ECE5554 – Computer Vision
Lecture 11c – Face Recognition

Creed Jones, PhD

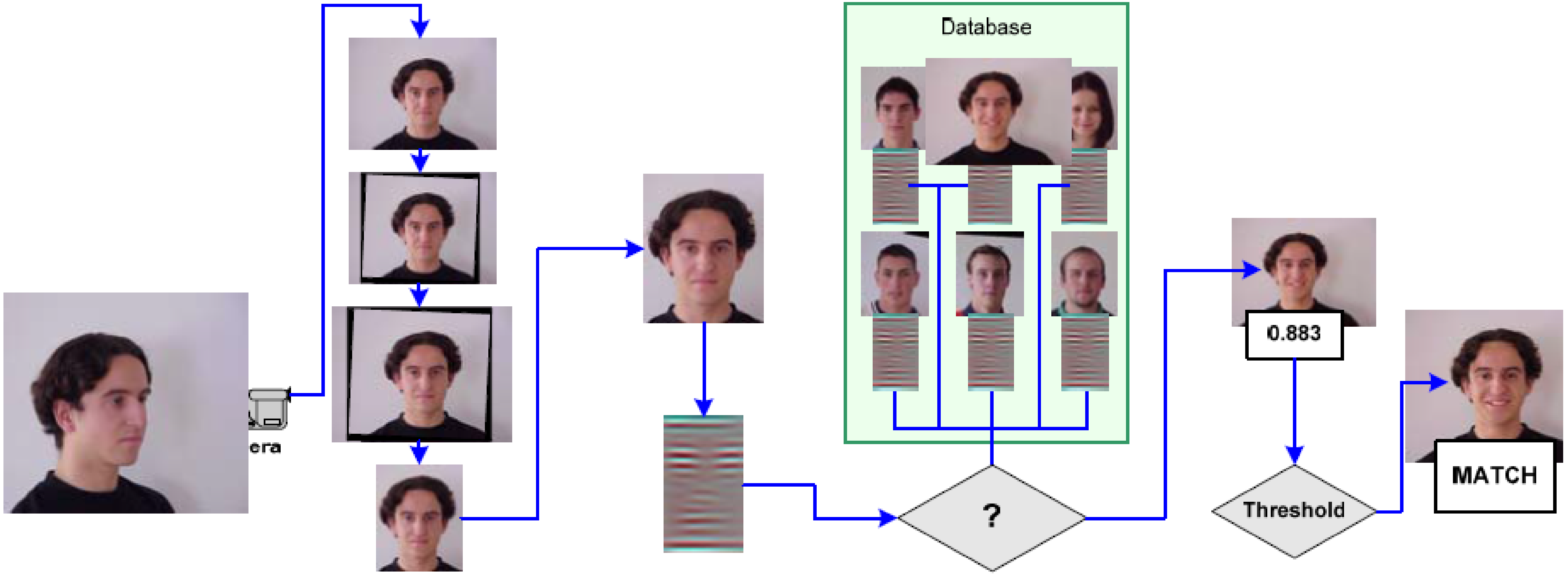
Today's Objectives

Face Recognition
- Concept and Terms
- Methods
  – Eigenface
  – Local Feature Analysis
  – Elastic Bunch Graph Matching
  – Newer Methods
- Performance Measurement
- Current Issues
  – Surveillance ethics
  – Bias

- A Brief Review

Concept and Terms

- Given a test image of a face, attempt to find the best match in a database of previously stored samples (face images or feature representations)
- Identification: a 1-to-many match against a set of previously stored identities.
- Verification: a 1-to-1 match against an asserted or presumed identity.
- What is the imaging scenario? ID photos? “Person in the street”?
- Many important issues regarding the stored data –
  - Quality, Age, Number of samples (overall, and per person)
- We need a measure of match quality, to enable a “face not found” result


era
Database
0.883
?
Threshold
MATCH

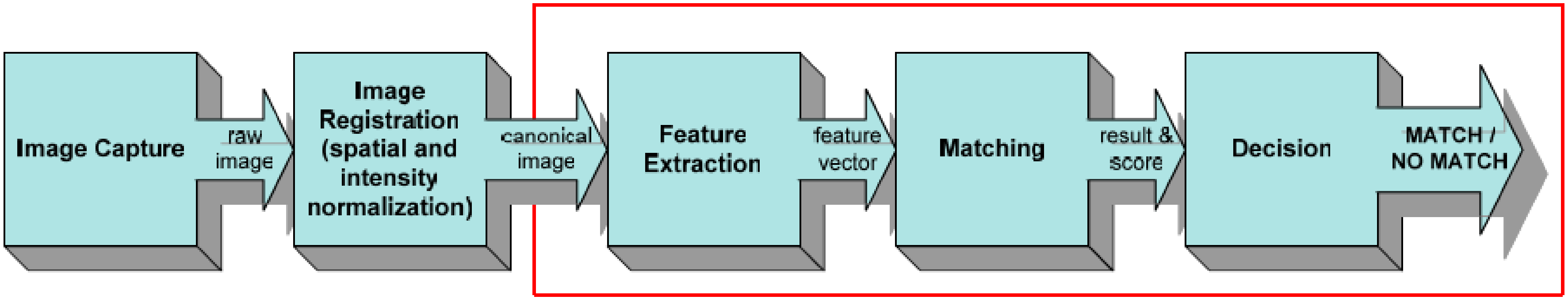


Image Capture
raw image
Image Registration (spatial and intensity normalization)
canonical image
Feature Extraction
feature vector
Matching
result & score
Decision
MATCH / NO MATCH

Face detection is a well-explored topic, thanks to facebook and related social media platforms

- The Viola-Jones algorithm works well for frontal face images
  - Apply Haar (square) wavelets to get a feature representation
  - Form the integral (cumulative) image
  - Train an Adaboost classifier
- Newer methods use Convolutional Neural Networks
  - But training time is long
  - Perhaps custom training is not needed?

Face Detection | Firebase

firebase.google.com/docs/ml-kit/detect-faces

Apps VT gmail VT cal gmail ECE OneCampus Canvas ECE Internal W! IEEE BBC Ms VT CVL Wiki Library Commi

Firebase Products Use Cases Pricing Docs Support Search Language Go to console

Documentation

Overview Guides Reference Samples Libraries Send feedback

Guides

Get started with Firebase

Manage your Firebase projects

Prototype and test with Emulator Suite β

Analytics

Extensions β

DEVELOP

Authentication

Realtime Database

Firebase > Docs > Guides

Face Detection

iOS

With ML Kit's face detection API, you can detect faces in an image, identify key facial features, and get the contours of detected faces.

With face detection, you can get the information you need to perform tasks like embellishing selfies and portraits, or generating avatars from a user's photo. Because ML Kit can perform face detection in real time, you can use it in applications like video chat or games that respond to the player's

Contents

Key capabilities

Example results

Example 1

Example 2 (face contour detection)

VIRGINIA TECH.

Face Registration ensures that most relevant features are in similar locations throughout the data

- To register the face to the rest of the database –
  - (a) The image is padded with black pixels
  - The eye positions are determined
  - (b) The image is rotated to align the eyes horizontally
  - The point of the chin is located
  - If necessary, the image is scaled in the vertical to the desired height factor
  - (c) The image is scaled to 60 pixels between pupil centers
  - (d) The result is translated and cropped to the proper size
    - Often, hair and clothing is cropped off

2D FACE RECOGNITION METHODS

Some popular methods for face recognition

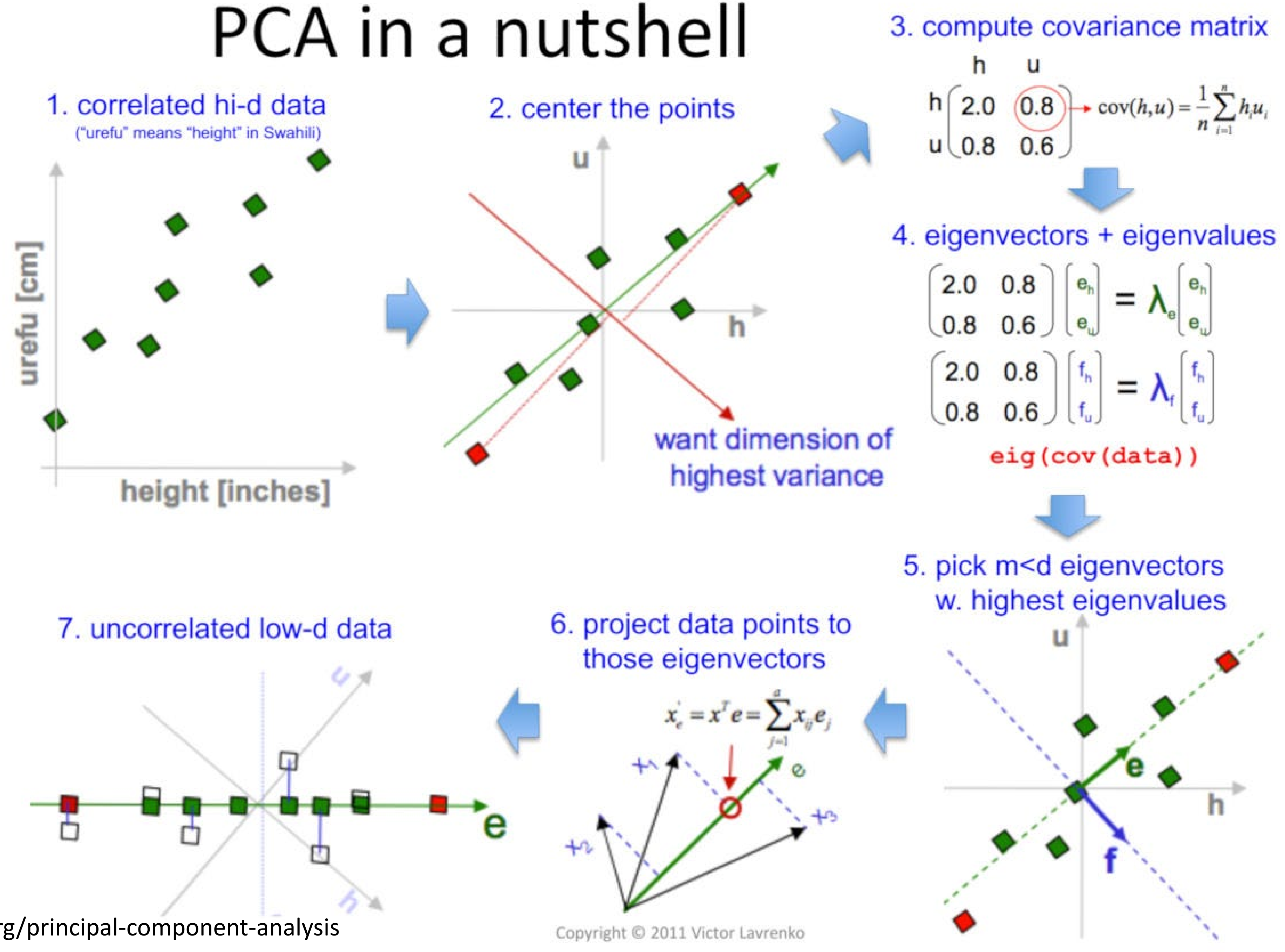
- Eigenface
  - Principal Components Analysis
  - Eigenfaces
  - Recognition
- Local Feature Analysis
- Bunch Graph methods
- Other methods
  - Range images (FaceID, …)
  - Pose-invariance

Principal Components Analysis analyzes a dataset and determines an alternate set of dimensions (the *principal components*) that are more suitable for representing the data

- Our data is a set of vectors $\boldsymbol{x}$ of length n
- The mean vector is $E\{\boldsymbol{x}\}$ and the covariance matrix is $\boldsymbol{C}_x = E\{(\boldsymbol{x} - \boldsymbol{m}_x)(\boldsymbol{x} - \boldsymbol{m}_x)^t\}$
  - The magnitude of off-diagonal elements (the covariance) indicates the degree of correlation between different elements of $\boldsymbol{x}$
- Assemble the eigenvectors of $\boldsymbol{C}_x$ into a matrix A, in decreasing order of eigenvalue
- Use A to transform the data set: $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{m}_{\boldsymbol{x}})$; this is called the Hotelling transform
  - Also, $\boldsymbol{C}_y = \boldsymbol{A}\boldsymbol{C}_x\boldsymbol{A}^{\boldsymbol{T}}$ is diagonal, with the diagonal elements being the eigenvalues

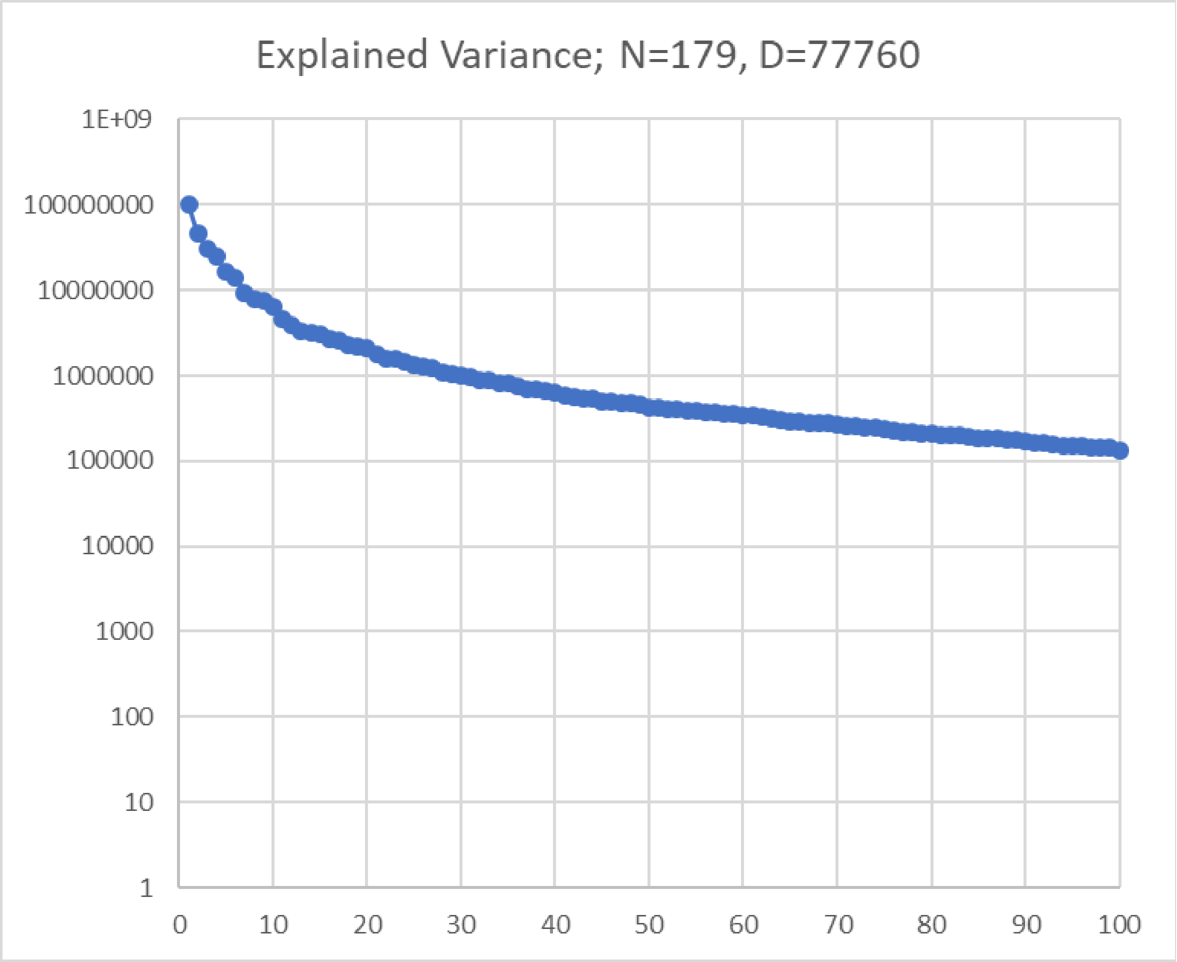
- **For data sets with some cross-correlation, the first k eigenvectors can be used as a new basis of representation, with minimal error**
  - **These k vectors are the Principal Components**

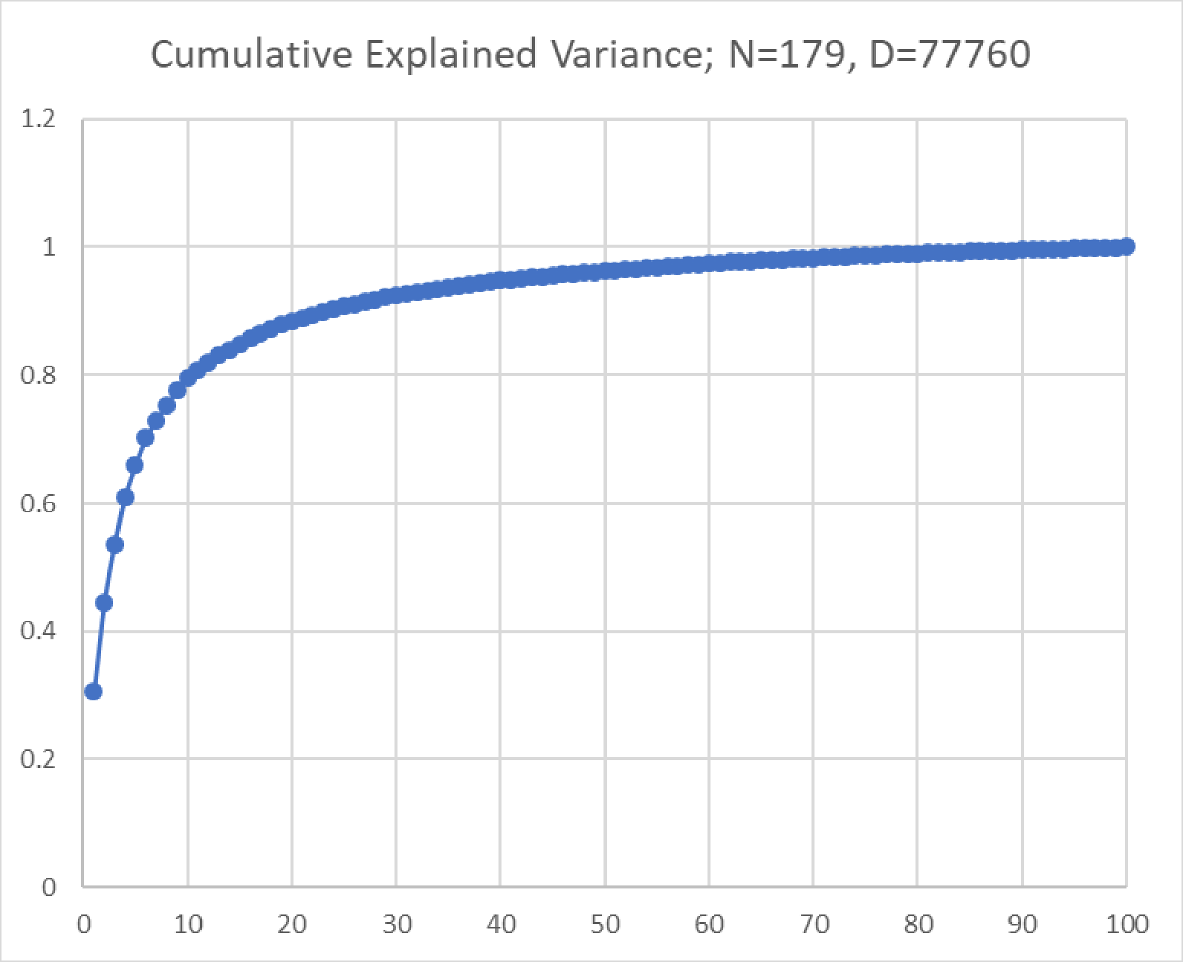
PCA in a nutshell



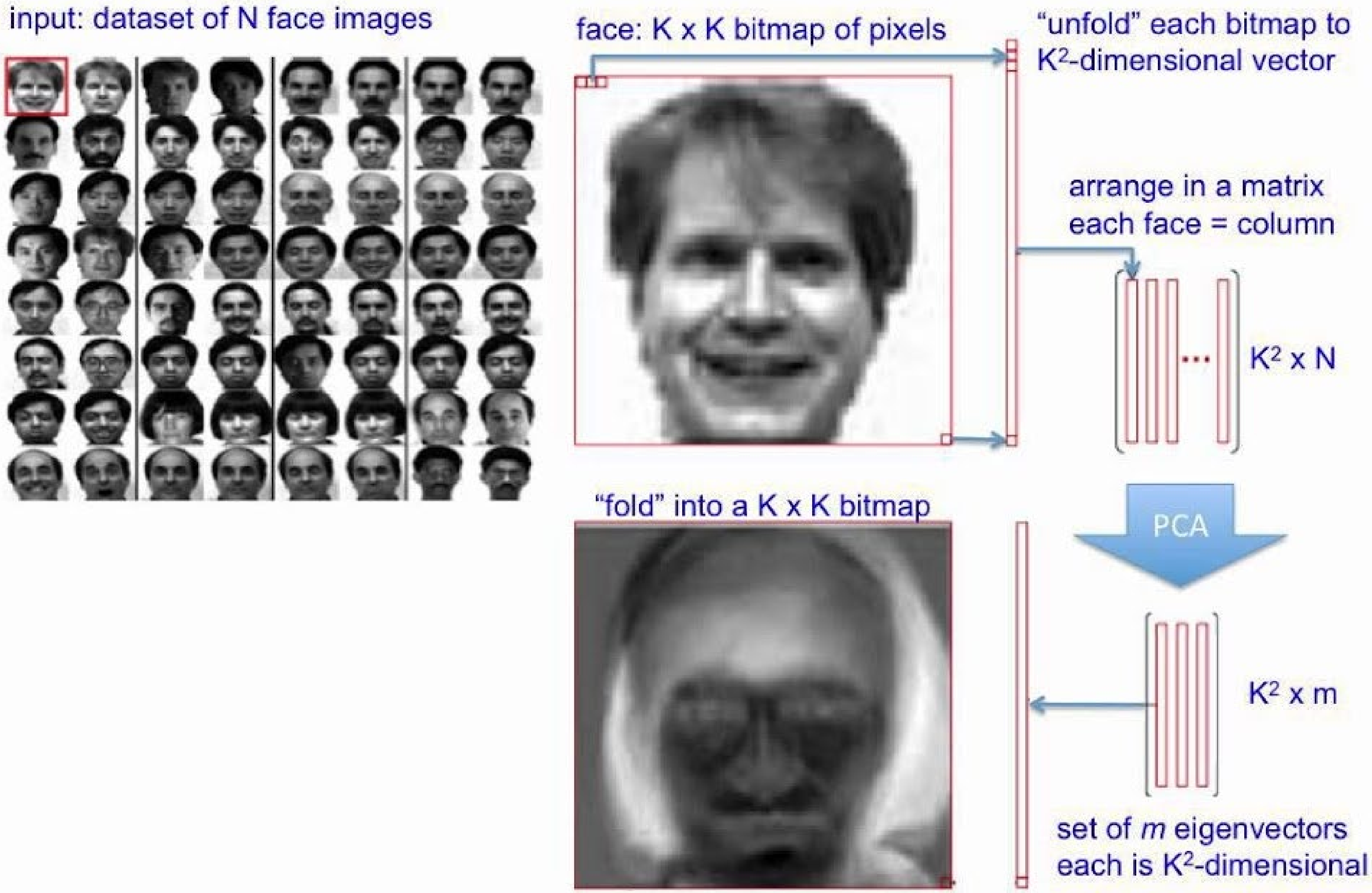
https://devopedia.org/principal-component-analysis

The amount of information (measured as variance) is concentrated in the first components





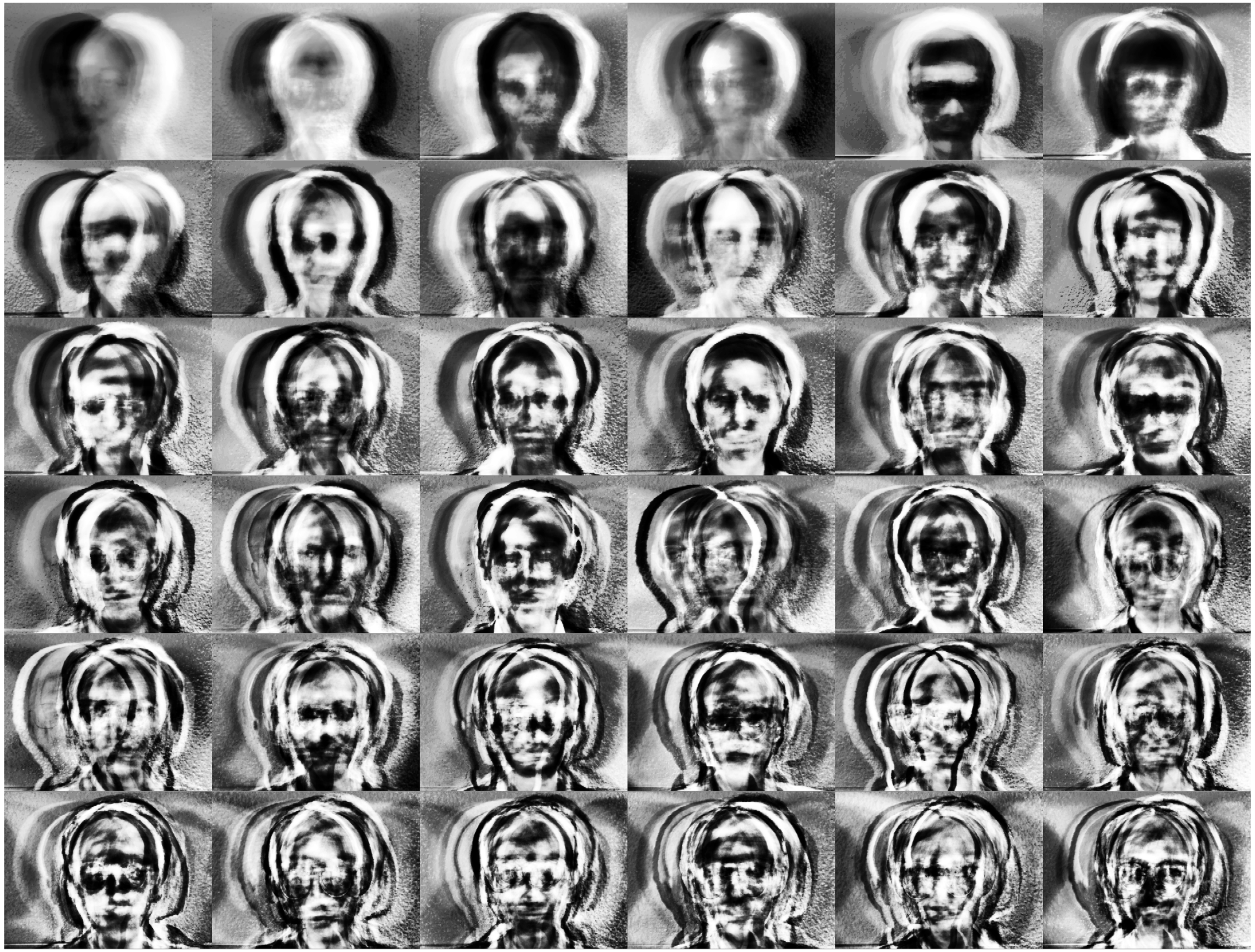
PCA example: Eigen Faces



```
YalePath = "C:\\Data\\RegisteredFaces\\Yale"
imgs = np.zeros((HEIGHT, WIDTH, NUMIMGS), float)
count = 0
for fileName in glob.glob(os.path.join(YalePath, '*.png')):
imgs[:, :, count] = cv2.cvtColor(cv2.imread(fileName), cv2.COLOR_BGR2GRAY)
    count = count + 1

result = np.mean(imgs, 2).astype(np.uint8)
saveImage(result, "averageFace")

reshaped = np.reshape(imgs, (HEIGHT*WIDTH, NUMIMGS))
pca = PCA(n_components=NUMCOMPS, copy=True, whiten=True)

pca.fit(reshaped.transpose())
components = pca.components_.transpose()
compImages = np.zeros((HEIGHT, WIDTH, NUMCOMPS), np.uint8)
for count in range(NUMCOMPS):
    thisImg = normImg(np.reshape(components[:,  count], (HEIGHT, WIDTH)))
    compImages[:, :, count] = thisImg
    saveImage(thisImg, "component" + str(count))
    print(pca.explained_variance_[count])
print("TOTAL Variance: ", np.sum(pca.explained_variance_))
compMosaic = makeMosaic(compImages, 6, 6)
saveImage(compMosaic, "ComponentMosaic")
```

The principal components are typically reshaped into images; these show the typical "ghostly" appearance

Eigenfaces computed on a different face database; the problem with poor registration can be seen



The Eigenface method is a simple algorithm with decent performance in controlled situations (ID card photos, etc.) – it's the usual "first algorithm" for face recognition

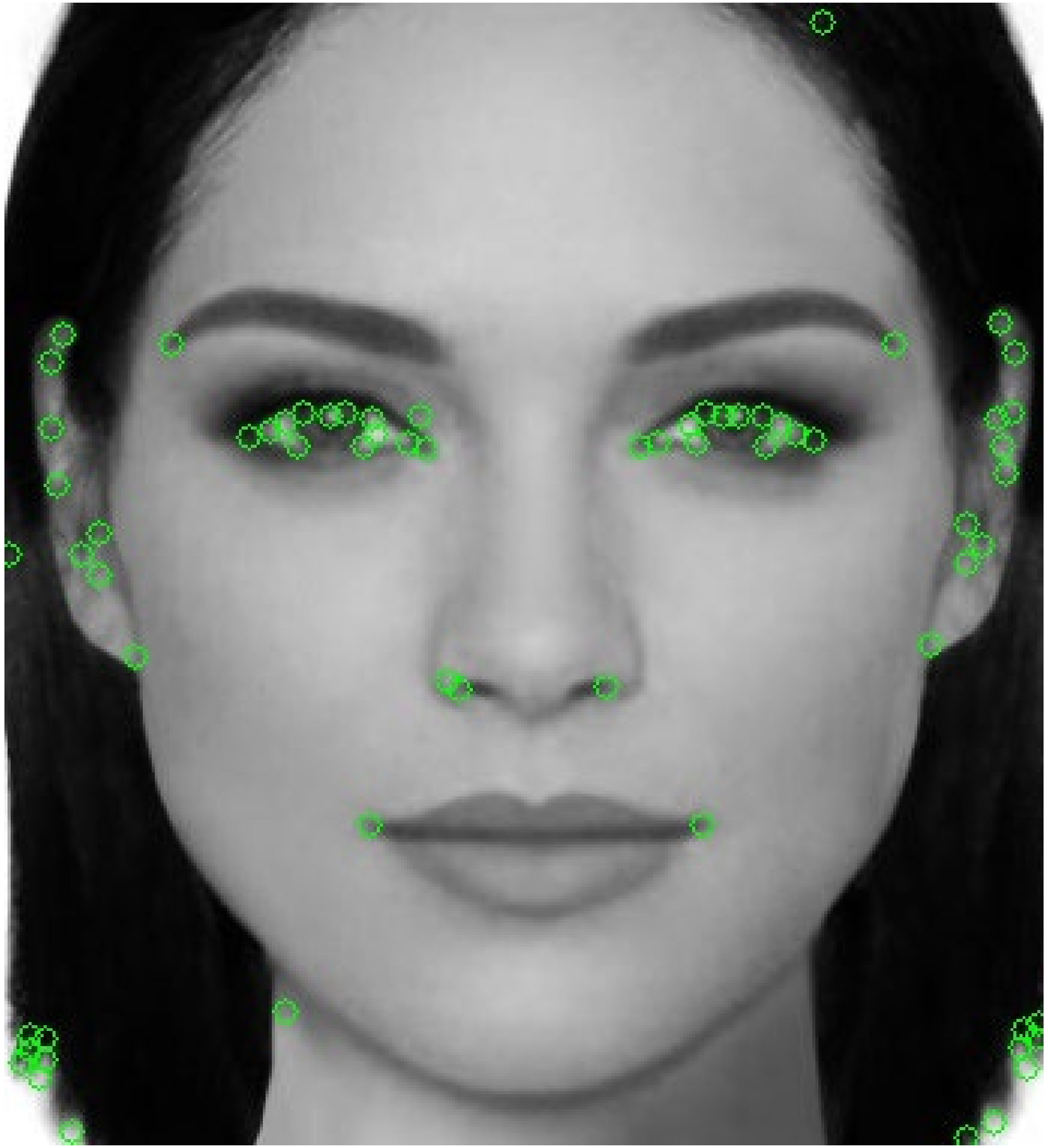
If the example faces are well registered and normalized, then the components that have been calculated to represent the variance will do a good job of discrimination

In the eigenface space, what known face (cluster center) is the test image closest to?

Problems with eigenface:

- It does not tolerate variations well (facial hair, cosmetics, etc.)
- It has no innate ability to deal with differences in pose of the face
  - Tilt of the head, for example
- Behavior on faces not in the database is unpredictable

Application of keypoint detection produces some good candidate locations for extracting jets (Shi-Tomasi is used here)



Local Feature Analysis is based on PCA, but preserves the spatial relationships of adjacent and nearby pixels

- First, the PCA representation is computed – eigenvalues λ_i and eigenvectors e_i
- Then a set of responses to the following family of kernels is computed and stored in a matrix with the same height and width as the original image

$$\boldsymbol{K}^{(n)}(x,y) = \sum_{r=1}^{k} \boldsymbol{e}_r(x) \lambda_r{}^{-n} \boldsymbol{e}_r(y)$$

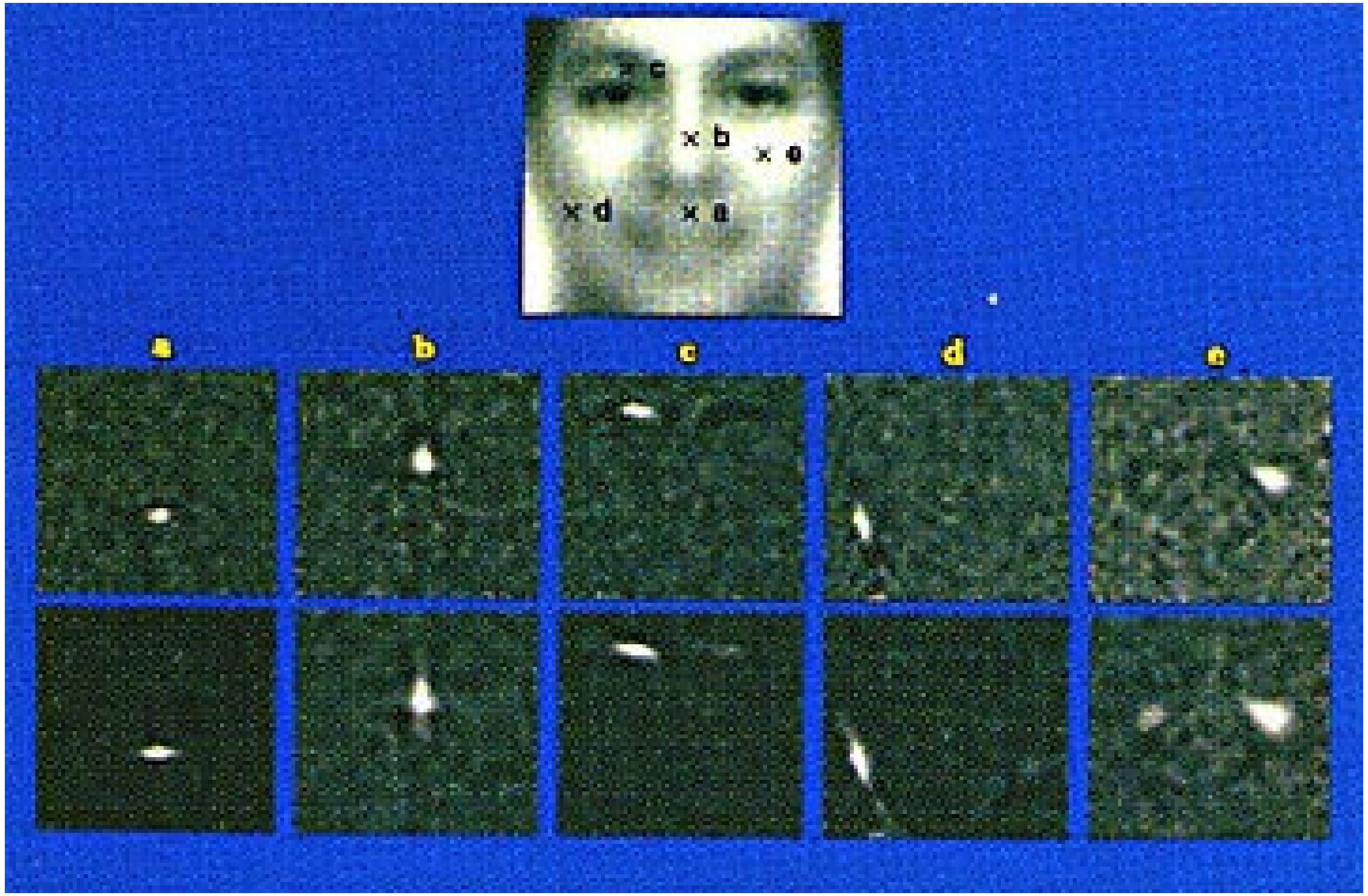
- The full representation has *k* planes, where k is the number of pixels in theimage

- On this multivalued "image", local features of most significance will appear bright
- These can be used in recognition, just as with PCA

- Fazi-Ersi 2009, "Local Feature Analysis for Robust Face Recognition", 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications

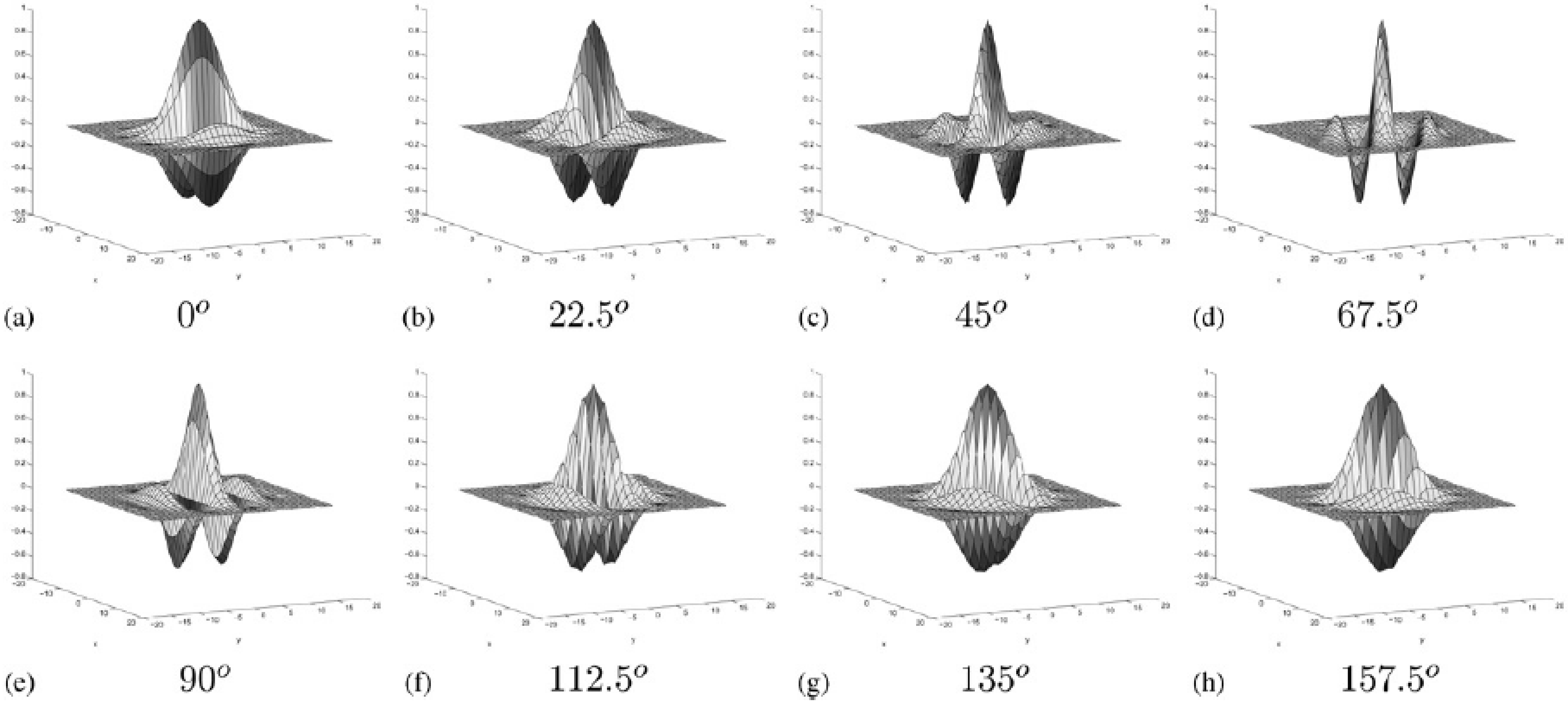
Local feature analysis finds *connected* patches of the face area that are the best in telling a large set of faces apart

- Nose, space above the lips, eyebrows, etc…

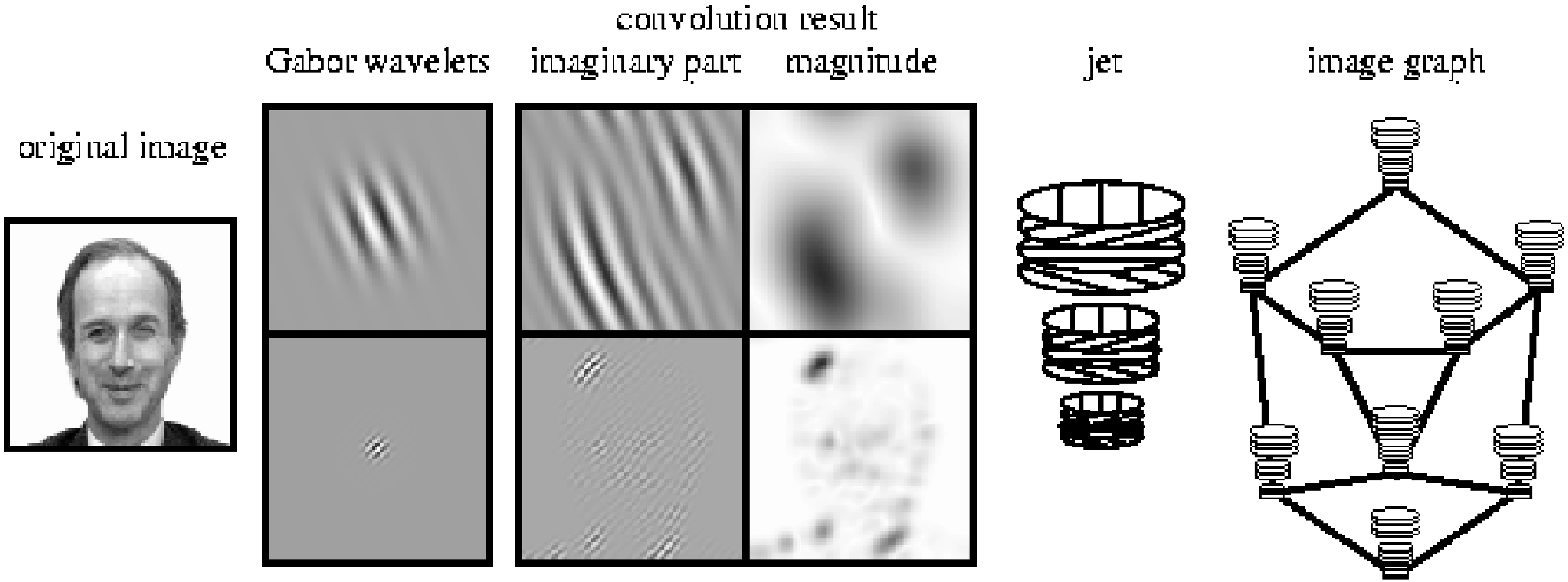


Elastic Bunch Graph Matching uses feature vectors extracted at key locations by the application of *Gabor filters*

- Simultaneous response to frequency and location of key features in the image

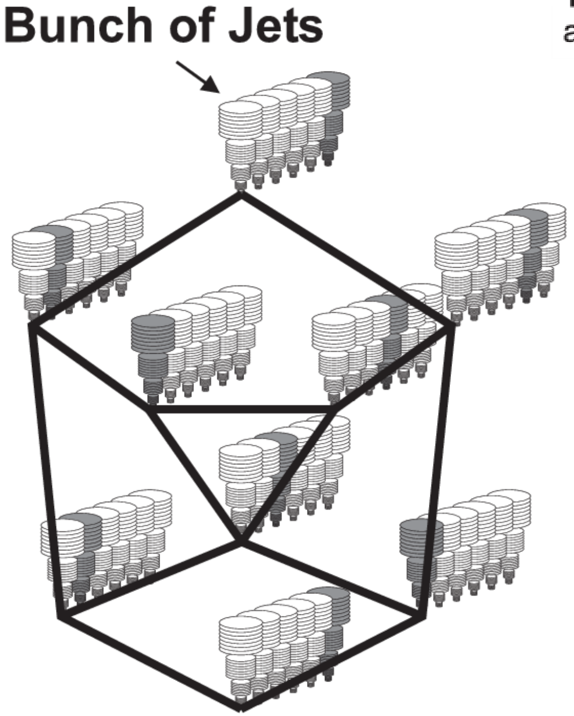


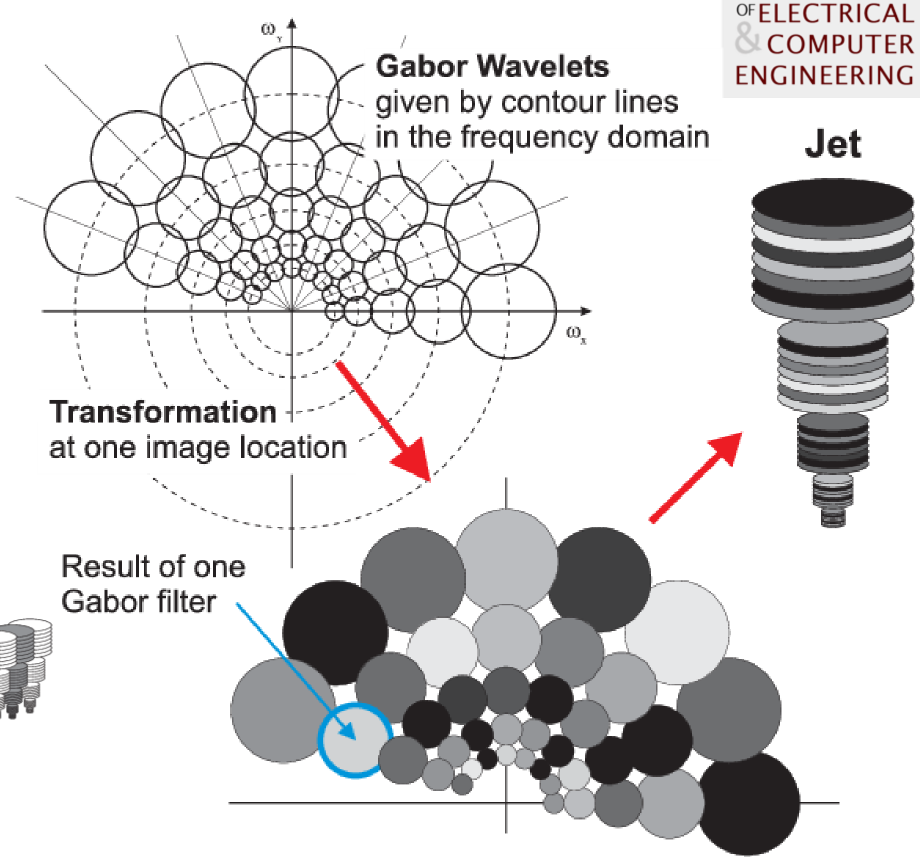
Applying the family of Gabor filters at each location will yield a (typically) 40-element complex vector for each pixel



EBGM is more tolerant of changes in pose than other methods, because the filter responses (called *jets*) are extracted at key locations

- Within a neighborhood, jets move to the location of best match to a candidate jet
- The best candidate face is selected by a goodness of fit measure, which is a weighted sum of the jet distances (test image to candidate image) and the distances between jet locations as compared to the candidate face





Elastic Bunch Graph Matching

Graph similarity measure

$$S_B(G^I, B) = \frac{1}{N}\sum_n \max_m \left(S_\varphi \left(J_n^I, J_n^{Bm} \right) \right) - \frac{\lambda}{E}\sum_e \frac{\left|\Delta \underline{x}_e^I - \Delta \underline{x}_e^B\right|^2}{\left|\Delta \underline{x}_e^B\right|^2}$$

$\Delta \underline{x}_e^I$: displacement on edge e

J_n^I : jet at node n

λ: weighting factor

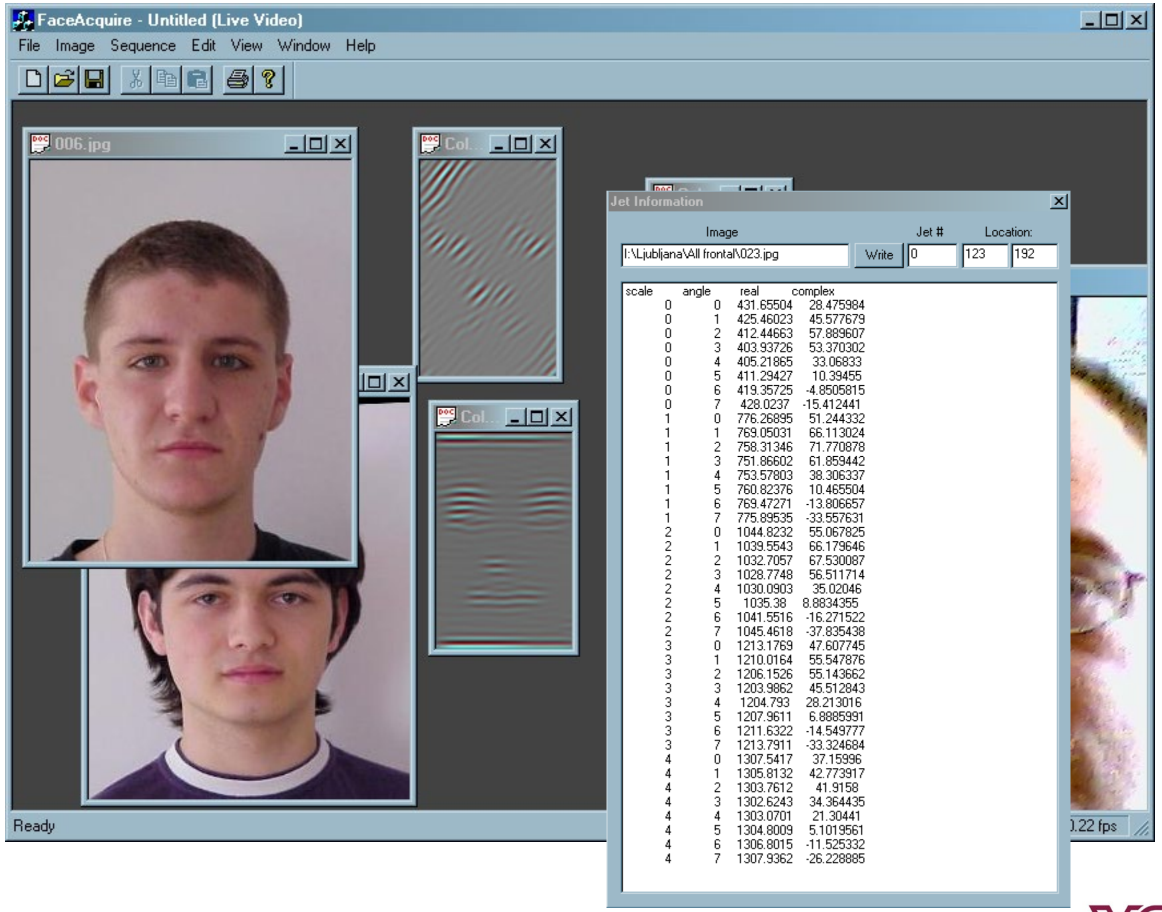
- Initially, manually generate a few FGs to create a FBG
- Heuristic algorithm to find the image graph that maximizes the similarity:
  - Coarse scan of image using jets to detect face
  - Varying sizes and aspect ratio of FBG to adapt to right format of face.
  - Finally, all nodes are moved locally to maximize S_B.

The newest methods of face recognition use deep learning for some or all of the classification task

- Li et al apply CNNs on age-wide databases to find optimal matches
  - Li 2018, "Distance metric optimization driven convolutional neural network for age invariant face recognition", Pattern Recognition #75, March 2018, pp. 51-62
- Cao uses the residuals of recognition to improve deep learning
  - Cao 2018, "Pose-Robust Face Recognition via Deep Residual Equivariant Mapping", IEEE CVPR 2018, pp. 5187-5196
- Lu et al address the problem of low resolution images from surveillance cameras by using networks trained at multiple image resolutions
  - Lu 2018, "Deep Coupled ResNet for Low-Resolution Face Recognition", IEEE Signal Processing Letters, #25:4, pp. 526-530

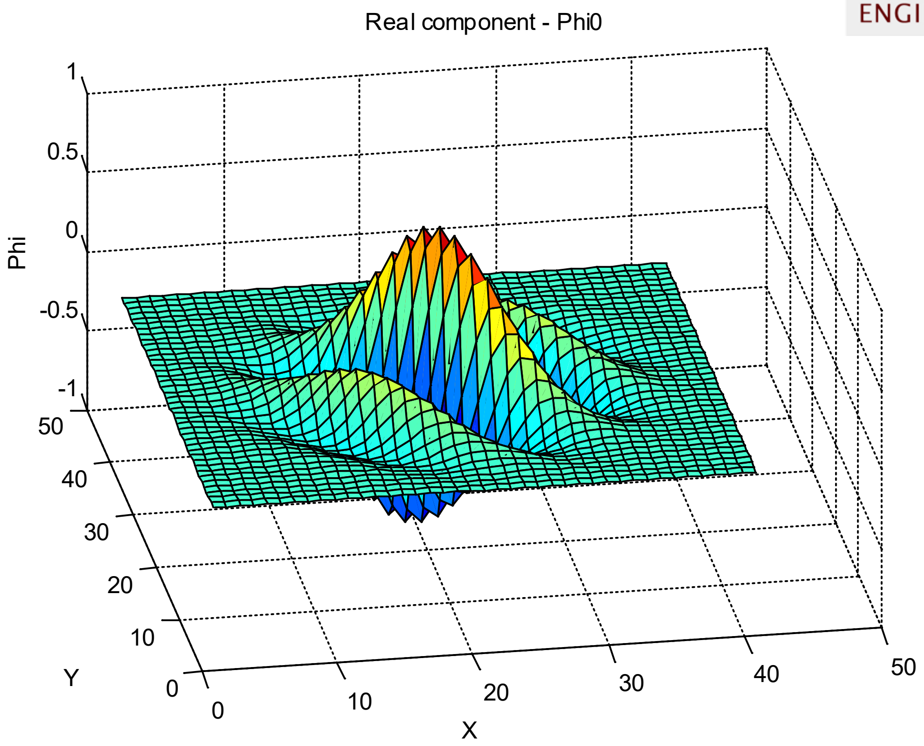
Color Face Recognition

- Currently, most automated face recognition is done using only monochrome images
- Color images contain more information – but new algorithms are needed to extract and use it



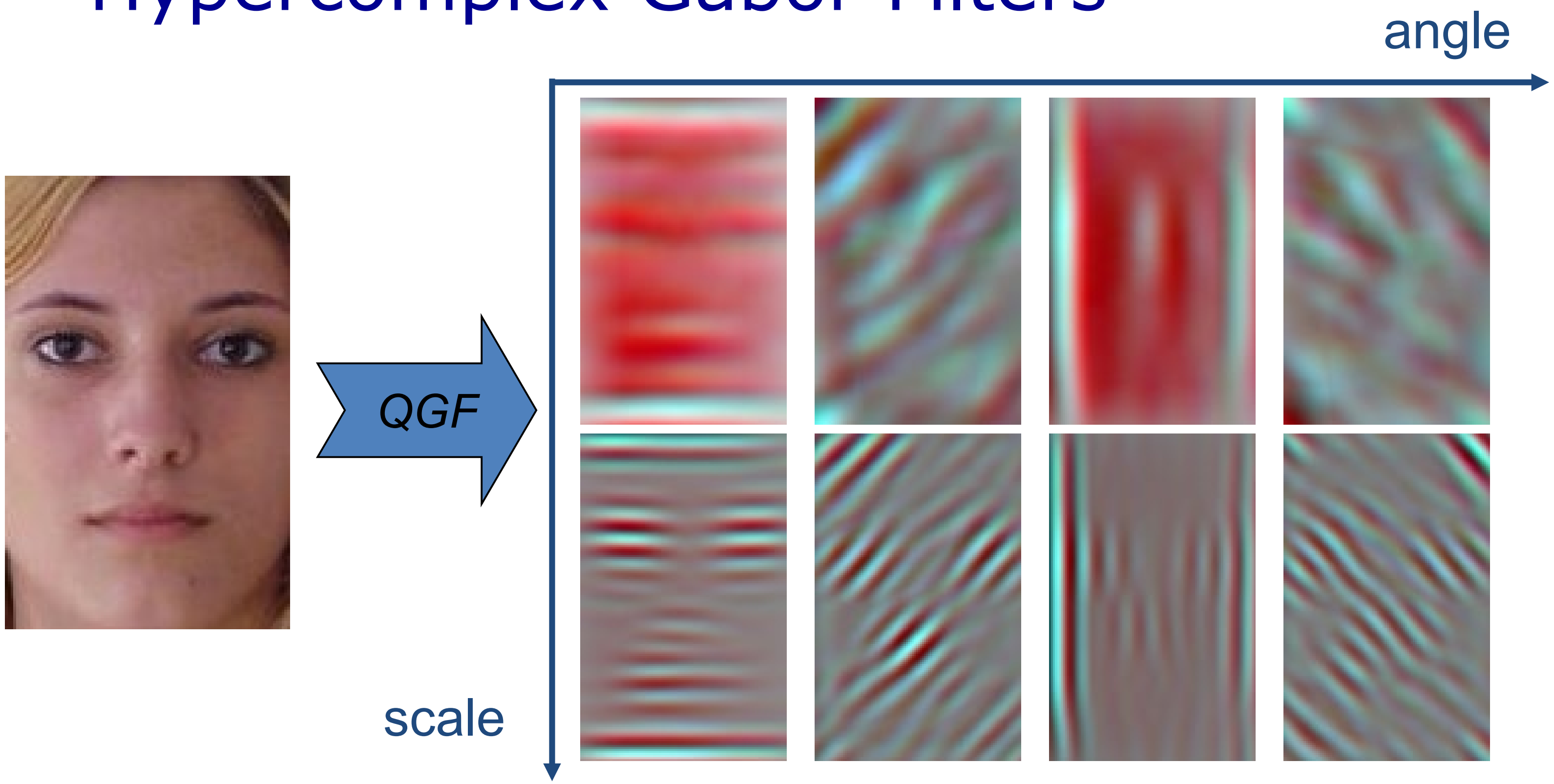
Quaternionic Gabor Filters

- Gabor filters (actually wavelets) are often used to extract face features
  - As in the Elastic Bunch Graph methods
- We have extended them to color by using quaternions (hypercomplex numbers)
- Applying the QGF yields a feature vector called a *jet*



$$\underline{\phi}_{\alpha,\beta}(x,y)=\frac{\left\|\lambda_{\alpha,\beta}\right\|^2}{\sigma^2}\left(e^{-\frac{\left\|\lambda_{\alpha,\beta}\right\|^2\left(x^2+y^2\right)}{2\sigma^2}}\right)\left(\cos\left(\lambda_{\alpha,\beta}\cdot[x,y]\right)+\underline{\mu}\sin\left(\lambda_{\alpha,\beta}\cdot[x,y]\right)-e^{-\frac{\sigma^2}{2}}\right)$$

Hypercomplex Gabor Filters



- Side-by-side implementation of monochrome and color methods shows an increase in accuracy due to color
- Application of the Quaternionic Gabor Filter (QGF) improves differentiation of faces of different individuals

Other features for face recognition

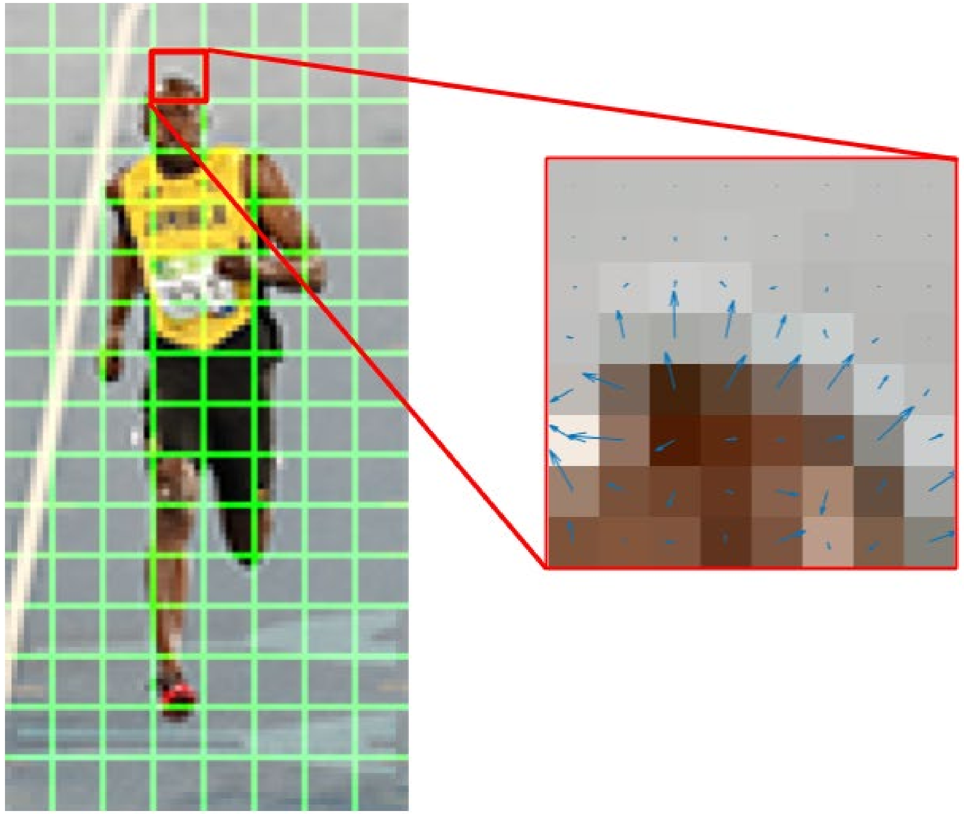
- The task is to find properties of the face that are stable as a person changes appearance, expression, normal activities and ages
  - Yet, are unique to the individual

- We don’t want features that have to be hand-selected
  - “distance from nose to eyes”, for example

- HoG (histogram of oriented gradients) at keypoints
- SIFT and FAST features
- distance between feature locations!

- Often, input to a machine learning model (but there are problems)

Histogram of Oriented Gradients



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

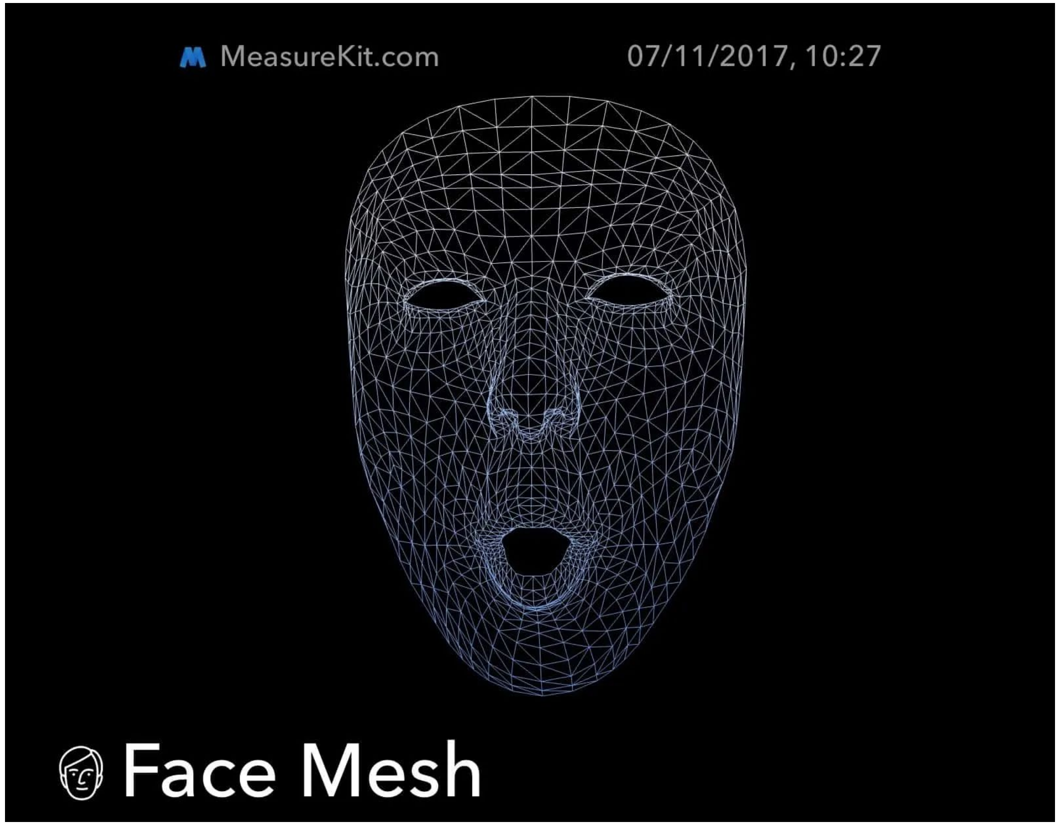
0	20	40	60	80	100	120	140	160
2	2			2				

Histogram of Gradients

3D FACE RECOGNITION METHODS

Apple FaceID uses 3-dimensional images of the face

- A depth map (3D) and an infrared intensity image are collected
- A machine learning model is used to compare to all stored FaceID templates
- Accuracy and robustness are “pretty good”



https://support.apple.com/en-us/HT208108
https://www.cultofmac.com/512671/iphone-x-face-id-views-your-face-mesh/

PERFORMANCE METRICS

FAR and FRR

<ins>False Acceptance Rate (FAR)</ins>

- Percentage of times system erroneously allows an imposter to claim someone else's identity
- False acceptance error can occur in either 1:many or 1:1 search operations

<ins>False Rejection Rate (FRR)</ins>

- Percentage of times the system falsely rejects the identity of a person who really is who they claim to be
- False acceptance errors occur <ins>only</ins> in the context of 1:1 search operations

Sensitivity and Specificity

	Positive Condition	Negative Condition
Positive Test	"True Positive" – Patient has disease, test says YES	"False Positive" (Type I Error) – Patient is well, test says YES
Negative Test	"False Negative" (Type II Error) – Patient has disease, test says NO	"True Negative" – Patient is well, test says NO

There are several measures for quantifying performance of a test (or similar systems)

- counts of True and False Positives and Negatives
- Sensitivity $= \frac{\sum True\ Positive}{\sum True\ Positive + \sum False\ Negative}$; higher is better
- Specificity $= \frac{\sum True\ Negative}{\sum False\ Positive + \sum True\ Negative}$; higher is better

Failure to Enroll (FTE)

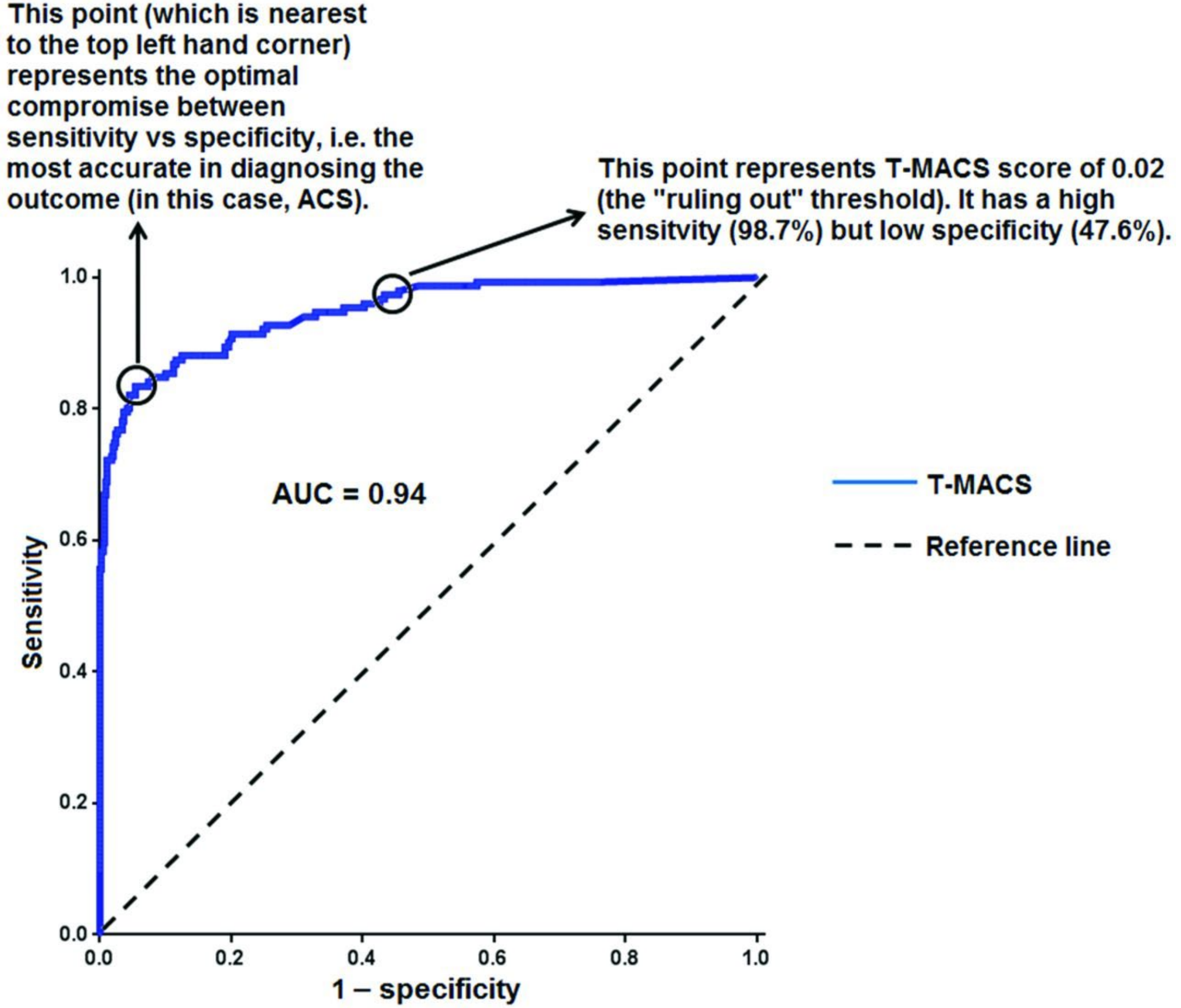
- Some people's biometrics can't be measured
  - Facial features
    - Glasses / no glasses, Facial hair, Tattoos
  - Fingerprints
    - Age, Occupation, Amputations
  - Iris & Retina
    - Disease (e.g., glaucoma, diabetes, and alcoholism), Intoxication
  - Hand Geometry
    - Amputation, Disease (e.g., arthritis)
  - Handwriting
    - Injury, Disease (e.g., Parkinson's disease)
  - Speech
    - Disease (e.g., cold), ambient noise
  - Signature (performance-based)

A Receiver Operating Characteristic (ROC) curve is a graph of True Positive Rate vs. False Positive Rate for different thresholds; a great system will have an ROC curve in the upper left corner

Two commonly used single-number metrics for system accuracy are drawn from the ROC curve:

- Equal-Error Rate EER – the point where TPR = 1-FPR
- Area under the ROC curve AUROC

The ROC and its derived metrics are used for all sorts of classification systems



SOME ISSUES IN FACE RECOGNITION

Bias is present in some systems – how can it be mitigated?

- Any trained system is dependent on the data used to train it
- For many years, research face databases had little representation of non-Caucasian ethnic groups
  - So accuracy on people of color is poor
  - Complicating this, images of people of color often have lower contrast
- Overzealous investigators sometimes muddle technical results with other identity factors
  - Brandon Mayfield
- Religious headwear, etc.

Where should face recognition be allowed?

- Only on arrested suspects?
- By court order?
- Given *probable cause?*
- Based on employment?
- As a condition of granting a driver license?
- In public, if a sign gives warning?
- Anytime, anywhere, as long as it’s live?
- On any video, anywhere?
- Children/juveniles?

A BRIEF REVIEW

ECE5554 SU22 Daily Schedule			
Module	**Day**	**Lec**	**Topics**
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

VIRGINIA TECH
ece
BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

ECE5554 Course Objectives; what skills should you acquire from this course?

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

ECE5554 SU22 Daily Schedule			
Module	Day	Lec	Topics
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

VIRGINIA TECH.

ece
BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

VIRGINIA TECH.

ECE5554 SU22 Daily Schedule			
Module	**Day**	**Lec**	**Topics**
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

Module	Day	Lec	Topics
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

Table title: **ECE5554 SU22 Daily Schedule**

VIRGINIA TECH.

ece BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

ECE5554 SU22 Daily Schedule			
Module	**Day**	**Lec**	**Topics**
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

VIRGINIA TECH.

ece BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

ECE5554 SU22 Daily Schedule			
Module	**Day**	**Lec**	**Topics**
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

VIRGINIA TECH.

ece BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

ECE5554 SU22 Daily Schedule			
Module	**Day**	**Lec**	**Topics**
1 - Introduction; Image Processing	6-Jul	1	Introduction; image formation; coordinate transformations
	11-Jul	2	Pixel operations; filtering; edge detection
	13-Jul	3	Edges; Area-based matching; interpolation; image pyramids
2 - Features	18-Jul	4	Fourier transforms; corner detection; Hough and SIFT
	20-Jul	5	Motion tracking; optical flow; texture
	25-Jul	6	Contours; curvature; region properties
3 - Segmentation	27-Jul	7	Segmentation methods
	1-Aug	8	Graph methods; active contours; color quantization
4 - Structure	3-Aug	9	Feature-based alignment; image stitching; image morphing
	8-Aug	10	Pose estimation; stereo vision
	10-Aug	11	Shape from shading and motion; deep learning; face recognition

VIRGINIA TECH.

ece BRADLEY DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Upon successful completion of this course, students will be able to:

1. contrast common image formation models;
2. implement various ways of extracting features from images;
3. segment an image into meaningful regions;
4. derive the theory behind multi-view geometry;
5. implement various approaches to recognizing objects and scenes in images; and
6. implement techniques for processing video sequences.

VIRGINIA TECH.

Today’s Objectives

Face Recognition

- Concept and Terms
- Methods
  - Eigenface
  - Local Feature Analysis
  - Elastic Bunch Graph Matching
  - Newer Methods
- Performance Measurement
- Current Issues
  - Surveillance ethics
  - Bias

- A Brief Review