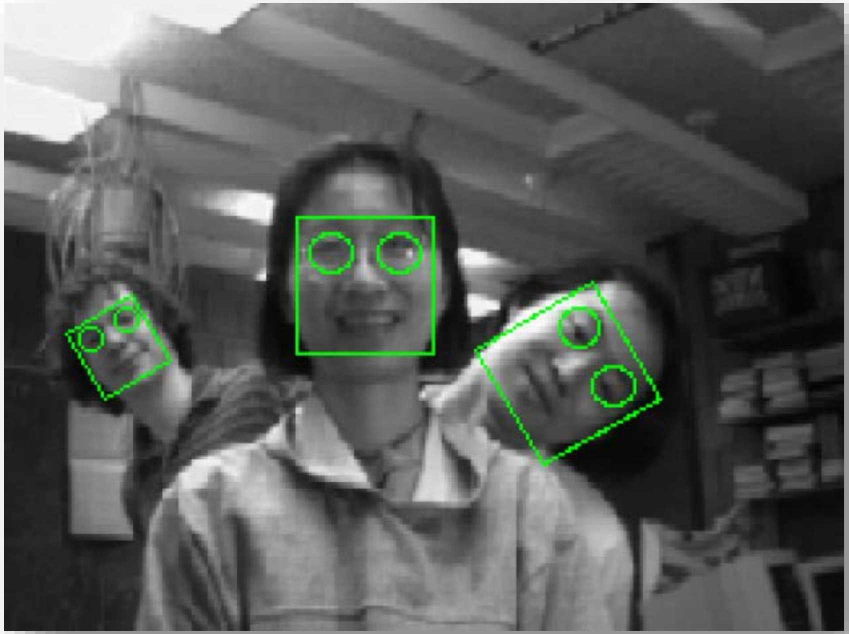
ECSE-626

Statistical Computer Vision

Probabilistic Face Detection

Face Detection

- Face Detection: given an image, where is the face?



Slide credit: Kristen Grauman (UoT)

Face Detection- Current Application Areas

- Face Detection in Cameras and Photo Sharing Websites

4 faces are detected

Image credit: Nikon S60

Face Detection- Current Application Areas

- Social Networking → fun applications
  - Face swap

Image courtesy of Facebook faceswap

Face Detection- Current Application Areas

- Facial Trait Classification → Soft Biometrics
  - Gender, Ethnicity, Age

Female, 32, Caucasian

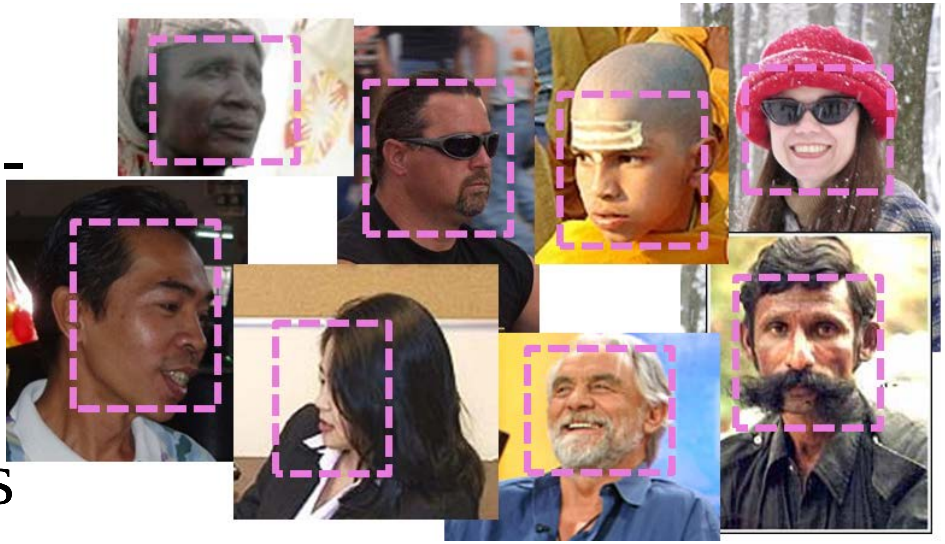
Male, 73, Black

- Fashion Industry
  - Virtual makeup

Image courtesy of dailymakeover.com and MoreIdeas, Inc.

Face Detection Challenges

- Various view-points
- Illumination changes
  - locally, globally, non-uniform
- Different skin colors
- Various face expressions
- Occlusions
  - sun glasses, hair, hand
- Various face scales



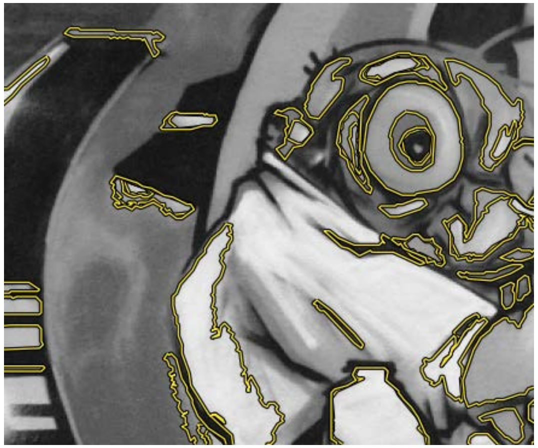
Face Representation

- We can either represent a face image
  - Globally
    - Eigenfaces

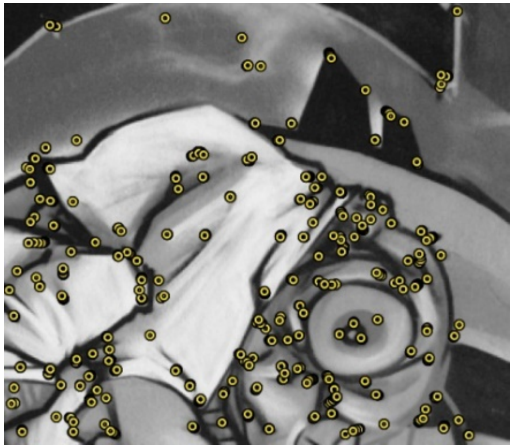
- Disadvantage
  - Cannot model local variation
    - A face with/without sunglasses
    - Pose variations

Face Representation

- We can either represent a face image
  - Locally, using features
    - Corners
    - Blobs and Regions
    - SIFT (Scale-Invariant Feature Transform)



Blobs and Regions (MSER: maximally stable external regions)



Harris Corners

SIFT features

Face Representation

- We need Feature Invariance
- Features that would be picked up if
  - Brightness changes
  - Scale changes
  - Rotation changes

Quick detour on SIFT

SIFT Features

- SIFT[1] → Scale-Invariant Feature Transform
- Identify the same image features in the presence of (invariant to)
  - Translation
  - Rotation
  - Scale change

Feature Description Step

Feature Detection Step

- Two Stage Feature Extraction
  - Feature Detection
  - Feature Description

SIFT Feature Detection

- Purpose
  - Automatically identify (detect) features in an image
- Three Main Steps
  - Create a Gaussian image scale space G(x,y,s)
    - Gaussian Pyramid
  - Generate Difference-of-Gaussians (DOGs)
  - Search for peaks (extremas) in DOGs

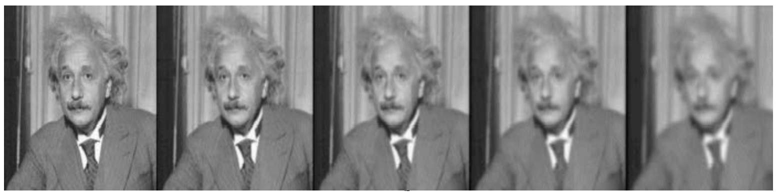
SIFT Feature Detection

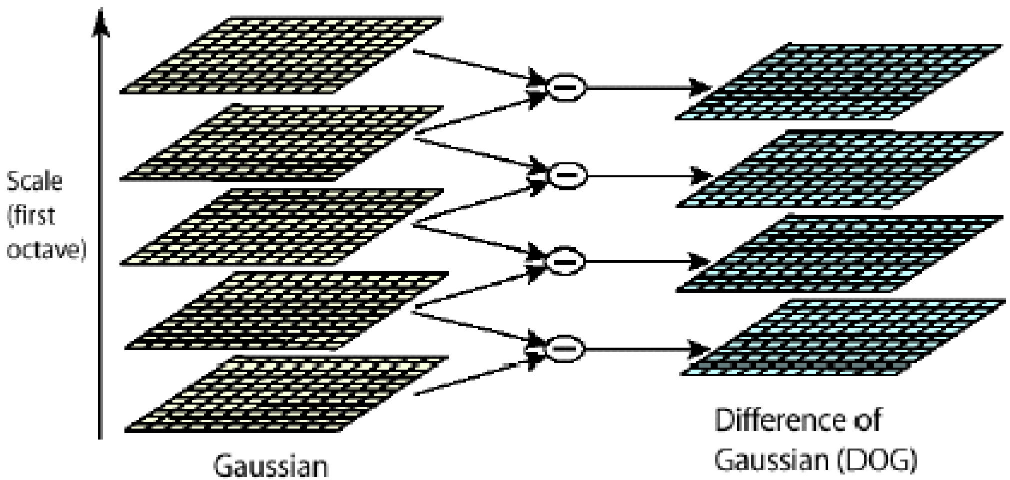
STEP 1:
Gaussian image
scale space

STEP 2: DOG
pyramid

SIFT Feature Detection

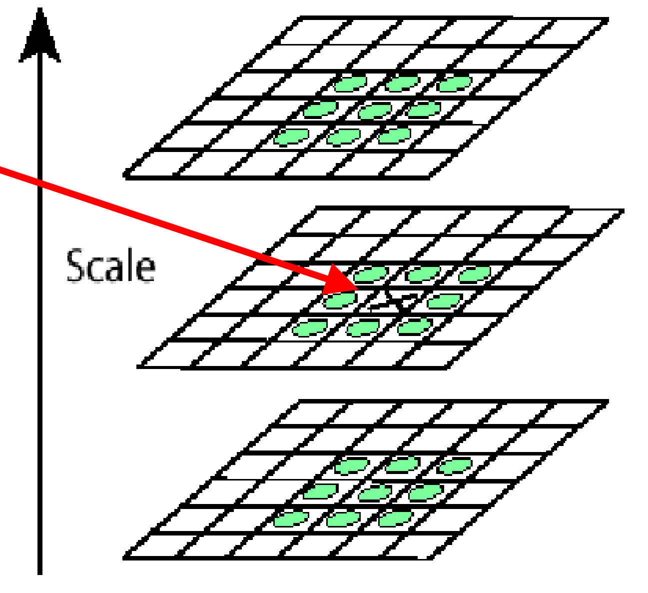
- Difference-of-Gaussian (DOG) Generation





SIFT Feature Detection

- STEP 3: Extrema (peak) detection in scale-sapce
  - 26 neighbors in a 3 × 3 × 3 neighborhood
  - Candidate Feature → Max or Min



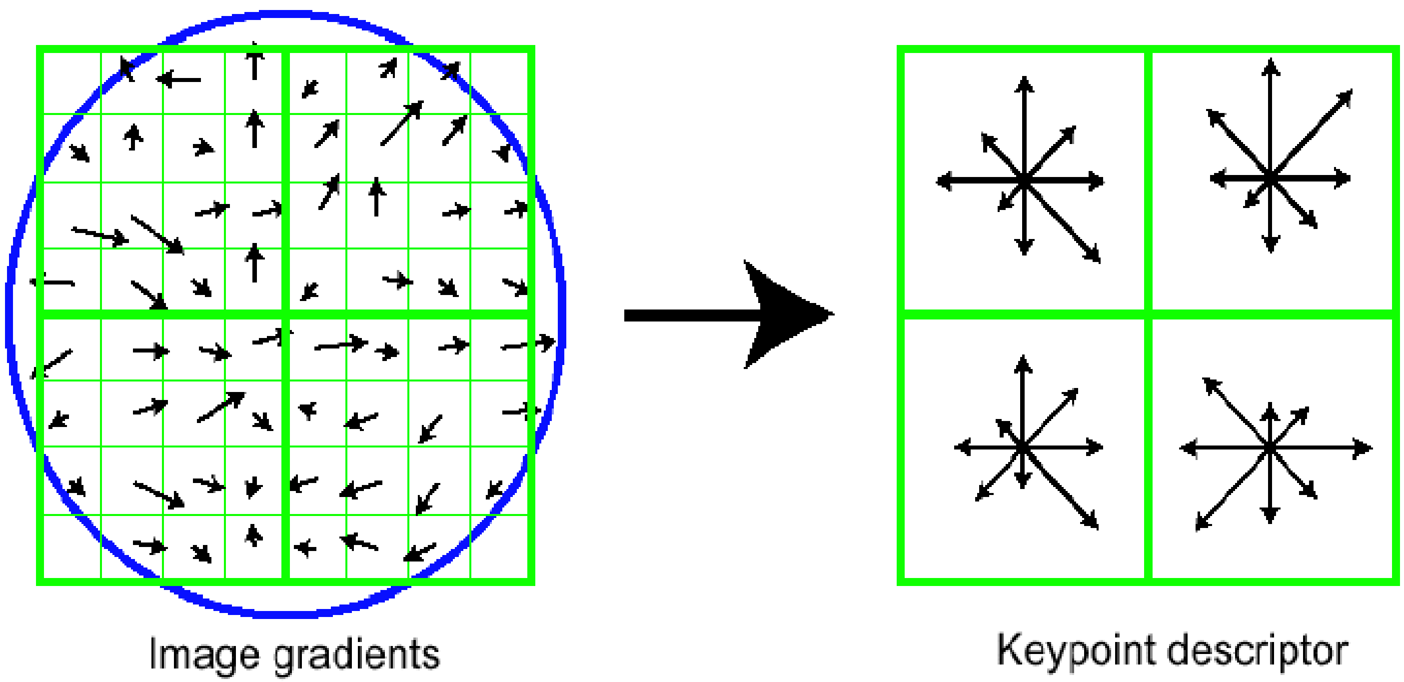
SIFT Feature Description

- Purpose
  - Encode feature image content
  - Distinguish one feature from another
- Different possibilities
  - Pixel intensities
  - Gradient information

SIFT Descriptor

SIFT Feature Description

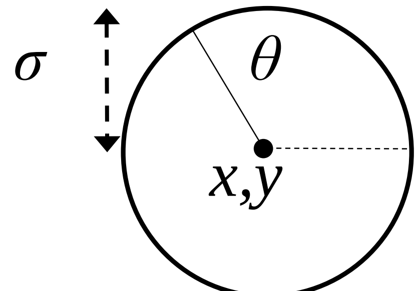
- Localized image gradient histograms



SIFT Feature Description

Feature Geometry

- Location *x,y*
- Orientation θ
- Scale σ



Feature Appearance

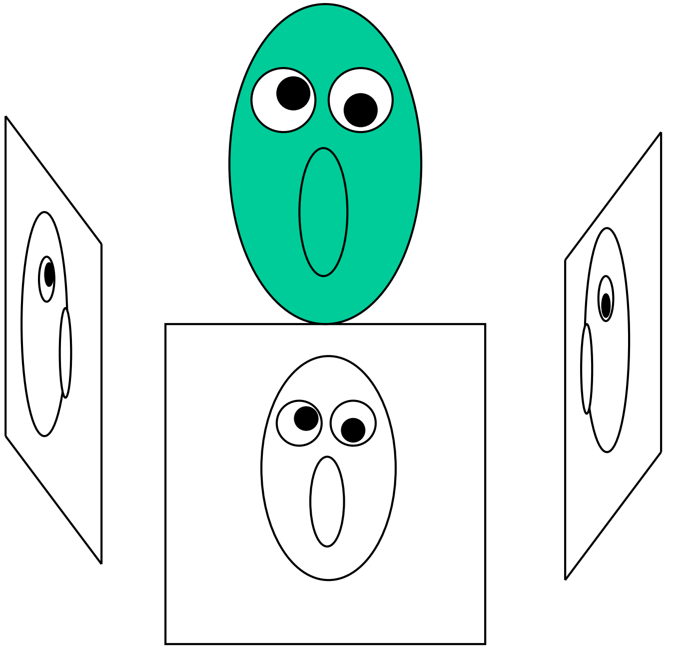
- Gradient Histogram
- 1x128 vector for each feature

same face
different
illumination

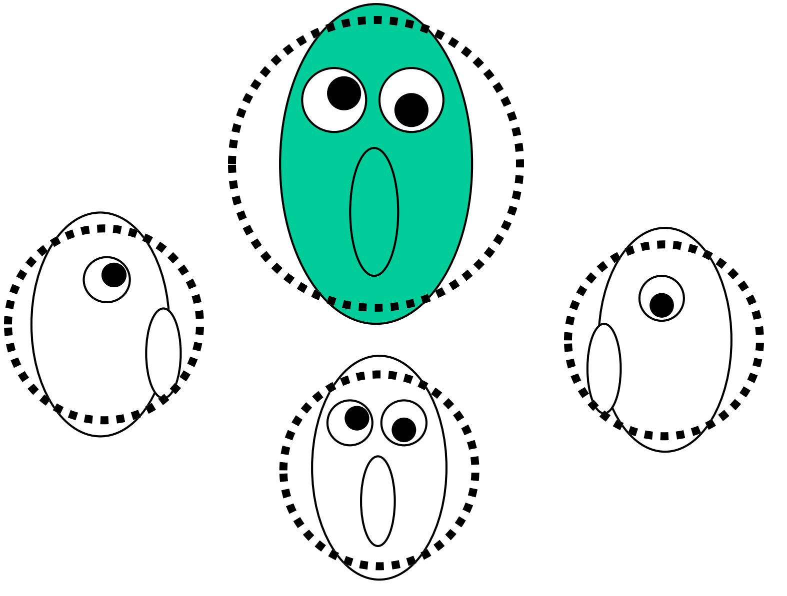
End of detour on SIFT

Face Detection via Different Models

Multi-view model



Viewpoint-invariant model



OCI-based Face Detection[1]

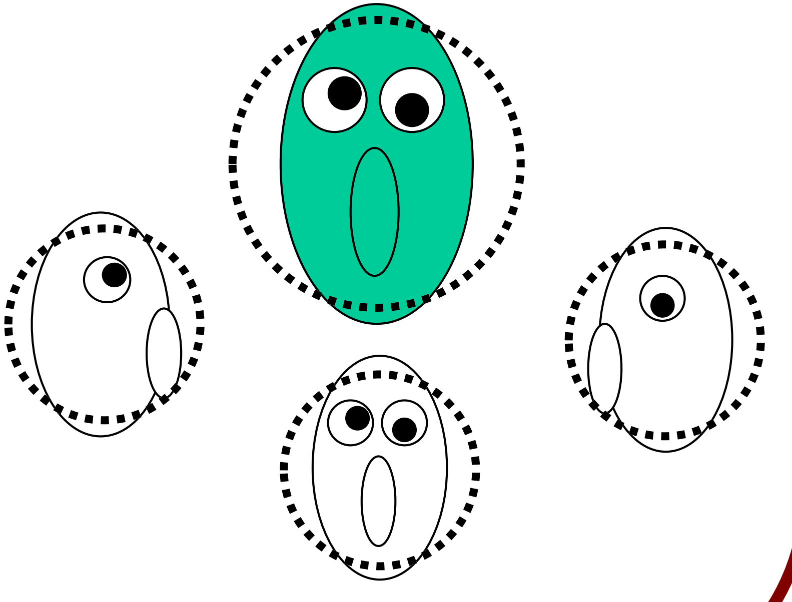
[1] *Toews & Arbel, PAMI 2009*

Viewpoint-invariant Model

- Simplicity: no viewpoint variable
- Needs a training database where all viewpoints are observed

Face Detection via the Object Class Invariant (OCI)

- Probabilistically models SIFT features over viewpoint change
- Introduces Object Class Invariant (OCI) idea
- Steps
  - OCI Model Learning → Training
  - OCI Model Fitting → Testing



OCI-based Face Detection

Face Detection via the Object Class Invariant (OCI)

- OCI (Object Class Invariant)
  - A geometrical reference frame that is
    - Uniquely defined for each object class instance
    - Invariant to the geometrical transforms



OCI Framework

- Training
  - Input data
    - Natural images, clutter, variation in all possible viewpoints
    - Manually labeled OCI for all training images
    - Extracted SIFT features for all training images

OCI Model Learning

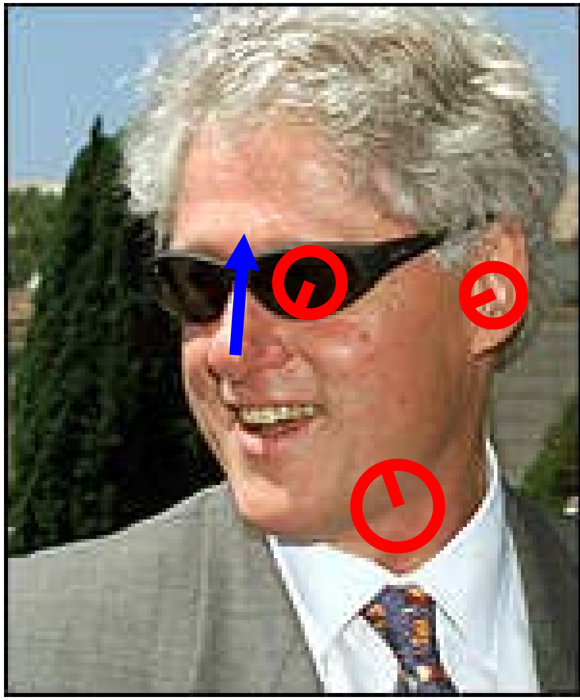
Object Class Invariant

$o:\{o^b, o^g\}$

Occurrence:
binary presence
or absence.
$o^b \in \{0,1\}$
A binary variable

Geometry: location,
scale, orientation of the OCI.
$o^g = [x, y, \sigma, \theta]$
Location: (x, y)
Scale: σ
Orientation: θ

OCI Model Learning



Object Class Invariant $o : \{o^b, o^g\}$

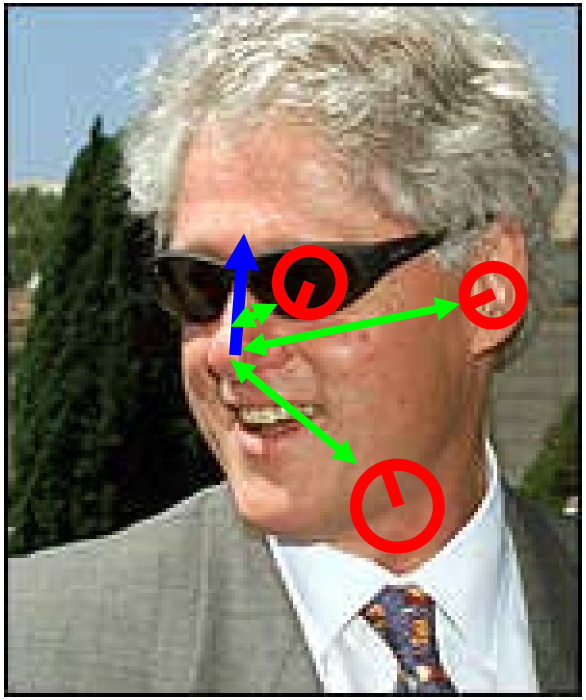
SIFT Features $m_i : \{m_i^b, m_i^g, m_i^a\}$

Occurrence:
binary presence
or absence.
$m_i^b \in \{0,1\}$
A binary variable

Geometry: location,
scale, orientation of the OCI.
$m_i^g = [x, y, \sigma, \theta]$
Location: (x, y)
Scale: σ
Orientation: θ

Appearance
SIFT descriptor
$m_i^a \in \mathbb{R}^{1\times128}$

OCI Model Learning



Object Class Invariant $o:\{o^b, o^g\}$

SIFT Features $m_i:\{m_i^{\,b}, m_i^{\,g}, m_i^{\,a}\}$

Transform feature and OCI geometries:

Normalize the geometry of each SIFT feature with respect to the geometry of the OCI

- location $(x_{m_i}, y_{m_i}) = (x_{m_i}, y_{m_i}) - (x_o, y_o)$
- Scale: $\sigma_{m_i} = \sigma_{m_i}/\sigma_o$
- Orientation: $\theta_{m_i} = \theta_{m_i} - \theta_o$

Face detection using OCI

- For a new test image
  - First, we need to extract SIFT features $\{m_i\}$
  - Then, we need to find the OCI (o) using them

- Probabilistic Relationship between OCI (o) and features (m_i) using Bayes rule

$$p(o|\{m_i\}) = \frac{p(\{m_i\}|o)p(o)}{p(\{m_i\})}$$

Face detection using OCI

- Posterior distribution of the OCI
  - Assuming conditional feature independence given the OCI

$$p(o|\{m_i\}) = \frac{p(\{m_i\}|o)p(o)}{p(\{m_i\})} = p(o)\frac{\prod_i p(m_i|o)}{p(\{m_i\})}$$

Prior distribution of the OCI

- Posterior distribution of the OCI

$$p(o|\{m_i\}) \propto p(o) \prod_i p(m_i|o)$$

- $p(o)$: Prior knowledge on the OCI
  - If do not have any prior knowledge , this would be a uniform distribution
  - If we know that it is more likely to have faces in the center of the image, we can assume the prior to be a 2D Gaussian distribution, centered on the image

Likelihood

- Posterior distribution of the OCI

$$p(o|\{m_i\}) \propto p(o) \prod_i p(m_i|o)$$

- $p(m_i|o)$ defines the relationship between an individual feature and the OCI
- We can learn this from the training set
- Therefore, modelling should focus on learning $p(m_i|o)$

Modelling $p(m_i|o)$

- Modelling should focus on learning $p(m_i|o)$

$$p(m_i|o) = p(m_i^a, m_i^b, m_i^g | o^b, o^g)$$

- Assuming conditional independence of feature appearance/occurrence (m_i^a, m_i^b) and feature geometry (m_i^g) given the OCI o

$$p(m_i|o) = p(m_i^a, m_i^b | o^b, o^g) p(m_i^g | o^b, o^g)$$

Modelling $p(m_i|o)$

$$p(m_i|o) = p(m_i^a, m_i^b|o^b, o^g)p(m_i^g|o^b, o^g)$$

- Using chain rule

$$p(m_i|o)$$
$$= p\left(m_i^a \middle| m_i^b, o^b, o^g\right)p(m_i^b|o^b, o^g)p(m_i^g|o^b, o^g)$$

Modelling $p(m_i|o)$

$$p(m_i|o) = p(m_i^a|m_i^b, o^b, o^g)p(m_i^b|o^b, o^g)p(m_i^g|o^b, o^g)$$

- Assuming conditional independence of feature occurrence m_i^b and the OCI geometry o^g given OCI occurrence o^b

$$p(m_i|o) = p(m_i^a|m_i^b, o^b, o^g)p(m_i^b|o^b)p(m_i^g|o^b, o^g)$$

Modelling $p(m_i|o)$

$$p(m_i|o) = p\left(m_i^a \middle| m_i^b, o^b, o^g\right) p(m_i^b|o^b) p(m_i^g|o^b, o^g)$$

- Assuming conditional independence of feature appearance m_i^a and the OCI o given feature occurrence m_i^b

$$p(m_i|o) = p\left(m_i^a \middle| m_i^b\right) p(m_i^b|o^b) p(m_i^g|o^b, o^g)$$

Modelling $p(m_i|o)$

$$p(m_i|o) = p(m_i^a|m_i^b)p(m_i^b|o^b)p(m_i^g|o^b, o^g)$$

- $p(m_i^a|m_i^b)$ is feature appearance given presence
  - Modeled as a Gaussian
- $p(m_i^b|o^b)$ is the feature occurrence given OCI occurrence
  - Modeled as a binomial probability
- $p(m_i^g|o^b, o^g)$ is the residual error in predicting the OCI geometry from the feature geometry
  - Modeled as a Gaussian

Modelling $p(m_i|o)$

$$p(m_i|o) = p\left(m_i^a \middle| m_i^b\right) p(m_i^b|o^b) p(m_i^g|o^b, o^g)$$

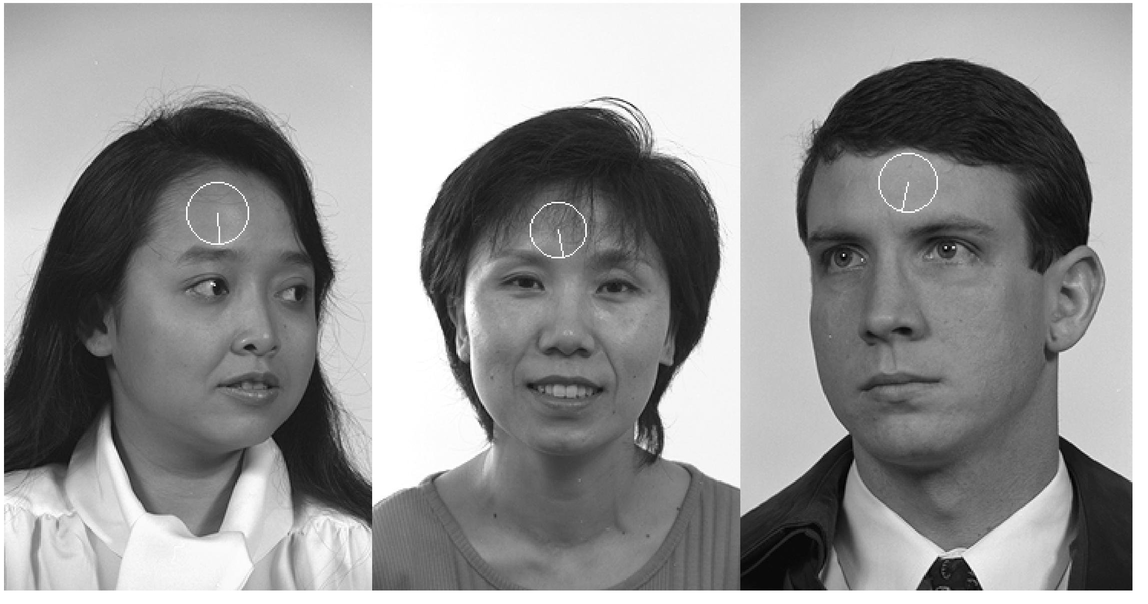
- Learning the OCI model requires estimating the parameters of each distribution

Feature Clustering

- Originally, we assumed each SIFT feature is an m_i
- This causes too many unrelated features for training
- A better choice is to cluster the SIFT features and use the cluster centers as m_i
- We can expect that each cluster center represents the geometry/appearance of many similar SIFT feature from all the training images

Example of Cluster Centers

- Identify clusters of features
  - Similar in geometry relative to OCI
  - Similar in appearance



Face Detection using the OCI

- 1. Extract SIFT features from the test image
- 2. Quantize the features to the closest cluster center from the training set
- 3. Use the learned $p(m_i|o)$ to estimate $p(o|\{m_i\})$
- Identifying OCI instance in new image
  - Probabilistic voting

Experimental Result