COMP 102: Computers and Computing
Lecture 23: Computer Vision

Instructor: Kaleem Siddiqi (siddiqi@cim.mcgill.ca)

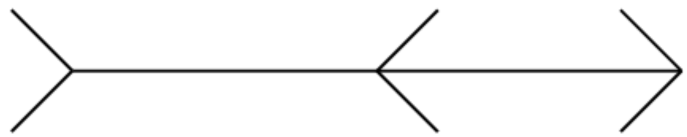
Class web page: www.cim.mcgill.ca/~siddiqi/102.html

What is computer vision?

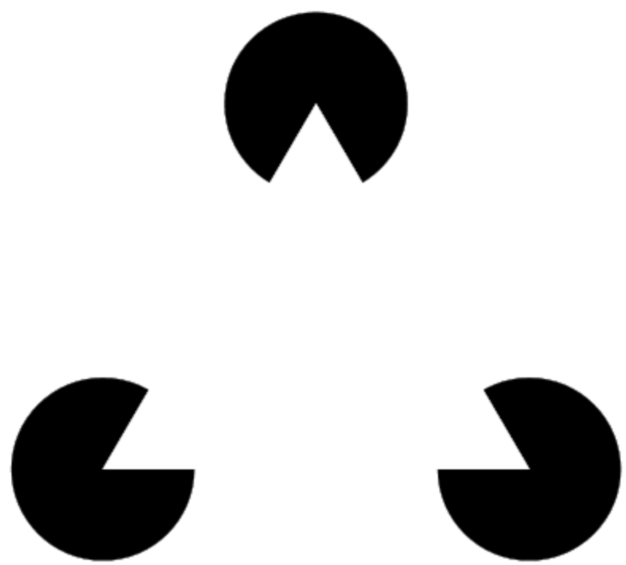
- Broadly speaking, it has to do with making a computer “see”.
- The complexity of this task is enormous!
- It is about understanding visual input and making inferences about the visual world.
- What you see is NOT necessarily what is there!

Visual illusions

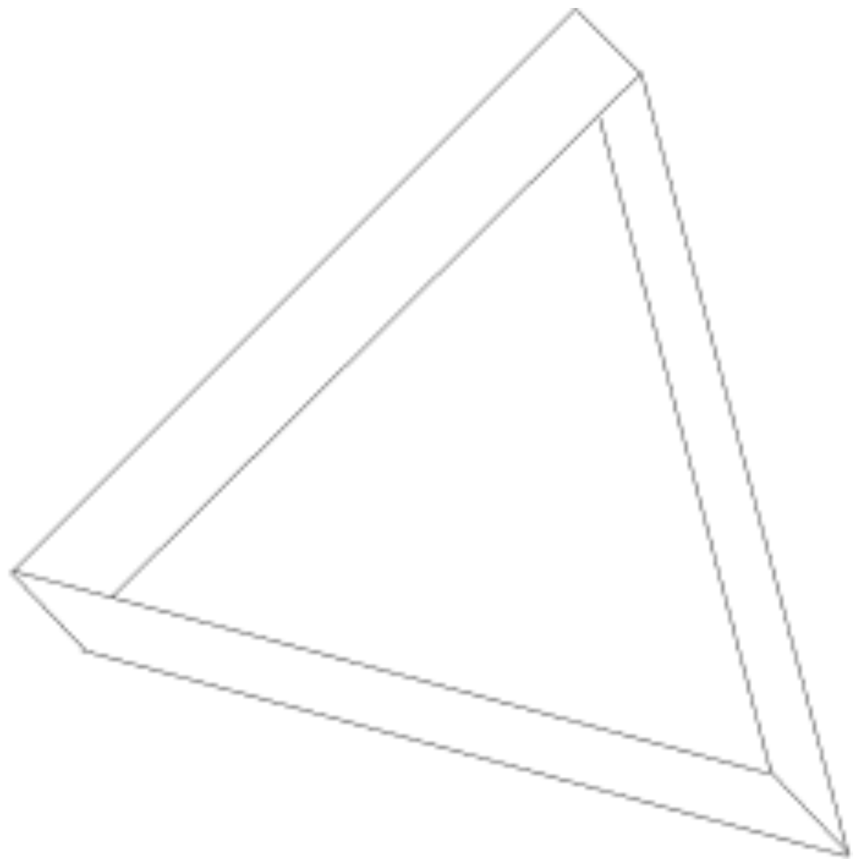
The Mueller–Lyer illusion



The Kanizsa Triangle

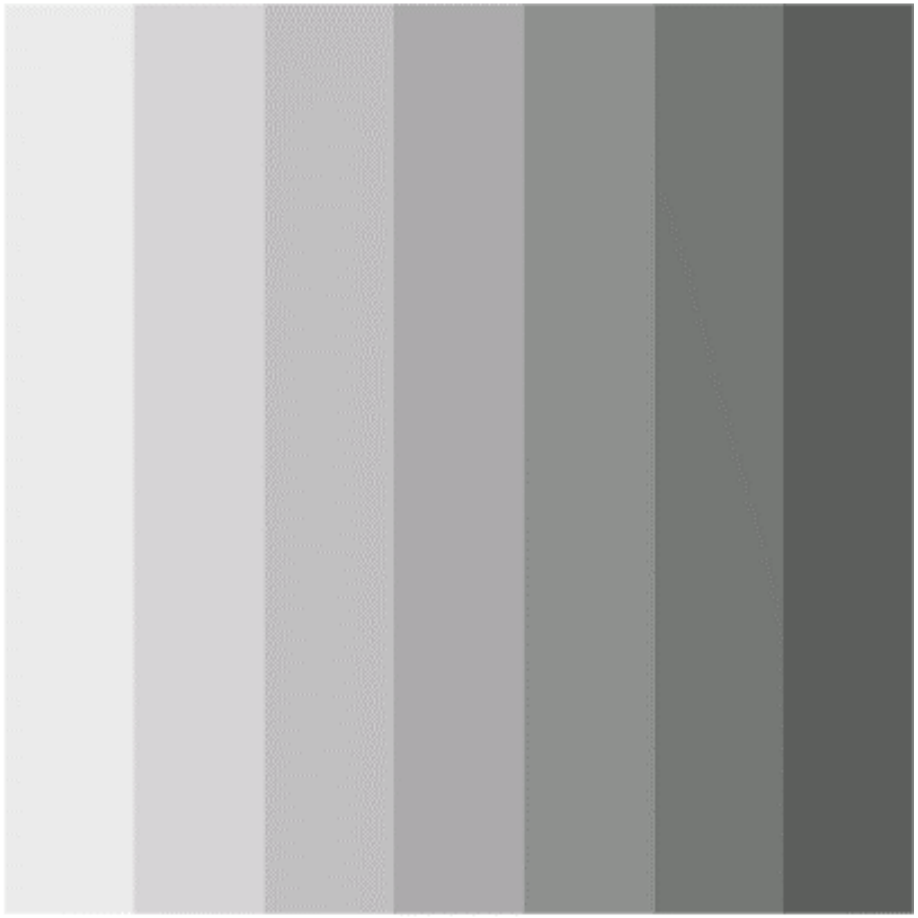


Visual illusions



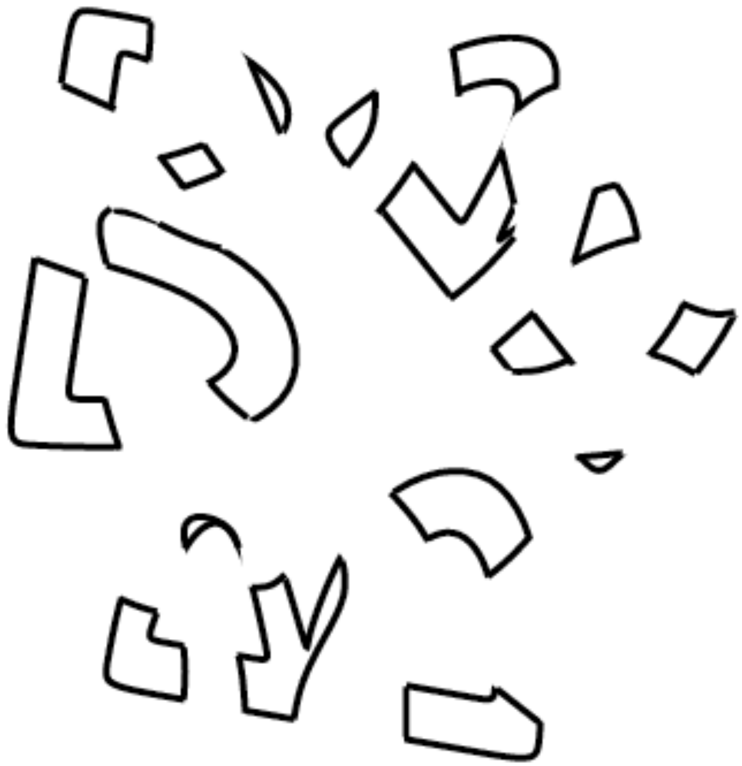
An “impossible” object

Visual illusions



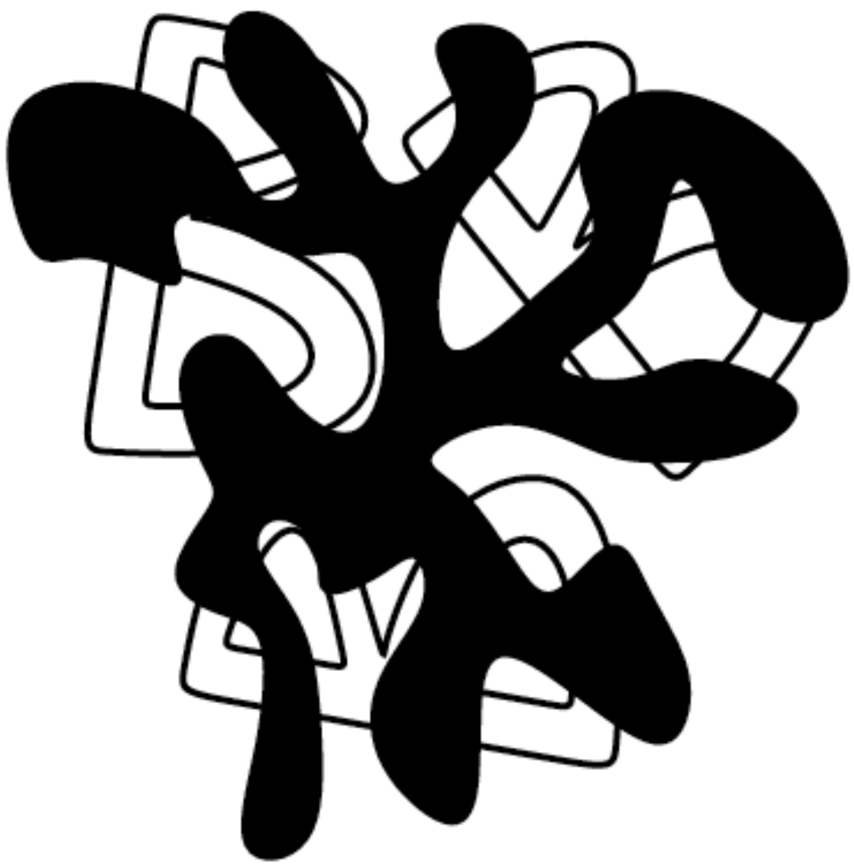
Mach bands

What do you see?



Is it clearer now?

Bregman B's



Why study computer vision?

- Scene reconstruction
- Object recognition and identification
- Object modeling
- Robot navigation
- Medical imaging
- Image-guided neurosurgery

3 Problems we will look at

- Segmentation using deformable models

  (moving curves and surfaces)
- Face Recognition
- Stereo Vision

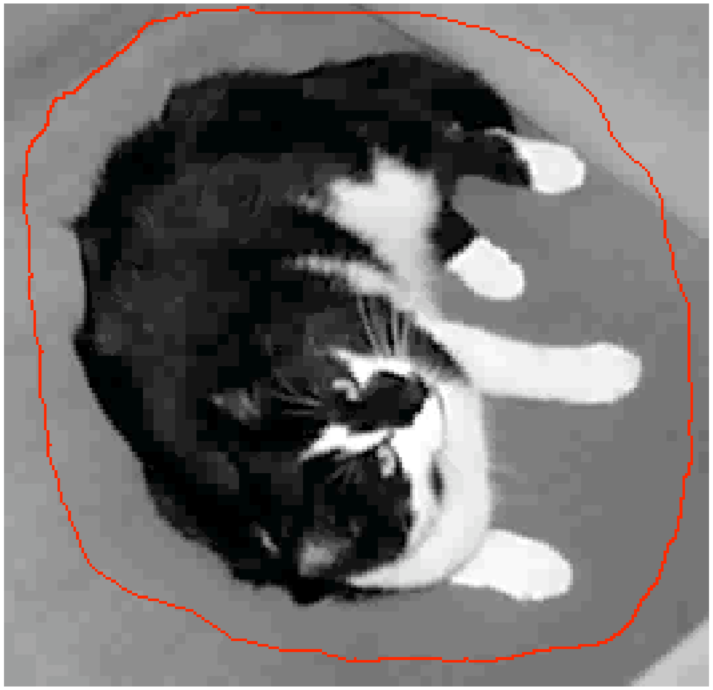
Problem 1: Segmentation

- Segmentation has to do with processing visual images to extract the outlines of objects of interest.
- This is a difficult problem for a computer, because it is a challenge to come up with a precise definition of an outline that generalizes to many classes of objects.
- In an image changes in contrast often signal object boundaries, but they can also arise due to other cues (shadow boundaries, texture, surface discontinuities, etc.).

A simpler scenario

- Assume that an object is on a homogeneous background
- Initialize a curve (a list of discrete points) and define a process by which these points move in the image, until they reach a place of high contrast, where they slow down.
- In the limit, this type of shrink wrap process can wrap around the outline of an object.
- This scenario can be modified to handle objects with a particular distinctive texture (e.g. a tissue class). In this case the stopping criterion takes into account the statistics of the enclosed region.

An example



Segmentation

- The previous process can also be carried out in 3D, where the evolving structure is no a surface.
- It deforms in 3D so as to wrap around the boundaries of 3D structures of interest, such as anatomical structures or tumours in an MRI scan.
- Initialize a curve (a list of discrete points) and define a process by which these points move in the image, until they reach a place of high contrast, where they slow down.
- In the limit, this type of shrink wrap process can wrap around the outline of an object.
- This scenario can be modified to handle objects with a particular distinctive texture (e.g. a tissue class). In this case the stopping criterion takes into account the statistics of the enclosed region.

Problem 2: Face Recognition

- Imagine that one has a very large database of mugshots of individuals which one wishes to be able to recognize.
- Given a query image of a face, the goal is to identify the person by comparing the query against the images in the database and finding the closest match.
- A difficulty with this strategy is that a face can have may different appearances, dependng on pose, facial expression,etc.
- A second difficulty is that it is simply too computationally expensive to compare the query, pixel by pixel, with all the images in a large database. It just takes too long!

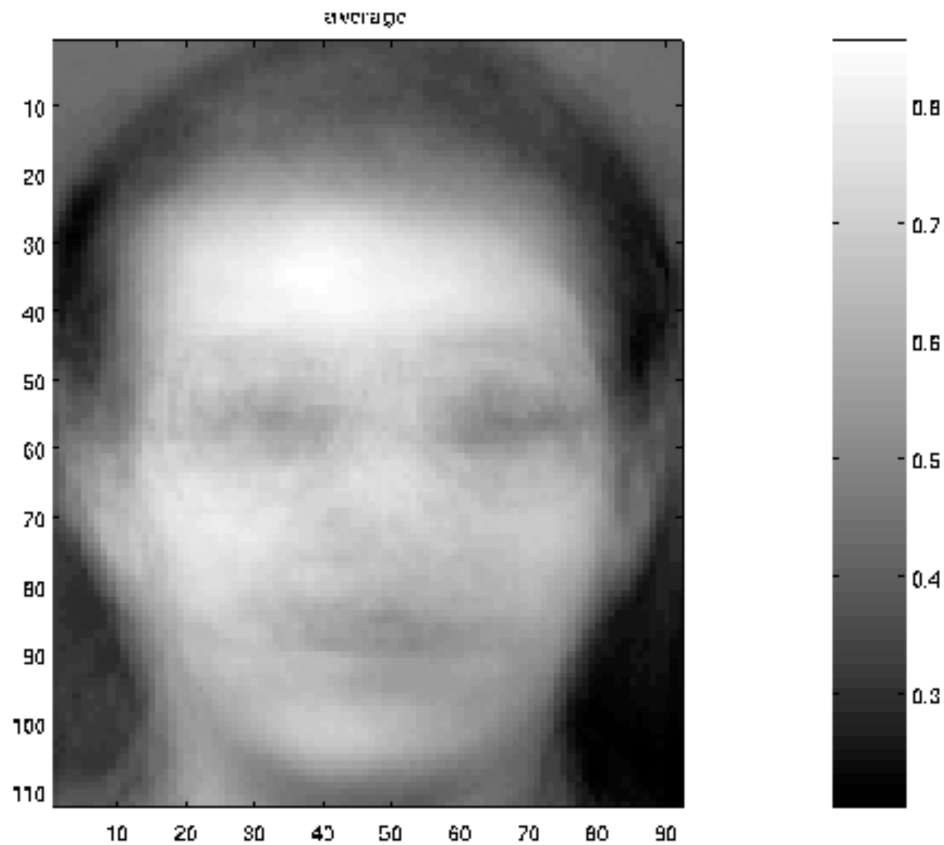
A simpler scenario

- Assume that faces are seen from a particular viewpoint (e.g. frontal views) and that for each person one wishes to recognize there are many candidate pictures a system can use for training.
- Given such a large database, imagine that there is a special set of “derived” images, called “principal components”, PC1, PC2, PC3, … with the following property: a candidate image is the average of all the images in the database, plus a weighted sum of the first few principal components.
- The representation is then just a vector of weights.

An example

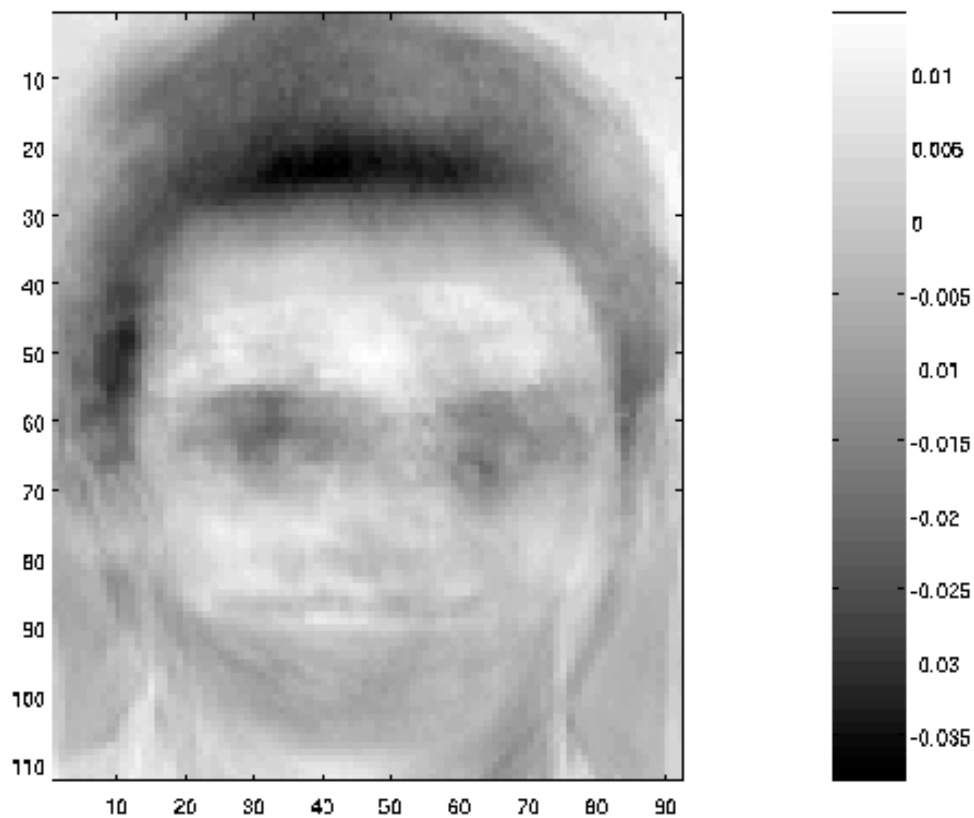
Some pictures from a face database containing 40 pictures:
10 individuals, 4 pictures per individual.

Face Recognition



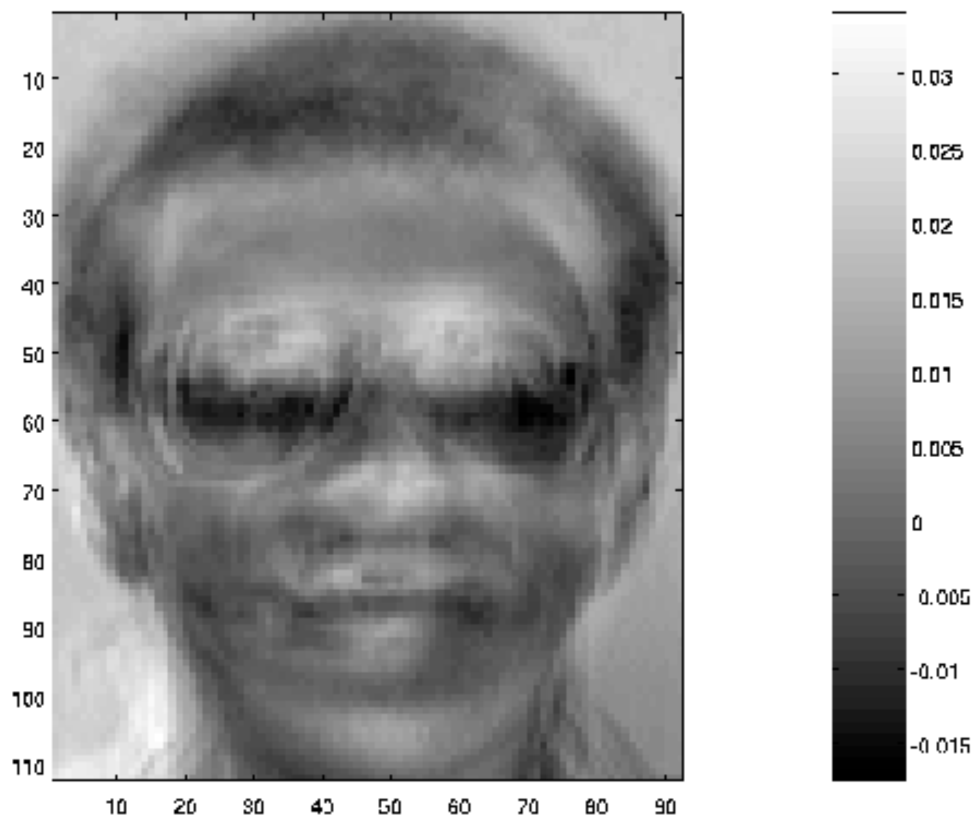
Average of the 40 pictures in the database

Face Recognition



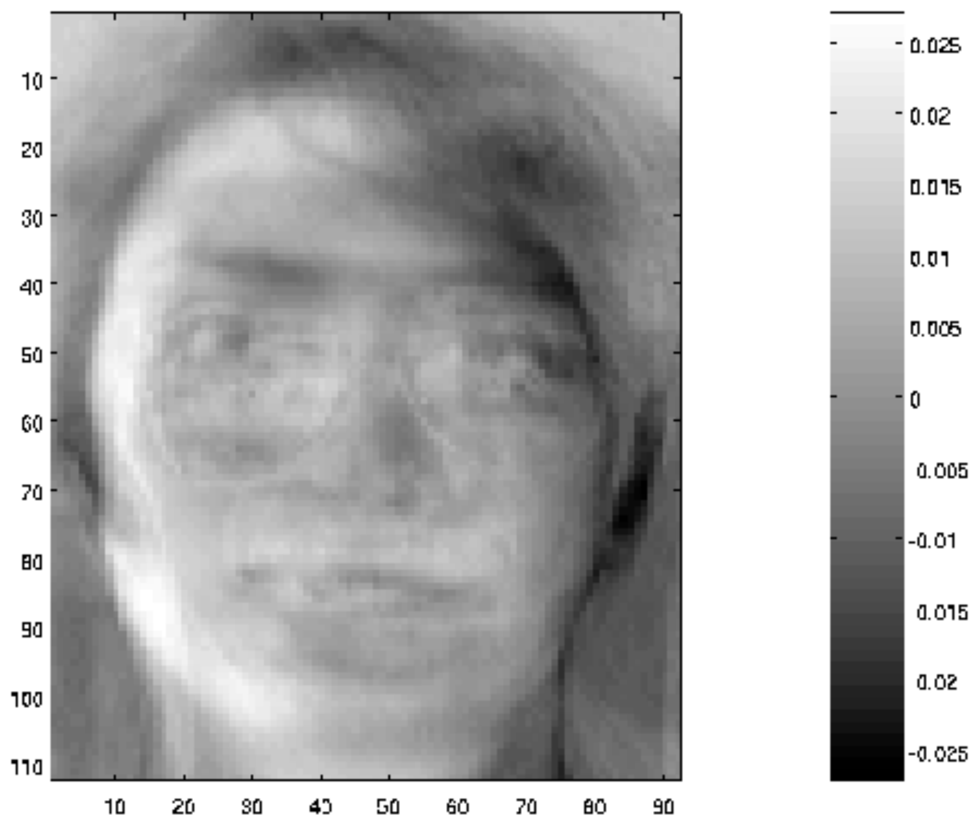
First principal component

Face Recognition



Second principal component

Face Recognition



Third principal component

Eigenspace representation

Query image (not in database)

Approximation using the first 10 principal components

How does a face get recognized?

- The query image gets represented by a weighted combination of the first few principal components.
- Those weights are then compared against the weights that represent each image in the database.
- This is much cheaper because we are basicall comparing a small vector with another small vector (rather than two possibly mega-pixel sized images).
- The identity assigned to the query image is then that of the closest image found in the database.

Problem 3: Stereo

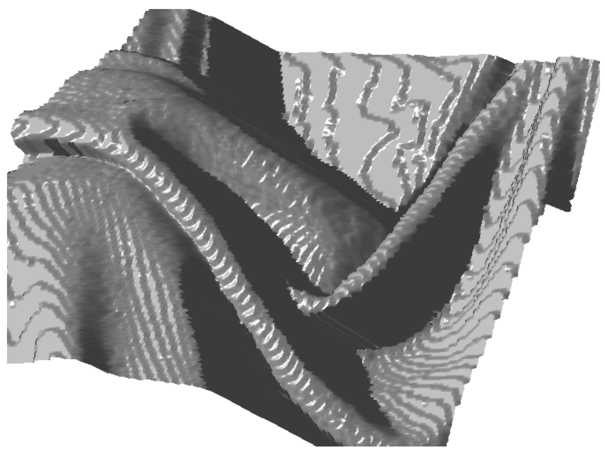
- How does one perceived depth (3D) from a pair of stereo images, each of which is 2D?
- The human visual system, and that of most primates, exploits the notion of "disparity". Given a fixed point in the 3D world, the location at which it is projected onto the left and right images in a stereo pair differs slightly. This difference is called disparity.
- The amount of disparity is inversely proportional to 3D depth.
- As a simple demonstration, hold your thumb out in front of your eyes and open one eye at a time (shutting the other eye when the first is open). The thumb will appear to "jump" from left to right. The amount of this jump decreases as you increase the distance of your thumb from your eye.

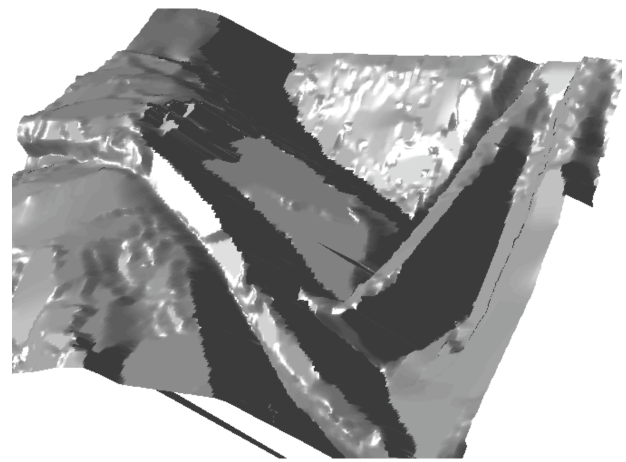
Correspondence and Reconstruction

- Given a fixed camera system, with a left image and a right image, stereo breaks down into two problems.
- The first is the “correspondence” problem, which is the harder problem, and it has to do with associating points in one image with their corresponding points in the other. The notion of correspondence here is that the same physical point in the 3D world gave rise to these two projected points, one in each image. Often this problem is not well defined, e.g., imagine looking at a regular texture or at a homogeneous region, where a point in one image can correspond to many points in the other.
- Assuming though that there are distinctive features, once a correspondence has been found, the 3D point it came from can be constructed by just tracing a ray back from each camera center, through the left (or corresponding right) image point and then noting where these two rays intersect. This is called the “reconstruction” problem.

Correspondence and Reconstruction

- Modern day stereo algorithms for rigid scenes are quite powerful.
- Given a fixed camera system, they can generate dense realistic 3D depth maps from left-right image pairs, such as the following





One image (from a stereo pair) — ground truth surface — reconstructed surface