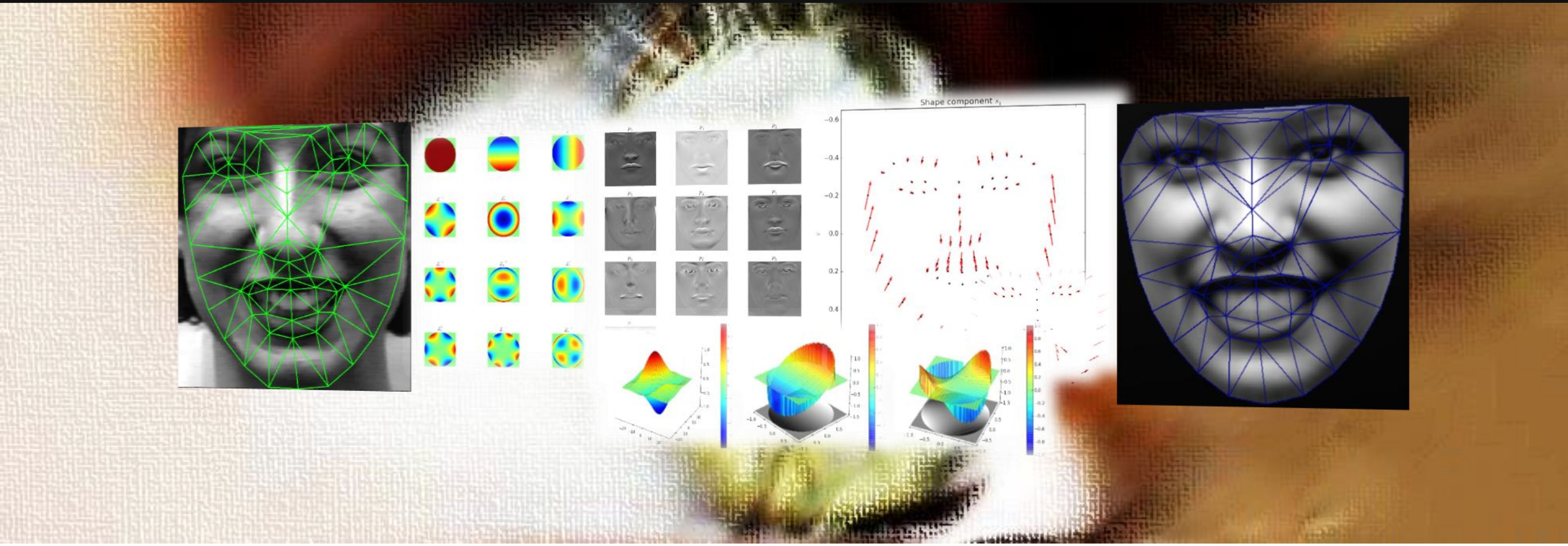
Statistical models for face analysis

4th Stochastic geometry days, August 2015



Pascal Bourdon
XLIM Laboratory (UMR CNRS 7252) / Université de Poitiers
pascal.bourdon@univ-poitiers.fr

Summary

- Introduction
- Early works
- A deformable face model
- Model fitting and data representation
- Conclusion

Summary

Introduction

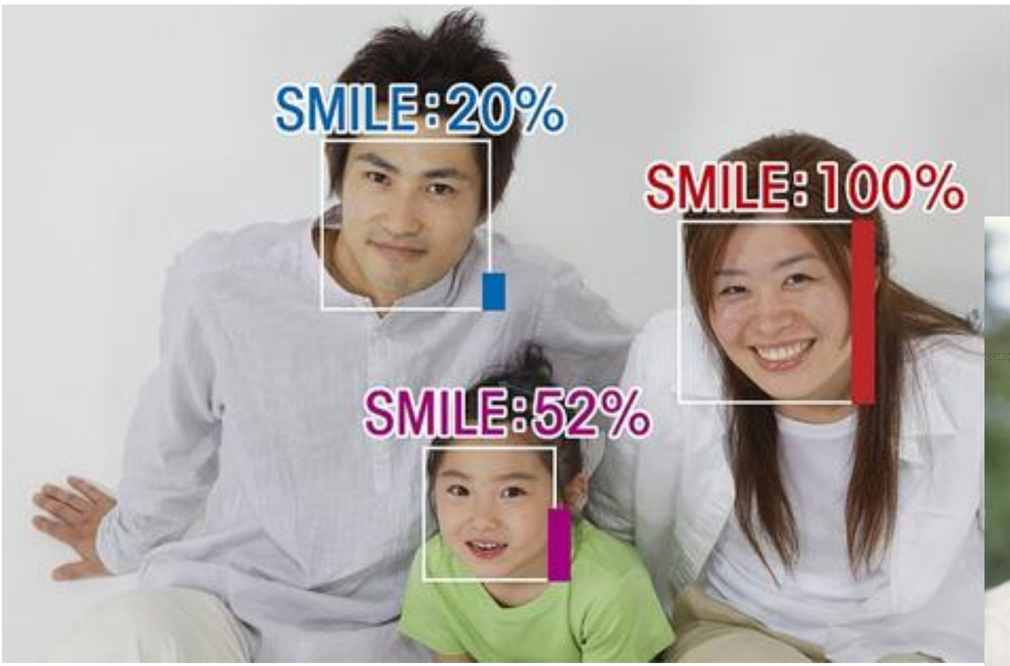
Why face analysis?

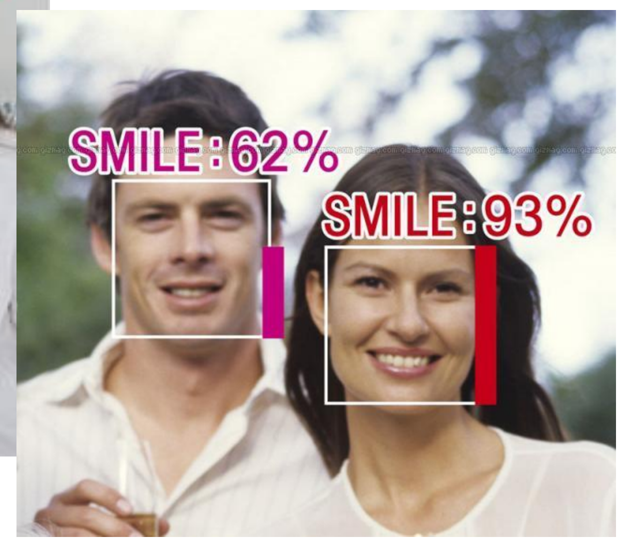
Introduction

Why face analysis?

Introduction

Why face analysis?





Introduction

Why face analysis?

- Identity recognition
- Morphing/VFX
- Aging simulation
- Fatigue monitoring (road safety)
- Behavior analysis (reactions, emotions…)
- Virtual mirror (online shopping: makeup, glasses…)
- Visio-conference, videos games…

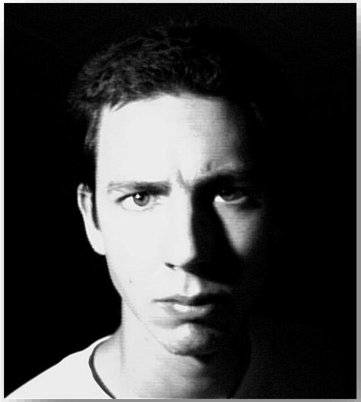
Introduction

A challenging task…

Pose

Expression

Illumination



Resolution

Summary

- Introduction
- Early works
- A deformable face model
- Model fitting and data representation
- Conclusion

Early works: object tracking/image alignment

Matching a reference image (template)

Object tracking associates image objects in consecutive video frames

$$I(x', y', t + \partial t) = I(x, y, t) \quad (x, y) \in \Omega$$

Early works: object tracking/image alignment

Matching a reference image (template)

Image alignment associates target objects from a reference

$$I(x', y') = T(x, y) \quad (x, y) \in \Omega$$

Early works: object tracking/image alignment

Both problems are traditionally solved using optical flow estimation

- Common assumption: flow is constant within the image object $I(x,y)$

$$I(x,y,t) = I(x+Vx, y+Vy, t+\partial t) \qquad \mathbf{\upsilon} = \begin{bmatrix} Vx \\ Vy \end{bmatrix}$$

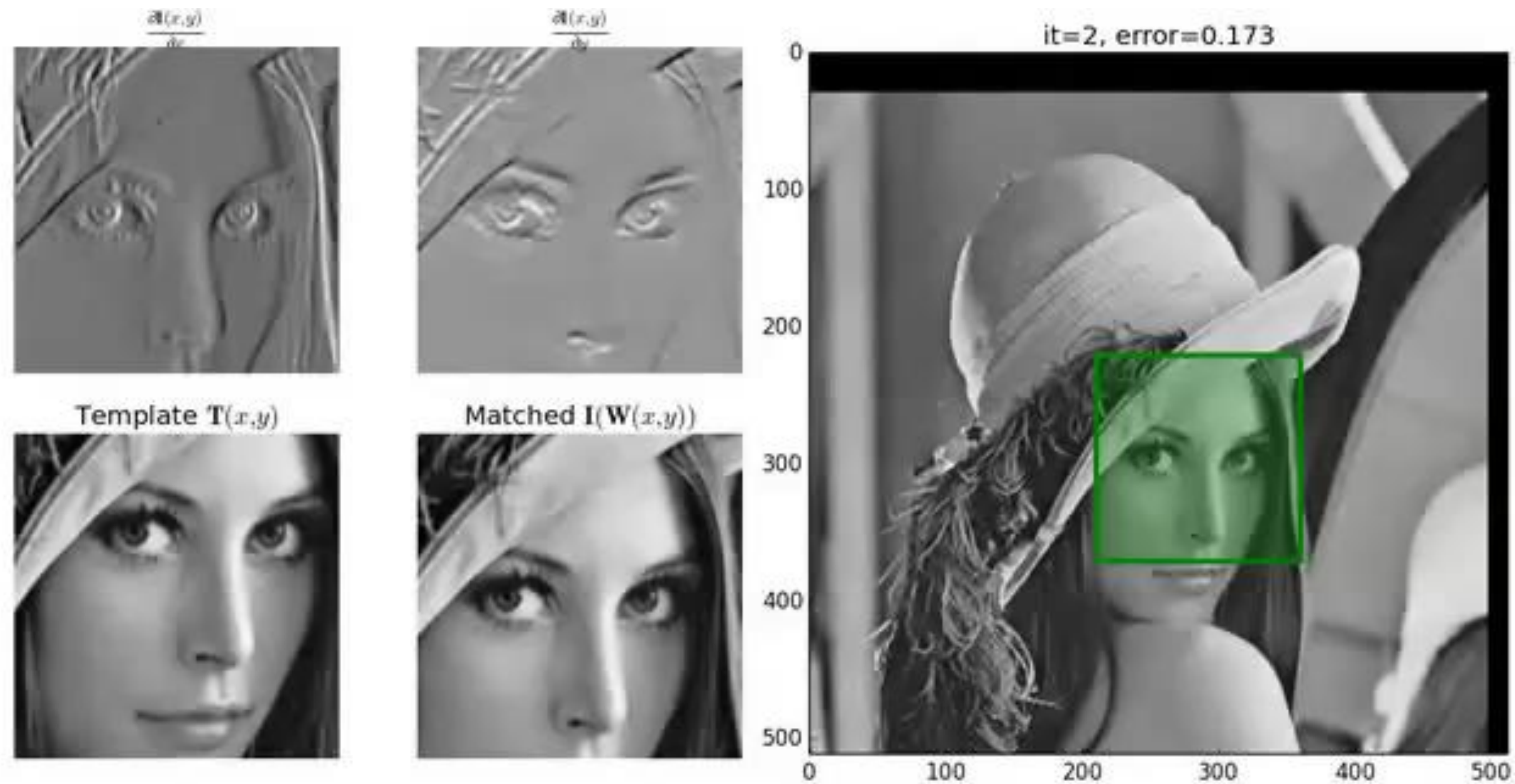
$$\Rightarrow -I_t(x,y,t) \approx \nabla I^T(x,y,t)\mathbf{\upsilon}$$

- Example: least mean squares method (Lucas-Kanade)

$$\xi(\mathbf{\upsilon}) = \sum_{(x,y)\in\Omega} \left(-I_t(x,y,t) - \nabla I^T(x,y,t)\mathbf{\upsilon}\right)^2 \qquad \frac{\partial \xi(\hat{\mathbf{\upsilon}})}{\partial \mathbf{\upsilon}} = 0$$

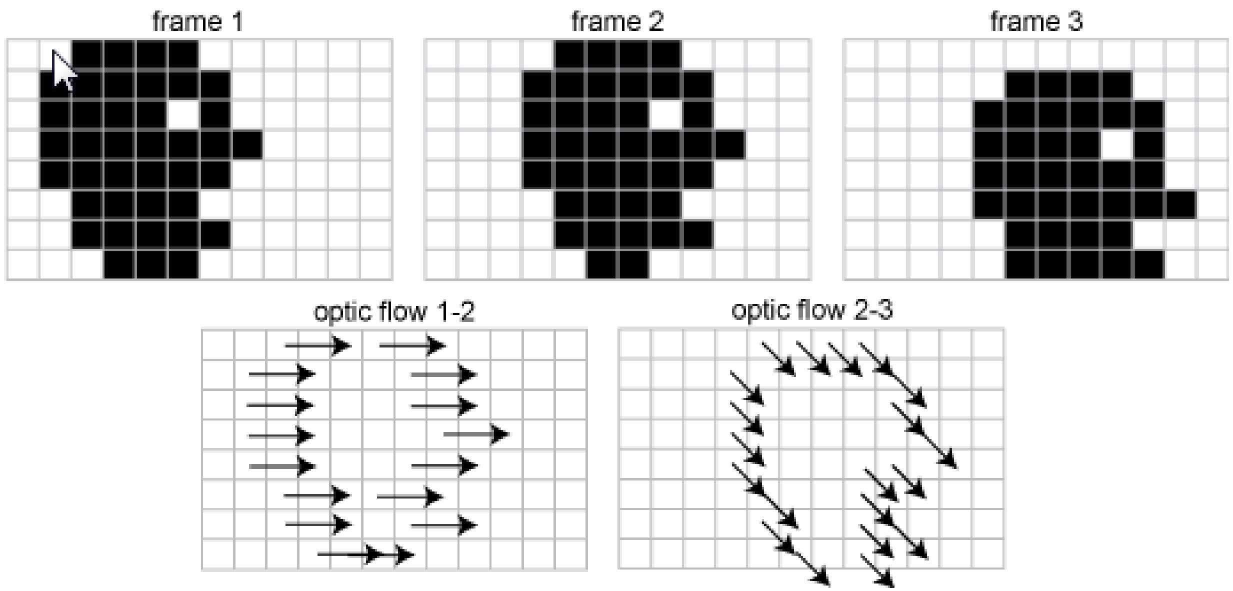
Early works: object tracking/image alignment

Lucas-Kanade tracker



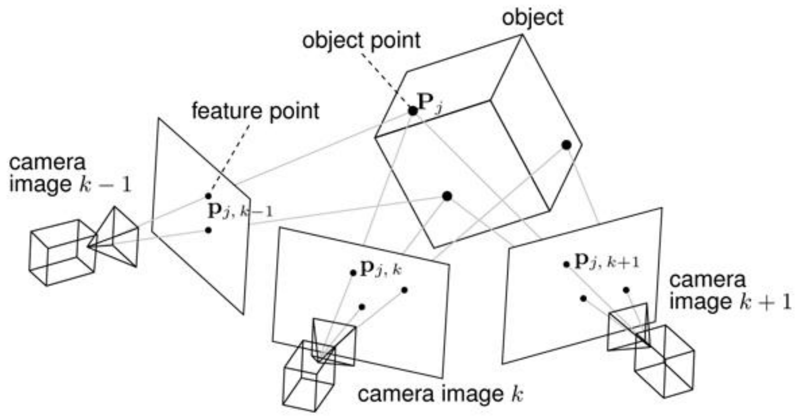
Early works: object tracking/image alignment

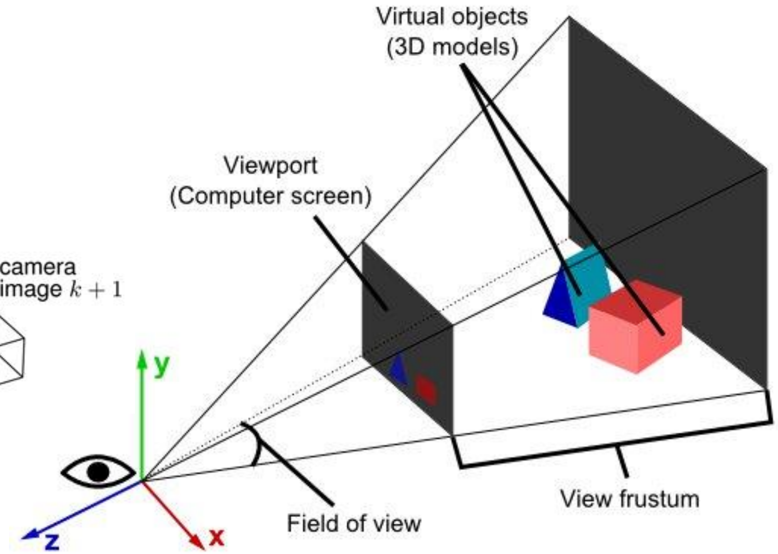
A constant flow within the image object?



Early works: object tracking/image alignment

A constant flow within the image object?

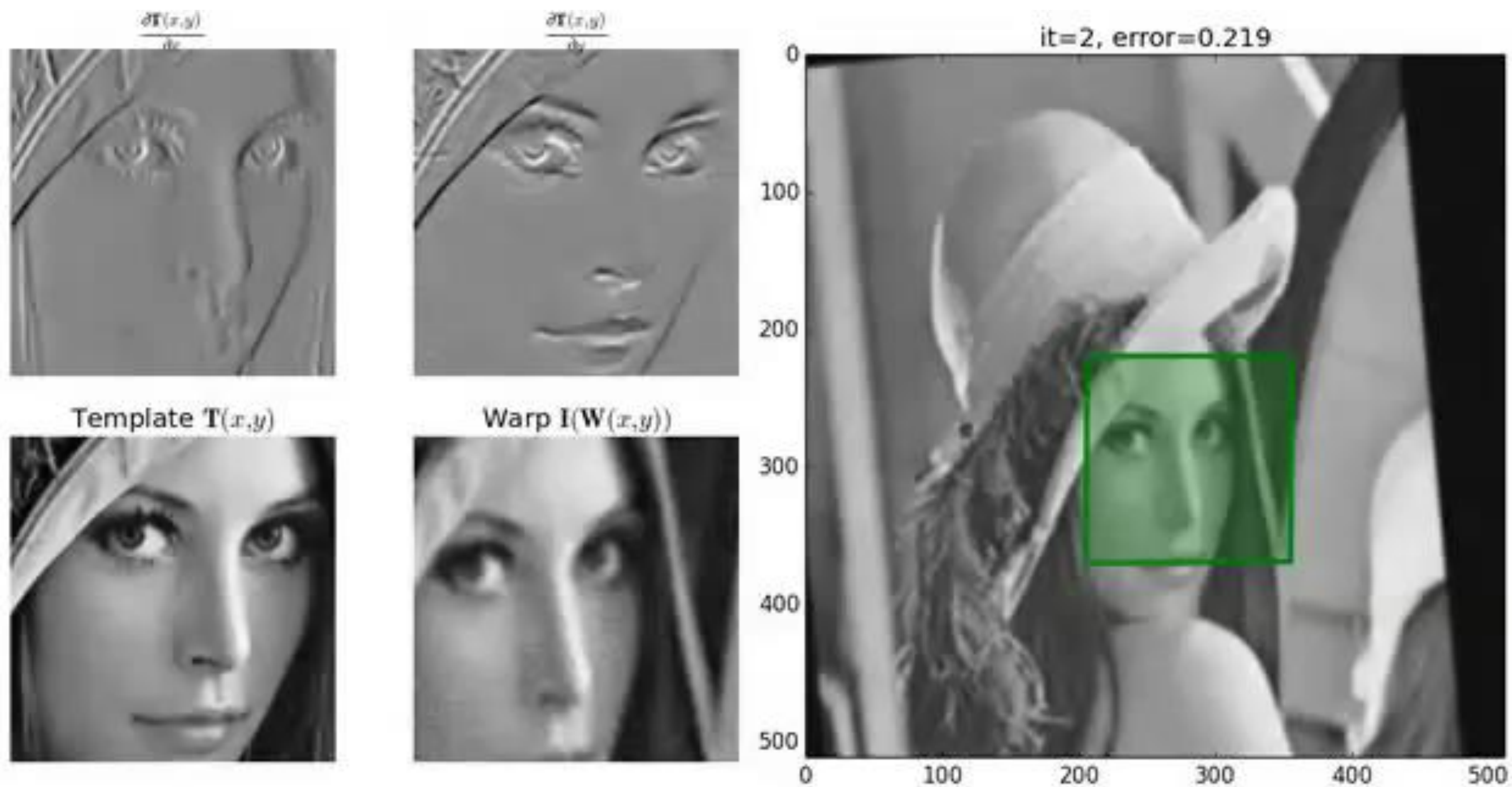




$$\xi(\mathbf{W}) = \sum_{(x,y)\in\Omega} \left(\underbrace{T(x,y)}_{template} - I\left(\underbrace{\mathbf{W}(x,y)}_{warp} \right) \right)^2 \qquad \mathbf{W}(x,y) = \begin{bmatrix} s_x x + w_z y + t_x \\ -w_z x + s_y y + t_y \end{bmatrix}$$

Early works: object tracking/image alignment

Matthews-Baker ICIA tracker



Early works: object tracking/image alignment

A constant flow within the image object?

Rigid objects : 3D-to-2D projections
Non-rigid objects : a deformable model is needed!

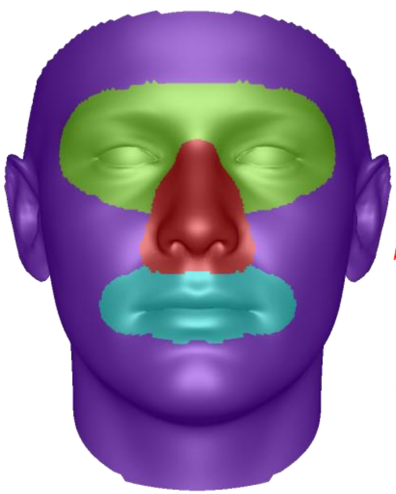
Summary

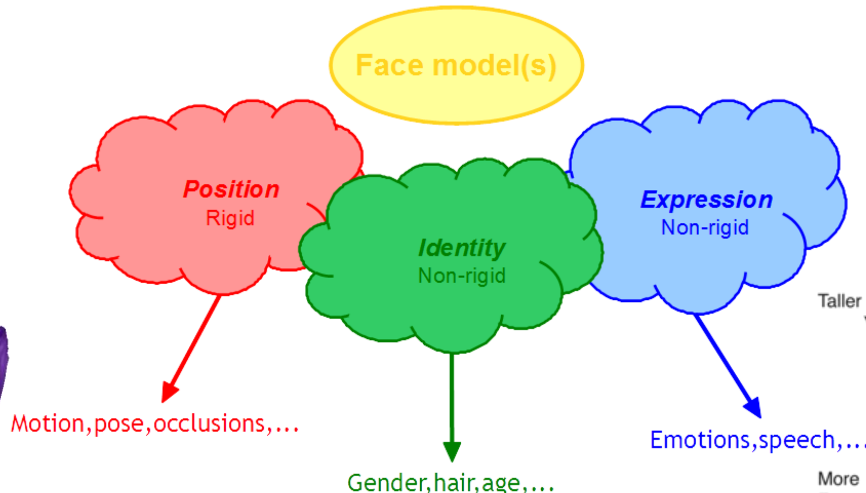
- Introduction
- Early works
- A deformable face model
- Model fitting and data representation
- Conclusion

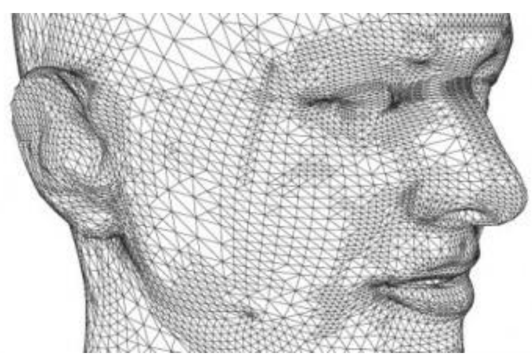
A deformable face model

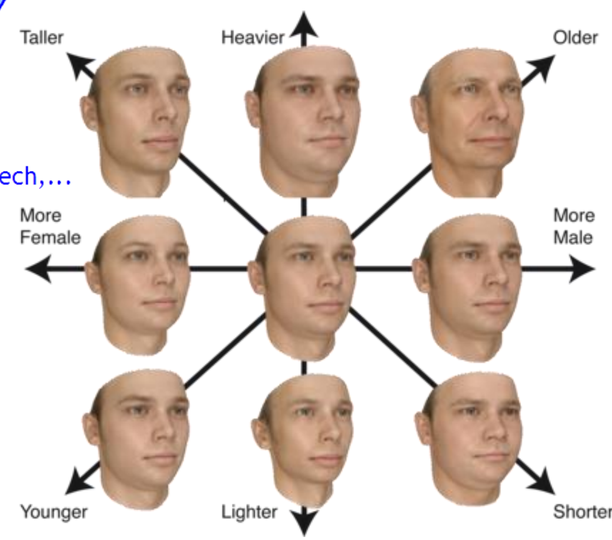
From rigid to deformable models

We want to enhance a previously static alignment model (i.e. template image) to take into account changes of identity, expression or luminosity. In computer vision, these changes are seen as local variations of shape and/or texture





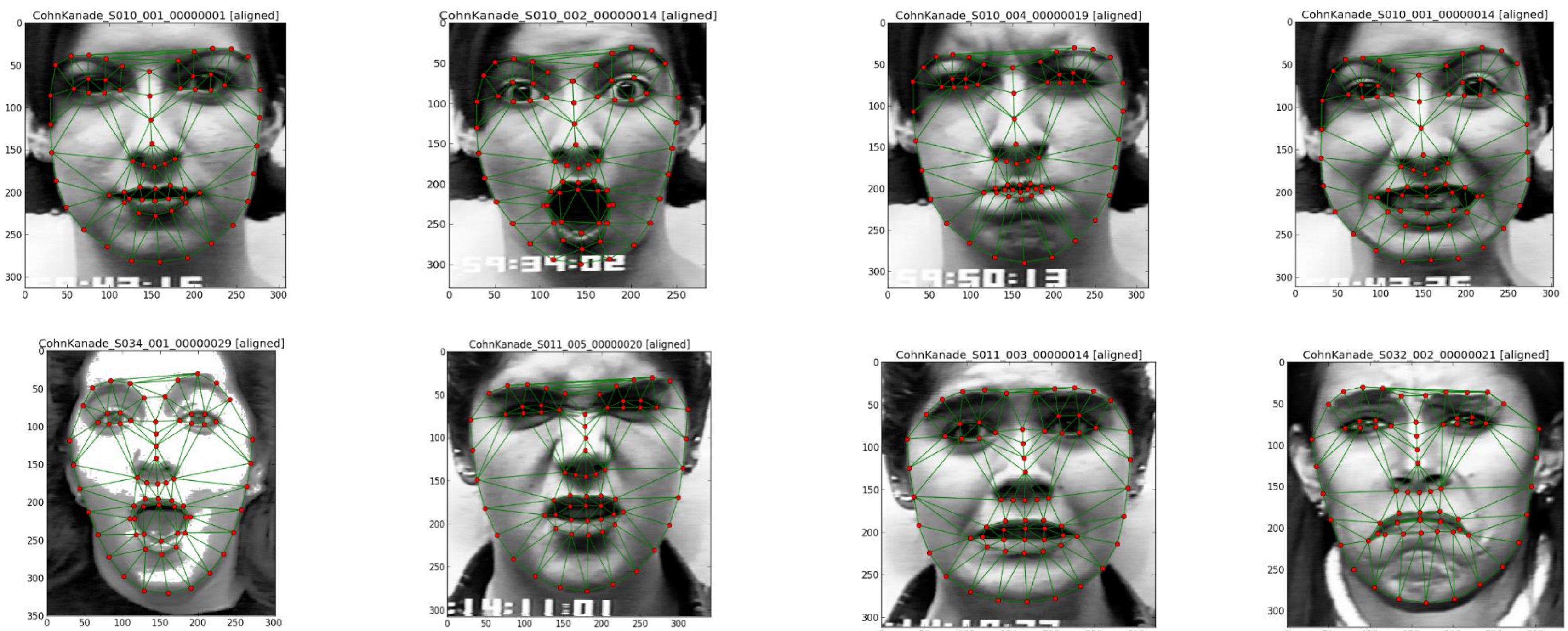




A deformable face model

Active Appearance Models (Cootes et al.)

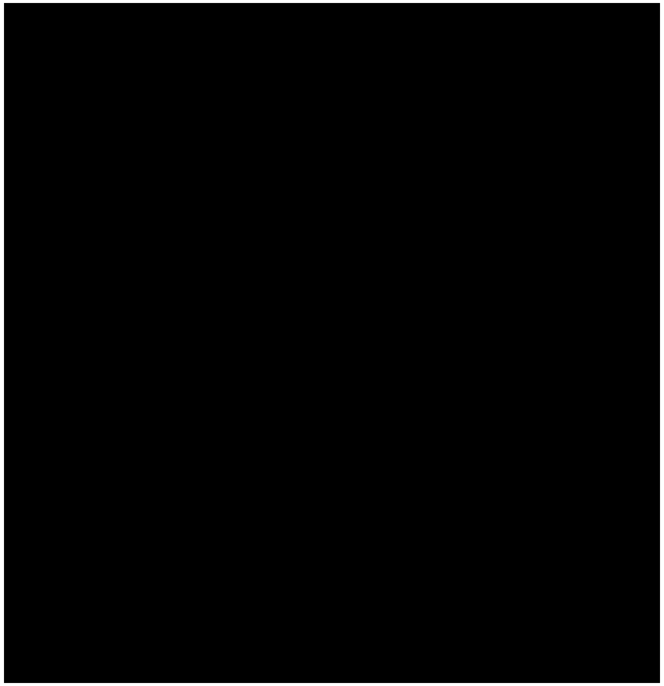
are able to jointly synthesize an image (texture) and the shape it relates to. An AAM model is built through principal component analysis (PCA) of aligned texture and shape data from a (manually annotated) image database



A deformable face model

Active Appearance Models (Cootes et al.)

are able to jointly synthesize an image (texture) and the shape it relates to. An AAM model is built through principal component analysis (PCA) of ***aligned*** *texture and* ***shape data*** *from a (manually annotated) image database*



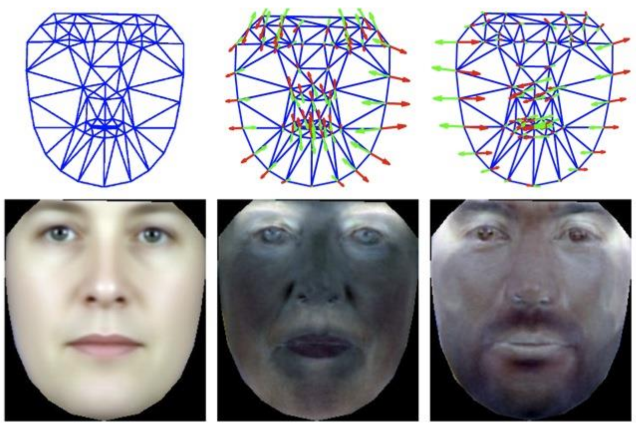
A deformable face model

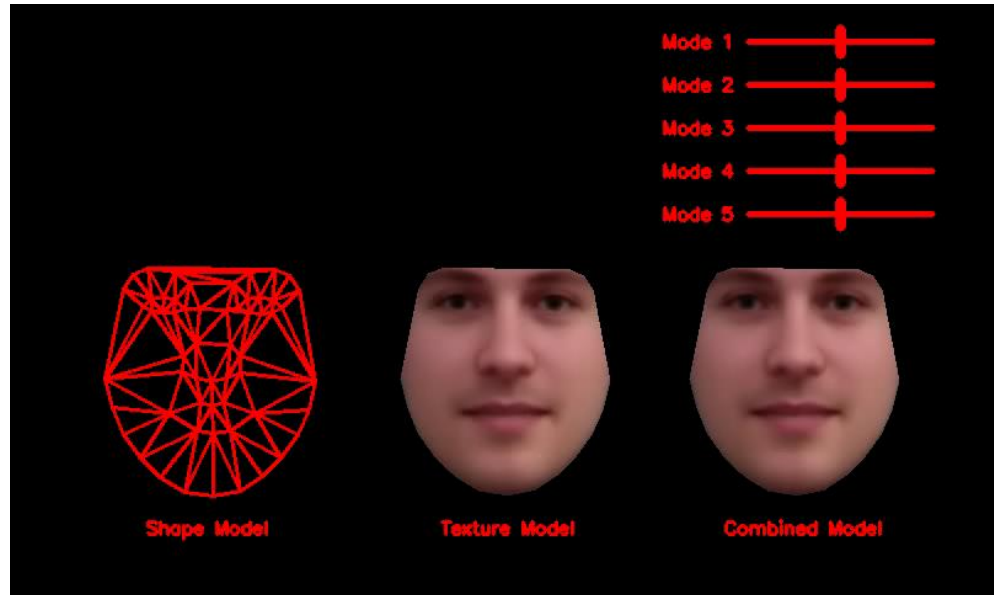
Active Appearance Models (Cootes et al.)

are able to jointly synthesize an image (texture) and the shape it relates to. An AAM model is built through principal component analysis (PCA) of ***aligned texture*** *and shape* ***data*** *from a (manually annotated) image database*

A deformable face model

Face synthesis from an AAM model





Shape model: $\mathbf{s} = \mathbf{s}_0 + \sum_{p=1}^{P_s} \Phi_p \mathbf{s}_p$ *Texture model:* $\mathbf{A} = \mathbf{A}_0 + \sum_{p=1}^{P_g} \Psi_p \mathbf{A}_p$

Summary

- Introduction
- Early works
- A deformable face model
- Model fitting and data representation
- Conclusion

Model fitting and data representation

Fitting a model to new data

- Original AAM (Cootes) : generating shape samples from the database with random noise and learning a regression model which minimizes the error between observed and synthesized textures

- Gradient-descent method (Matthews-Baker) : a least mean squares-based approach inspired by Lucas-Kanade, where AAM parameters are estimated together with global transform parameters

$$\xi(\mathbf{\Psi},\mathbf{p},\mathbf{\Phi}) = \sum_{(x,y)\in\Omega}\left(\underbrace{A(x,y;\mathbf{\Psi})}_{template} - I\left(\underbrace{\mathbf{W}(x,y;\mathbf{p},\mathbf{\Phi})}_{warp}\right)\right)^2$$

$(\mathbf{\Psi},\mathbf{\Phi})$ *: AAM parameters* $\quad$ $\mathbf{p}$ *: global (e.g. affine) transform parameters*

$\mathbf{W}$ *: image warping function, computed from all geometry-related parameters*

Model fitting and data representation

Fitting a model to new data

Model fitting and data representation

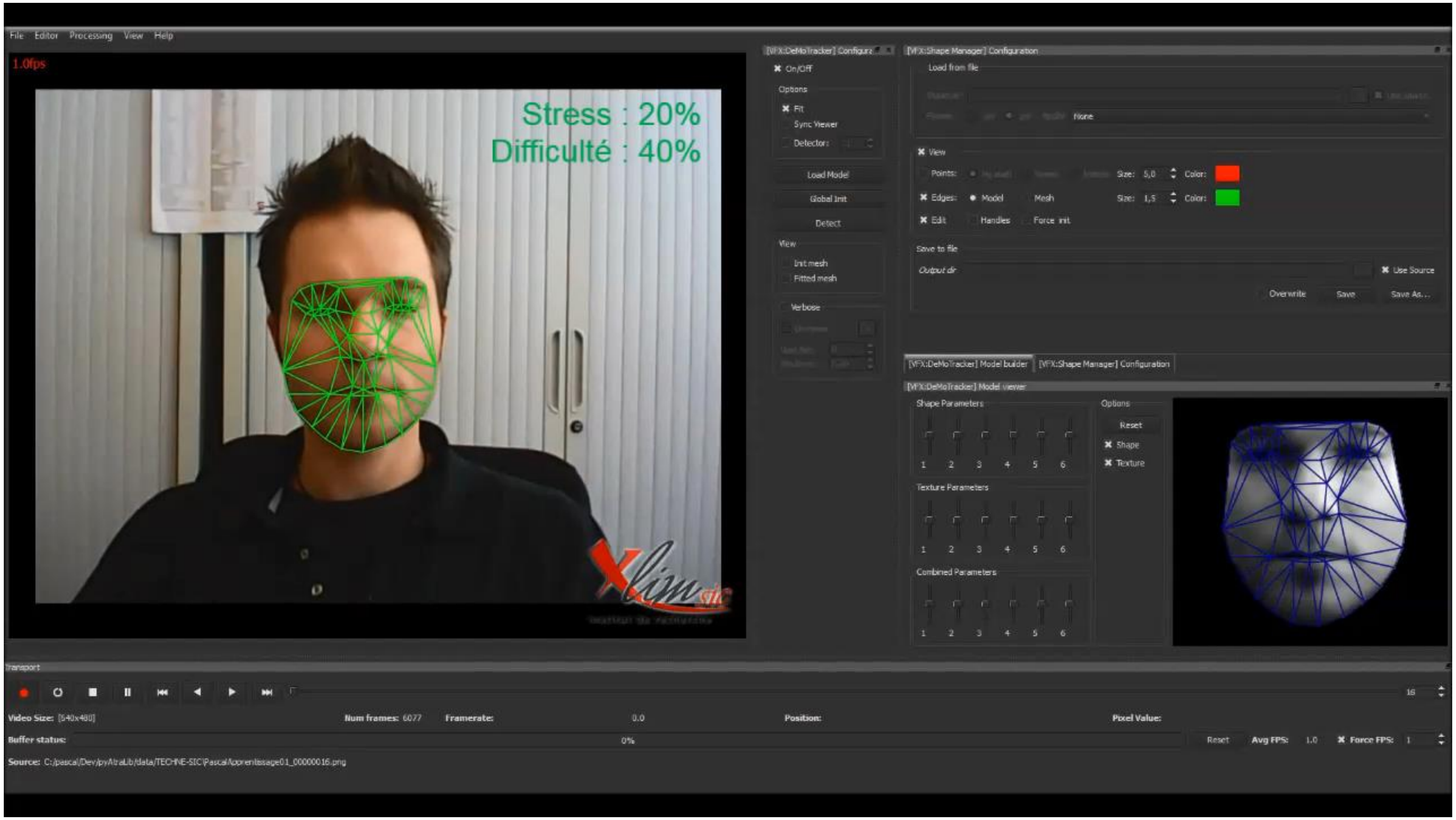
Further data analysis from estimated AAM parameters

Motion capture and animation

Model fitting and data representation

Further data analysis from estimated AAM parameters

Emotion analysis (e.g. online learning)



Summary

- Introduction
- Early works
- A deformable face model
- Model fitting and data representation
- Conclusion

Conclusion

- A statistical face model learned from observed data
- Compacts relevant information into just a few parameters
- Applications in the field of information retrieval
- Current work : polynomial representation of texture model and applications in animation and e-learning
- Next : force semantic information into PCA analysis i.e. individual muscle activity parameters (local constraints)

Applications in alignment, tracking and landmarking

Thank you ! ☺