be sparse:

$$\hat{\alpha}_1 = \arg\min \|\alpha\|_1 \quad subject\,to \quad \|A\alpha - y\|_2^2 < \varepsilon \quad (3)$$

# 3 PROPOSED METHOD

Our framework consists of two steps: training and tracking. The proposed framework is basically similar to (Chen and Davoine, 2006) to create a database of synthesized faces but we propose a new way of tracking face poses. In this section, we describe our method in detail.

## 3.1 Training

The campaign of acquisition of ground-truth is very costly and the databases need often manual annotation which is also time consuming. To circumvent this drawback, many people used synthetic databases (Chen and Davoine, 2006; Ybáñez-Zepeda et al., 2007; Lefevre and Odobez, 2009) generated with Candide model. To collect training data, they do the three following steps to obtain images using Candide and building codebooks for the next tracking step:

### 3.1.1 3d Model Initialization

In the work of (Chen and Davoine, 2006), the authors align manually the Candide model on the first video frame $Y_0$ and warp and map the texture from the image to the model. In our work, we manually annotate several landmarks on the first video frame, then using the POSIT algorithm (Dementhon and Davis, 1995) we estimate the pose based on these landmarks and the corresponding Candide model landmarks to get the initial model parameters $b_0$.

### 3.1.2 Data Generation

After initialization, the texture is warped and mapped from the first video frame to the Candide model. We obtain our training database by rendering model different shapes and views around this frontal image. Let us remark that the full dimension of the parameters to track is 12. Therefore, we cannot explore this space finely. However,we can realize that the translation parameters $t_x$ and $t_y$ will not affect the face appearances and although the different expressions (corresponding to changes of animation parameters) can account non-rigid motions which generate different face appearance, it will not significantly influence local features. Therefore, only the rotation are gridded for building the training database. Specifically, 7 values of Pan and Tilt and Roll from -30 to +30 by step 10 are taken to create $7^3 = 343$ pose views as Fig. 1.

### 3.1.3 Codebook Building

In our framework, we take advantage of sparse representation like in (Wright et al., 2009) and it requires to build linear codebooks as discussed at Eq. 6 of Section 2.2. The framework adopts 26 local descriptors as $9 \times 9$ squared blocks that form 81-dim vectors around 26 landmarks as Fig. 1. Each codebook $A_i$ plays a role as learning data for landmark ith, it is a matrix $81 \times m$ where $m$ is the number of training samples. It consists of 343 positive training samples of ith landmarks extracted from the synthesized data and $m - 343$ negative training samples chosen randomly on first frame. The negative samples are very important to reduce noise during reconstruction. it means a tracked ith landmark, if it is good should approximately lie in the linear span of the training samples and, ideally, very few coefficients associated to positive landmarks should have non-zero values.

## 3.2 Tracking

Our tracking method refers to the Likelihood approach which searches the efficient distribution of $p(Y_t|b_t)$ where $Y_t$ is the observation of landmarks at time $t$ and $b_t$ is the hidden state, $b_t = (r_x, r_y, r_z, t_x, t_y, t_z, r_a)$ is the 12-dimensional vector in our context.

The tracking system starts from the frontal face that Candide is fitted on, and then it finds the candidate of face in the next frame $t$ from state vector at time $t-1$, with $t = 0$ at the first frame. In order to obtain the hidden state $b_t$ at time $t$, we initialize thirteen hidden states at time $t$ from previous state: $b_t = b_{t-1} + \delta_b$ to form a simplex, where $\delta_b$ is chosen randomly around previous state. The optimum

solution that can then be found, based on the simplex using a derivative-free optimizer such as downhill simplex (Nelder and Mead, 1965). For each parameter $b_t$, the 3d Candide is projected onto the next 2D frame at $t$ to localize 2D landmark positions, and the appearance texture $Y_t$ is concatenation of local textures $y_i(b_t), i = \{1,...,n\}$ which are extracted around the landmarks as the observed appearance. These observations can then be used to establish the observation model for tracking and the most important thing is how to find the efficient observation model.

In (Chen and Davoine, 2006; Lefevre and Odobez, 2009), the authors assumed that the local appearances around landmarks are independent and obey multivariate Gaussian distributions. So, the observation model is defined as a joint probability of Gaussian distributions and the tracking problem can be solved as maximum likelihood problem of a non-linear function.

$$p(Y_t|b_t) = \prod_{i=1}^{n} \varphi(y_i(b_t)|\mu_i, \Sigma_i) \tag{4}$$

where $Y_t = [y_1(b_t), y_2(b_t), ..., y_n(b_t)]$, n is the number of landmarks, $\varphi(y_i(b_t)|\mu_i, \Sigma_i)$ denotes multivariate Gaussian distribution of function value at observation around the ith landmark $y_i(b_t)$ with $\mu_i$ and $\Sigma_i$ prelearned from rendered images during training. Taking the logarithm of likelihood, they finally attempt to minimize the sum of Mahalanobis distance:

$$\hat{b}_t = \arg\min_{b_t} \sum_{i=1}^{n} \|y_i(b_t) - \mu_i\|^2_{\Sigma_i^{-1}} \tag{5}$$

The key of their proposition is that all points are assumed to be in an ellipsoid represented by a fixed mean and covariance and the best observation is the candidate which has the minimum distance to the mean. It is not really realistic.

In our work, we tackle the problem from another perspective with a similarity distance between the observation and its reconstruction from training samples defined as follows:

$$\hat{b}_t = \arg\min_{b_t} \sum_{i=1}^{n} \|y_i(b_t) - \hat{y}_i - \hat{\varepsilon}_i\|_2^2 \tag{6}$$

where $\hat{y}_i$ denotes the reconstruction of the observation $y_i(b_t)$ of ith landmark at time $t$ from training data and and $\hat{\varepsilon}_i$ is the noise should be removed from the observation. In order to obtain the reconstruction and noise, we attempt to minimize the problem:

$$\begin{aligned} \{\hat{\alpha}_i, \hat{\varepsilon}_i\} &= \arg\min_{\alpha_i, \varepsilon_i} \|\alpha_i\|_1 + \|\varepsilon_i\|_1 \\ \text{st.} \quad & A_i\alpha_i + \varepsilon_i = y_i \end{aligned} \tag{7}$$

where $A_i$ is the codebook of ith landmark. The reconstruction can be computed using $\hat{y}_i = A_i\rho(\hat{\alpha}_i)$ and the function $\rho(.)$ keeps only coefficients associates to the positive ith landmarks in its codebook and others are set zeros, see (Wright et al., 2009). The equation (7) could be converted to a basic $\ell_1$-norm problem of coefficient vector $[\hat{\alpha}_i\ \hat{\varepsilon}_i]$ and the codebook $[A_i\ I]$:

$$\begin{aligned} \{[\hat{\alpha}_i\ \hat{\varepsilon}_i]^T\} &= \arg\min_{\alpha_i, \varepsilon_i} \left\| [\alpha_i\ \varepsilon_i]^T \right\|_1 \\ \text{st.} \quad & [A_i\ I]\left[\alpha_i\ \varepsilon_i\right]^T = y_i \end{aligned} \tag{8}$$

where $I$ is identity matrix and (8) can be solved using (3). Equation (7) means that a new ith observed patch $y_i$ on the frame should lie on span of linear combination of atoms of $A_i$ codebook with some noise. If $y_i$ is a well-localized ith patch, the vector $\alpha_i$ should be sparse and very few coefficients associated to positive atoms of $A_i$ should be non-zero and the opposite for negative atoms. This sparse vector brings $\rho(\hat{\alpha}_i) \approx \hat{\alpha}_i$ and the error reconstruction will be small. To sum up, the better well-localized the landmark is, the smaller the objective function gets. From another point of view, for example in (Chen and Davoine, 2006; Lefevre and Odobez, 2009), the contribution of all training samples is the same to find the best candidate, but there are many landmarks not actually related to current observation that can cause noise. It is better to choose only the nearest neighbors around the observation to contribute to the objective function and our proposed function somehow is a quite reasonable selection. It is illustrated in Fig. 2: the large coefficients are associated to positive landmarks (red) and the noise is reduced in reconstruction, whereas the values of coefficients are distributed on both the positive and negative sides in the case of negative ones (yellow). Finally, this method makes no assumption of Gaussian distribution of landmarks appearance as in (Chen and Davoine, 2006; Lefevre and Odobez, 2009). In optimization context, the error function in (6) with constraints (7) is a multi-dimensional function of model parameter $b_t$ which can be solved using derivative-free optimizer as discussed above.

# 4 EXPERIMENTAL RESULTS

In order to evaluate the performance of our approach, we used the Boston University Face Tracking (BUFT) database(Cascia et al., 2000). This dataset contains two subsets, uniform-light and varying-light, where the ground-truth is captured by magnetic sensors "*Flock and Birds*" with an accuracy of less than $1^o$. The uniform-light set has a total of 45 video sequences (320×240 resolution) for 5 subjects (9 videos

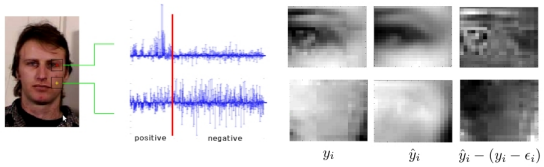


Figure 2: A visualization of observation, reconstruction and the error reconstruction of positive and negative landmarks. Large coefficients are associated to the positive side for positive landmarks, whereas this is the opposite for negative landmarks.

per subject) and the second set of video sequences for each of three subjects, they both have available ground-truth formatted as (*X_pos, Y_pos, depth, roll, yaw (or pan), pitch (or tilt)*). In this paper, we use the first set to evaluate our methods.

For each frame of a video sequence, we use the estimation of the rotation error $e_i = [\theta_i - \hat{\theta}_i]^T[\theta_i - \hat{\theta}_i]$ like in (Lefevre and Odobez, 2009) to evaluate the accuracy and robustness, where $\theta_i$ and $\hat{\theta}_i$ are (pan,tilt,roll) of the ground-truth and estimated pose at frame $i$ respectively. The robustness is the number $N_s$ of frames tracked successfully and $P_s$ is the percentage of frames tracked over all videos. A frame is lost when $e_i$ exceeds the threshold. The precision includes the pan, tilt, roll and average rotation errors (MAE measure) as the measure of tracker accuracy over tracked frames: $E_{pan}, E_{tilt}, E_{roll}$ and $E_m = \frac{1}{3}(E_{pan} + E_{tilt} + E_{roll})$ where $E_{pan} = \frac{1}{N_s}\sum_{i \in S_s}(\theta^i_{pan} - \hat{\theta}^i_{pan})$ (similarly for the tilt and roll) and $S_s$ is set of tracked frames.

In our framework, we used the synthesized database as discussed above, with 26 landmarks chosen around the eyes, nose and mouth to build codebooks. For ith landmark, the codebook $A_i \in R^{343\times1200}$ where the number of columns is 1200 which includes 343 positive samples and the remaining negative samples are chosen randomly. In order to solve the $\ell_1$-norm problem, we used the fast and efficient algorithm described in (Yang et al., 2010).

For evaluation, we evaluate our performance with framework of Chen and Davoine (Chen and Davoine, 2006) with and without using PCA to reduce the dimension of features before computing Mahalanobis distances and compare to state-of-the-art methods. We also evaluate the state-of-the-art of landmark tracking (Saragih et al., 2011) and compare to our work. As can be seen in Table 1, the Chen and Davoine's framework (Chen and Davoine, 2006) is slightly worse than the second model using PCA for feature reduction. Our proposed approach is better than the two others both in terms of precision and robustness because we took into account the error of observation and found contributions only from the nearest atoms as discussed above. Comparing to state-of-the-art methods, we outperform (Cascia et al., 2000)

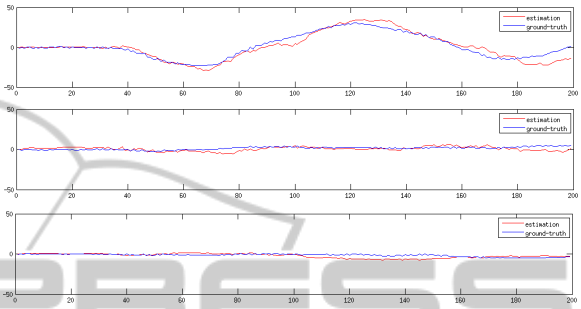


Figure 3: One example result on (jim1.avi) between our estimation and ground-truth. From first to third rows: Pan error: 3.99, Tilt error: 2.32 and Roll error: 2.02.

at both the accuracy and robustness, slightly worse than (Saragih et al., 2011) at robustness and finally lower than remaining methods at both the robustness and accuracy. The result shows that our method is still not robust and gets a high error of Pan rotation when the failure of tracker is caused by large Pan angle. For example, several landmarks around one eye disappear when the Pan is more than $30^o$ and this leads to uncertainty of observations around landmarks and makes the tracker failed, see Fig. 3. And this is also the reason causes the failure of robustness during tracking in our framework. However, this problem could be solved by generating more training data, using different weights of landmarks or by making the observation model adaptive to changes of the environment.

## 5 CONCLUSIONS

In this paper, we propose a new way to deal with the problem of face tracking using sparse representation. In our method, we synthesize a database and local features are extracted around landmarks to build codebooks. For tracking, an energy function which is the sum of similarity distances between the observations and their reconstructions using a sparse representation, is minimized. The result shows that the use of a sparse representation is better than the use of mean and covariance matrices to describe the observation model. It suggests that mean and covariance matrices in the same framework are inadequate to model the

Table 1: The comparison of robustness $P_s$ and accuracy ($E_{pan}$, $E_{tilt}$, $E_{roll}$, $E_{avg}$) between our method and state-of-the-art on uniform-light set of BUFT dataset.

| Approach | $P_s$ | $E_{pan}$ | $E_{tilt}$ | $E_{roll}$ | $E_{avg}$ |
|---|---|---|---|---|---|
| (Cascia et al., 2000) | 75% | 5.3 | 5.6 | 3.8 | 3.9 |
| (Xiao et al., 2003) | 100% | 3.8 | 3.2 | 1.4 | 2.8 |
| (Lefevre and Odobez, 2009) | 100% | 4.4 | 3.3 | 2.0 | 3.2 |
| (Morency et al., 2008) | 100% | 5.0 | 3.7 | 2.9 | 3.9 |
| (Saragih et al., 2011) | 92% | 3.9 | 3.9 | 2.3 | 3.4 |
| Mahalanobis Distance (Chen and Davoine, 2006) | 85% | 5.1 | 3.8 | 2.0 | 3.6 |
| PCA + Mahalanobis Distance | 87% | 5.2 | 3.5 | 2.0 | 3.6 |
| **Our model** | **89%** | **4.7** | **3.5** | **2.1** | **3.4** |

variations of appearance around landmarks. Although performance is improved by our method, it remains quite far from state-of-the-art methods. However, it could be more efficient if the dynamical model or state evolution was taken into account. Or the weights of contribution to energy function were dependent on the confidence of landmark observations at each time. Finally the observation model could also be adapted to changes through frames and be made more robust for face tracking and pose estimation.

## ACKNOWLEDGEMENTS

This work was financially supported by the ANR-CONTINT-ORIGAMI2 project (ANR-10-CORD-0016) and the LTCI of the Telecom-ParisTech Institute, France.

## REFERENCES

Ahlberg, J. (2001). Candide-3 - an updated parameterised face. Technical report, Dept. of Electrical Engineering, Linkping University, Sweden.

Cascia, M. L., Sclaroff, S., and Athitsos, V. (2000). Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Trans. PAMI*, 22(4):322–336.

Chen, Y. and Davoine, F. (2006). Simultaneous tracking of rigid head motion and non-rigid facial animation by analyzing local features statistically. In *BMVC*.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. *TPAMI*, pages 484–498.

Cootes, T. F. and Taylor, C. J. (1992). Cj.taylor, "active shape models - "smart snakes. In *BMVC*.

DeCarlo, D. and Metaxas, D. N. (2000). Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99–127.

Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *IJCV*, 15:123–141.

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745.

Gross, R., Matthews, I., and Baker, S. (2006). Active appearance models with occlusion. *IVC*, 24(6):593–604.

Lefevre, S. and Odobez, J.-M. (2009). Structure and appearance features for robust 3d facial actions tracking. In *ICME*.

Mei, X. and Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans. PAMI*, 33(11):2259–2272.

Morency, L.-P., Whitehill, J., and Movellan, J. R. (2008). Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation. In *FG*.

Nelder, J. A. and Mead, R. (1965). A simplex algorithm for function minimization. *Computer Journal*, pages 308–313.

Saragih, J. M., Lucey, S., and Cohn, J. F. (2011). Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91:200–215.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288.

Vacchetti, L., Lepetit, V., and Fua, P. (2004). Stable real-time 3d tracking using online and offline information. *IEEE Trans. PAMI*, 26(10):1385–1391.

Wright, J., Yang, A., Ganesh, A., Sastry, S., and Ma, Y. (2009). Robust face recognition via sparse representation. *TPAMI*, 31(2):210 –227.

Xiao, J., Baker, S., Matthews, I., and Kanade, T. (2004). Real-time combined 2d+3d active appearance models. In *CVPR*.

Xiao, J., Moriyama, T., Kanade, T., and Cohn, J. (2003). Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13:85 – 94.

Yang, A. Y., Ganesh, A., Zhou, Z., Sastry, S., and Ma, Y. (2010). A review of fast l1-minimization algorithms for robust face recognition. *CoRR*, abs/1007.3753.

Ybáñez-Zepeda, J. A., Davoine, F., and Charbit, M. (2007). Local or global 3d face and facial feature tracker. In *ICIP*, volume 1, pages 505–508.