Face Recognition Based on Fitting a 3D Morphable Model

by Volker Blanz and Thomas Vetter

Presented by A. Brian Davis

What lies ahead

- Introduction
- 3D Morphable Model
- Face Vectors
- Optical Flow
- Using Face Vectors
- Image Synthesis {shape / colors}
- Fitting the model
- Optimization
- Results

Introduction

- Face recognition
  - Intrinsic vs extrinsic parameters
    - Extrinsic: head pose, illumination
    - Intrinsic: shape of face, texture
    - Get one set without the other?
      - Eigenlighting
    - Automatically extract?
- What is our 3D morphable model?

3D Morphable Model

- How to separate intrinsic from extrinsic? (calculate both)
  1) Hypothesize all parameters
  2) Synthesize face from parameters
  3) Record “reconstruction error” wrt pixels
  4) Minimize error (gradient descent)
- 3D model?
  - Estimate orientation of face
- Reconstruction vs Recognition
  - Change extrinsic, classify intrinsic

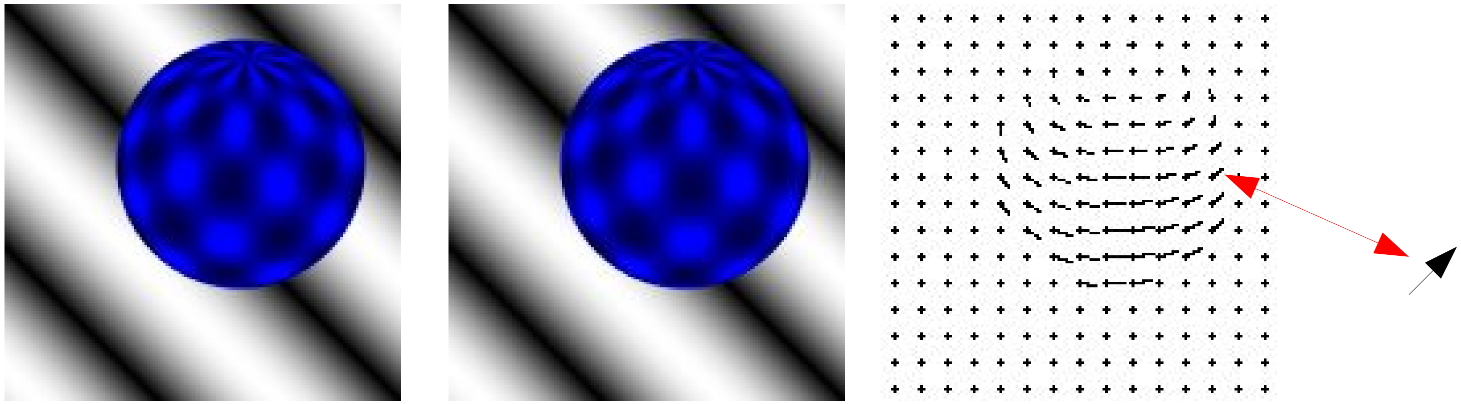
Face Vectors

- How do we get 3D?
  - Database 3D laser scans (100m,100f)
    - Race specific
  - 262144 points, radii, RGB
    - Preprocessing
      - Manually nix noise
      - Forehead trimming
      - Cut behind the ears
- Correspondence
  - Reference face
  - Points densely correspond – Optical flow

Optical Flow

- Assumptions
  - Constant objects moving between frames
  - Constant brightness wrt velocity
  - Different objects?
- Then, change in intensity equal to gradient * actual change (equal to zero)
- Calculate for each small neighborhood
  - Erratic, smoothing
  - Connected springs

Optical Flow cont.



http://www.cs.otago.ac.nz/gpxpriv/vision_optflow.html

http://www.societyofrobots.com/images/programming_computer_vision_optical_flow.gif

Face Vectors cont.

- Reference face has N vertices
  - One color for each vertice
- For each new 3D scan:
  - Calculate optical flow (invalid assumptions)
  - Save N points from new scan
    - Interpolate from optical flow result
- Have all our scans. Now what?

3D faces in usable form

- PCA on resulting vectors
  - We all saw this coming
    - Fewer parameters for reconstruction, recognition
- Treat radii, textures independently
- From pattern rec:
  - PCA de-correlates data
  - Assumed to be multi-variable Gaussian
  - For point p, $P(p) = \prod_{i=1}^{m} e^{-\frac{1}{2}(\frac{\alpha_i}{\sigma_i})^2} = e^{\sum_{i=1}^{m} -\frac{1}{2}(\frac{\alpha_i}{\sigma_i})^2}$
    - Useful calculating priors

Optimize the modelling

- Given:
  - Low number of 3D faces
  - Unbounded potential number faces to match
- Large variations between faces hard to model
  - Global least-squared sense
- Solution: Fit model to face globally and segments
  - Eyes, nose, mouth, surrounding area
    - Blend all to look good

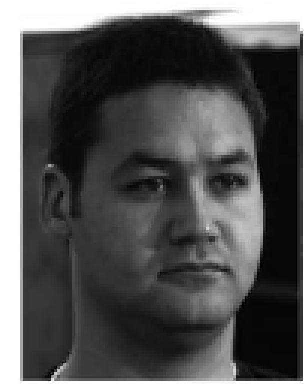
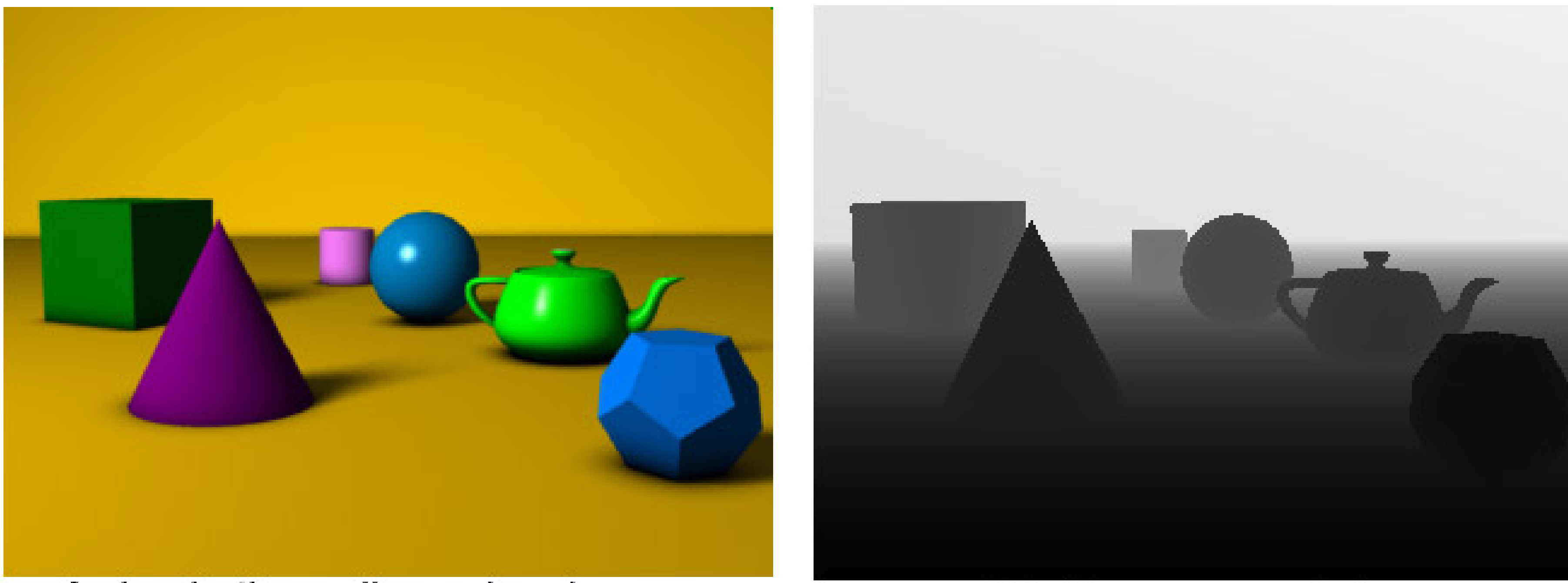


Image Synthesis

- How to create face image from parameters
  - Shape, texture coefficients, rotation of face, translation in picture
  - Apply 3D affine transform
  - Perspective projection – project onto plane as we perceive it
  - Occlusion / shadows?
    - Z-buffer
- What about colors?

Z-buffer



A simple three dimensional scene

Z-buffer representation

http://en.wikipedia.org/wiki/Image:Z-buffer.jpg

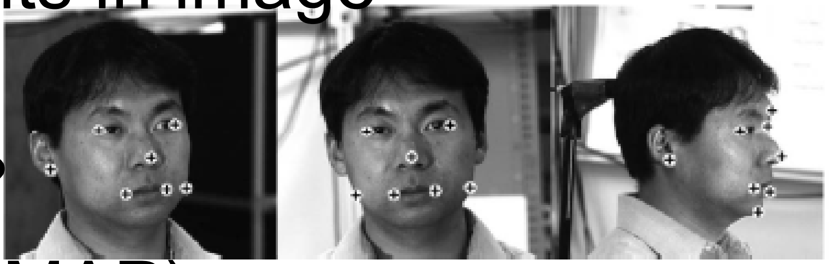
Image Synthesis - colors

- Color from ambient light, directed light, colors of texture
  - Phong illumination model – easy computations, good empirical performance
- Manually modify contrast / gains of colors
  - Fitting faces to pictures, paintings, etc
- Now we have raw pixels from parameters

Phong Illumination

Fitting the model

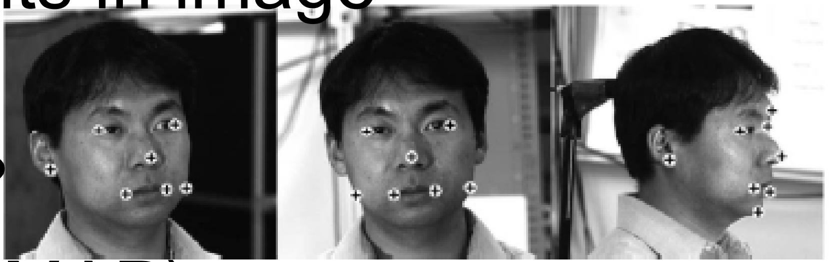
- Guess all parameters (about 3D model) from 2D image
- Cost function? Sum square differences of pixels
- Require user identify feature points in image corresponding points in ref face
  - Match up those points, call it good?



- Maximum a posteriori estimator (MAP)
  - Find most probable parameters given feature points, image
  - Bayes rule + liberal assumptions about independence / normality
  - Maximize: $p(I_{input}|\alpha, \beta, \rho) * p(F|\alpha, \beta, \rho) * P(\alpha) * p(\beta) * p(\rho)$

Fitting the model

- Guess all parameters (about 3D model) from 2D image
- Cost function? Sum square differences of pixels
- Require user identify feature points in image corresponding points in ref face
  - Match up those points, call it good?



- Maximum a posteriori estimator (MAP)
  - Find most probable parameters given feature points, image
  - Bayes rule + liberal assumptions about independence / normality
  - Maximize: $p(I_{input}|\alpha,\beta,\rho) * p(F|\alpha,\beta,\rho) * P(\alpha) * p(\beta) * p(\rho)$

Mean pixel error

Feature point error

Assumed uncorr. Multi-variable Gaussian

Optimization

- Minimize error in reconstruction
  - Newton's method
    - Assume minimum near zero of line with slope of gradient at certain point
    - Computationally efficient?
  - Stochastic Newton's method
    - Compute difference in pixels at subset of points (chosen probabilistically)
    - Calculate shadows sparingly
  - Optimize shape, texture, rigid transformation variables first (largest impact) then optimize others
  - After general parameters, compute segments

Experiments – the setup

- Performed model fitting / identification from two databases
  - CMU's PIE – 68 individuals, 66 images per person, different illuminations / viewpoints
  - FERET – 194 individuals, 10 images per person (relatively same expression)
- 6 feature points per image (standardized)
- Textures as shadows

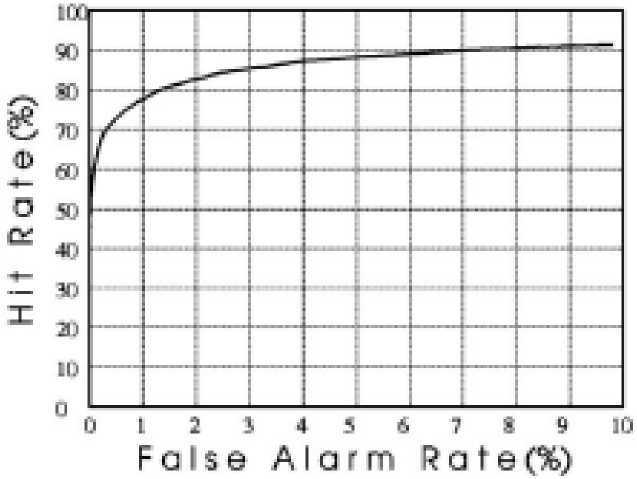
Recognition from coefficients

- After fitting model to image, what to do?
  - Concatenate all unit-var shape / texture coefficients
- Nearest neighbor classifier
  - Who is my nearest neighbor?
    - Mahalanobis distance?
    - Cosine of angle between vectors?
    - PCA analyze individual coefficient variance / LDA?
      - Ambiguous but apparently effective

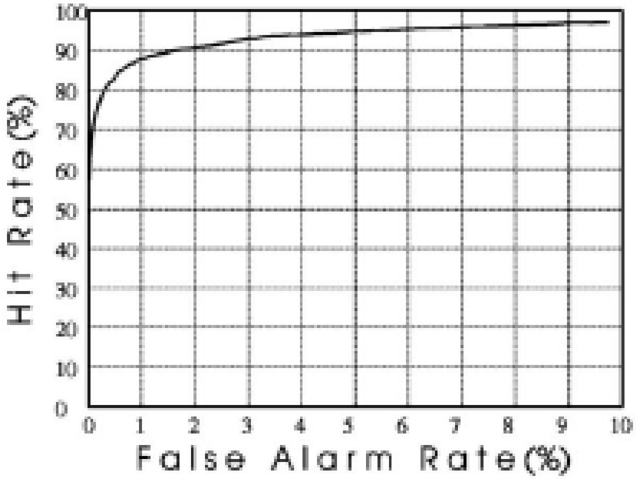
Database	d_M	d_A	d_W
CMU-PIE	87.2%	94.2%	**95.0%**
FERET	80.3%	92.2%	**95.9%**

Recognition performance

probe view	gallery view					
	front		side		profile	
front	99.8%	(97.1–100)	99.5%	(94.1–100)	83.0%	(72.1–94.1)
side	97.8%	(82.4–100)	99.9%	(98.5–100)	86.2%	(61.8–95.6)
profile	79.5%	(39.7–94.1)	85.7%	(42.6–98.5)	98.3%	(83.8–100)
total	92.3 %		95.0 %		89.0 %	



CMU PIE



FERRET