ECSE-626

Statistical Computer Vision

Object Recognition

Object Recognition Problem

- Object recognition has been one of central problems addressed in field of computer vision.
- Most common approach is to acquire database of known objects off-line during training.
- On-line, sensor measurements (e.g. from a camera) of an unknown object are acquired.

Inverse Problem

- Given a set of sensor measurements of an object in a scene (e.g. from a camera), determine the object in the database that generated those measurements.

Object Recognition Problem

Generate algorithm to get a computer recognize an object placed in front of it *in any orientation* from a single 2D image.

What is this?

Database of objects

Which 3D object in the database gave rise to this image?

Object Recognition Problem

Is this problem *well-posed*?

Recognition - Ill-posed Problem

This problem is not well posed primarily due to the violation of the uniqueness constraint.

Uniqueness: Several objects can give rise to the same measurement.

Recognition - Ill-posed Problem

What is this?

Viewpoints where impossible to pick one solution!

Object Recognition Problem

- As a result, it is not possible to identify an unknown object uniquely and several solutions may be equally viable.
- Imposing a closed-world assumption (i.e. only database objects can exist) helps to constrain the solution space.
- However, requiring a single object identity to be chosen can lead to instability in the solution – violation of condition of continuity.

Object Recognition Problem

- In addition, the solution process can be ill-conditioned (i.e. in terms of robustness against noise). Experimental uncertainty gives rise to uncertain measurements.

Object Recognition Solutions

- In general, shortcomings in early solutions to the problem of object recognition can be categorized as follows:

Object Recognition Solutions

1. **Shortage of general solutions**:

   There are various ways to condition the ill-posedness but these require strong, often implicit, *a priori* assumptions about the nature of the world.

   As a result, a method may work well in specific cases but, because of the hidden, implicit nature of the conditioning assumptions, is not portable to other contexts

Object Recognition Solutions

2. **Deterministic strategies**:

Uncertainties are not explicitly represented in the solution.

Solutions return a single object label from a set of measurements, even with the possibility of ambiguous or erroneous results.

No way to assess the validity of the result.

Object Recognition Solutions

3. **Most strategies are static**:

   Most approaches focus on the problem of identifying an object based on data acquired from a single viewpoint in a scene.

   Usually results based on a single image are not sufficient.

Classification

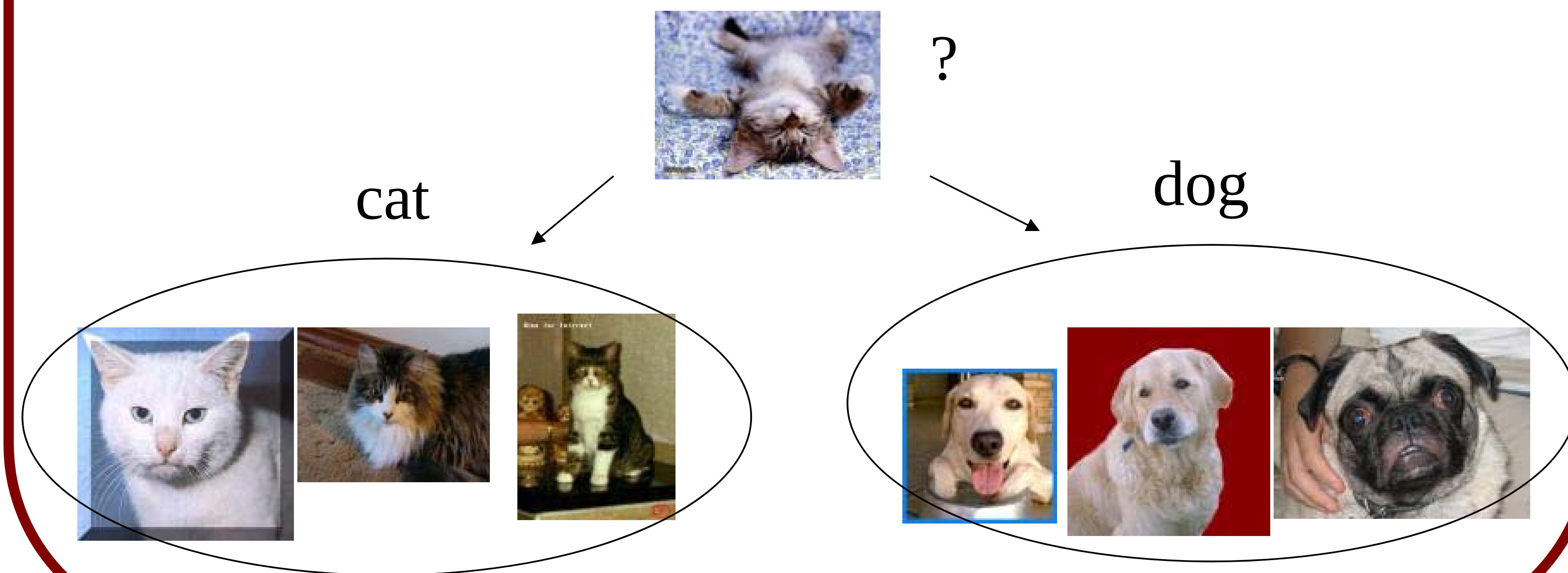
- A more general problem is that of classification: Camera measurements of objects of different classes are acquired off-line during training.

cat

dog

Classification

- Given a new image, which class does this new object belong to?



Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Aspect Graphs

- Many early schemes involved substantial amount of on-line computations due, in part, to model analysis required for each model in the database.
- A need arose to reduce the expense of on-line computations, with much of the database processing performed off-line.

Aspect Graphs

- Schemes emerged to attempt to perform much of the “precompiling” prior to recognition, thus improving the efficiency of the task at run-time.

Aspect Graphs

- The idea for recognizing 3D objects is to:
  - Obtain a series of 2D views of a known object,
  - Maintain them in some convenient representation in storage,
  - Match one or more 2-D views on an unknown object against the stored views of the known object.
- This reduces the 3D matching problem to a series of 2D problems.

Aspect Graphs

- How can we compute and store 2D views of a known object?
- One of the first schemes to move computations off-line involved representation called a*spect graphs*, initially described by Koenderink and van Doorn J. J.[1]

[1] **Koenderink and A. J. van Doorn** , The singularilarities of the visual mapping, **Biol. Cyber.**, 24:51--59, 1976

Aspect Graphs

- Objects are represented by an *aspect graph*:
  - Each **node** represents a *stable* viewpoint of an object from which a particular set of local features are visible.
  - Stored with each node is enough information to allow reconstruction of the view.

For a survey: http://cis.poly.edu/tr/tr-cis-2001-01.pdf

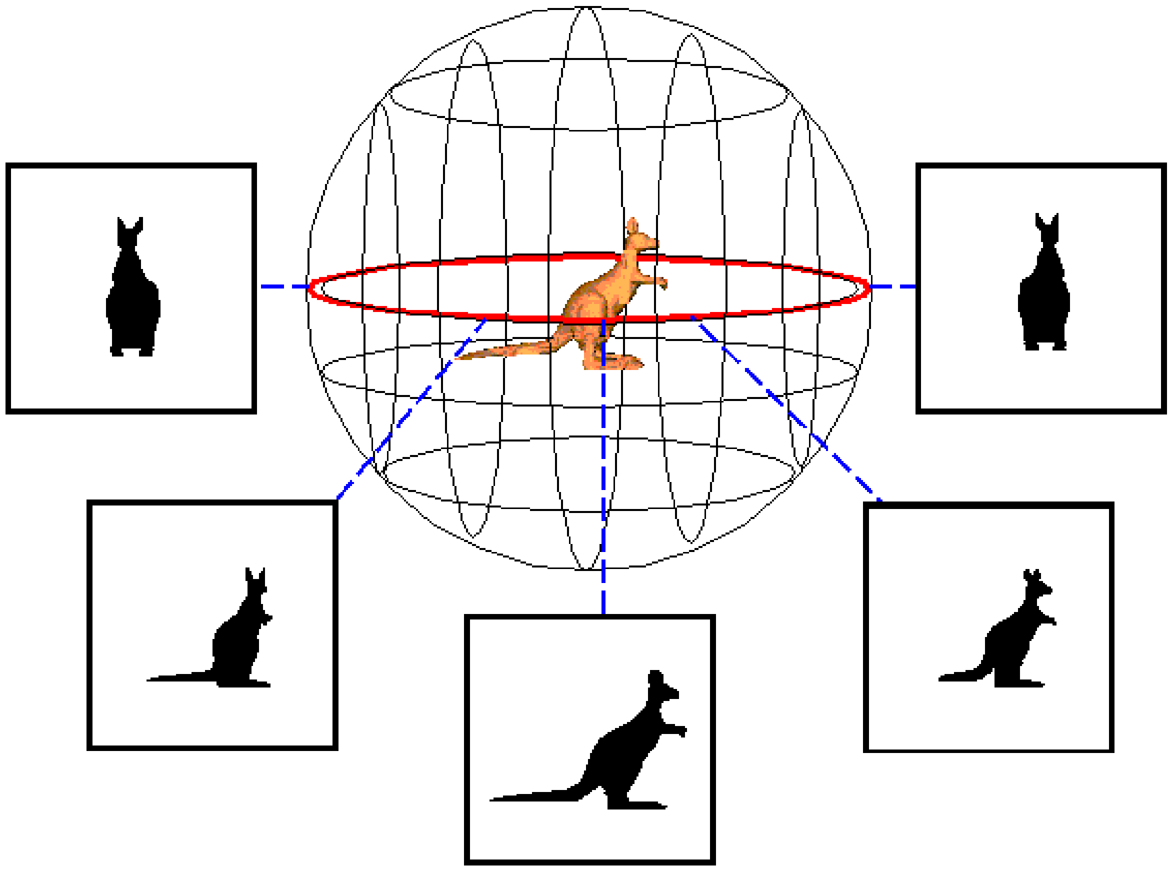
Aspect Graphs

– **Edges** represent transitions (*visual events*) between viewpoints.

This is where adjacency information is stored – 2 stable views are said to be *adjacent* if a moving viewer can pass from one view to another with no intervening stable views.

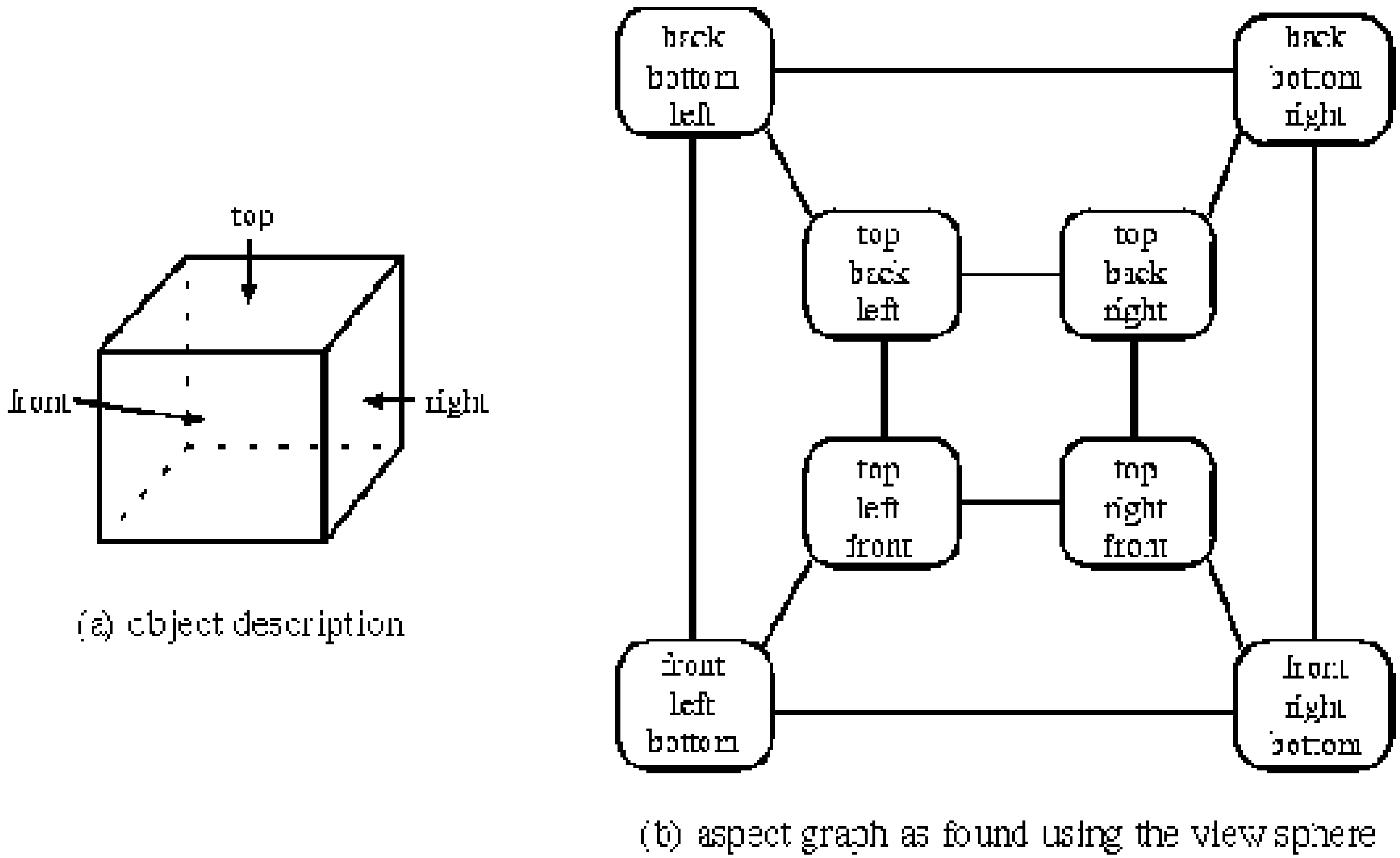
– Essentially, the graph divides the viewsphere into stable regions defining *characteristic views*, where small changes in viewing position don't affect topological structure of set of visible features.

Aspect Graphs



http://www.lems.brown.edu/~cmc/3DRecog/aspect.html

Aspect Graphs



http://www.lems.brown.edu/~cmc/3DRecog/aspect.html

Aspect Graphs

- Pre-computing aspect graph for each model in the database improved efficiency of recognition task at run-time.
- On-line, recognition can be performed by matching an input image of the object into the database of views formed by the graph for an object.
- Worked well for convex polyhedral objects. Features consisted of edges, corners.

Aspect Graphs

- Problems when attempting to generalize to complex objects in terms of :
  - *High storage requirements*: size of the resulting graphs.
  - *Large construction times.*

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Feature Matching

- Many types of features are used for object representation and recognition:
  – Local features:
    - Boundary segments, Corners, Curvature
    - SIFT, SURF, LBP
  – Global features:
    - Size, color
    - Fourier descriptors
    - Moments

Feature Matching

- Recognition is performed by matching features in the input image to those in the database objects.
- Matching based on many local features provides a certain robustness against occlusion.
- Global features are more robust to changes in viewpoint and tend to be object-centered.

Feature Matching

- Early work focused on subjective notion of features of interest (e.g. corners).
- Later work has led to the development of interest points that are invariant to changes in scale and viewpoints (e.g. SIFT).

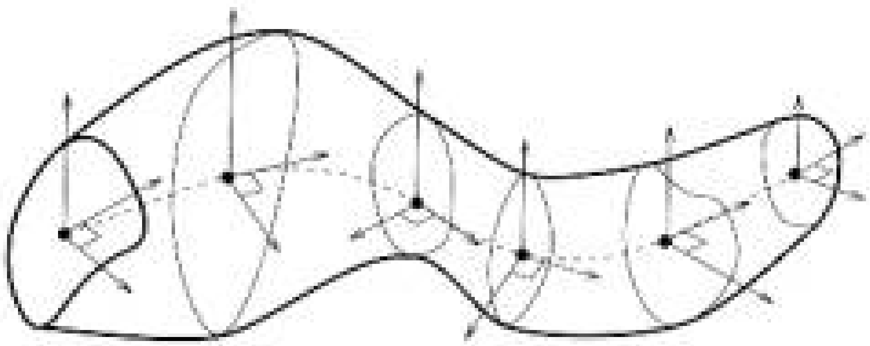
Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Parametric Shape Primitives

- Early on, a popular method of representing 3D volumes is through parametric shape descriptors:
  - Superquadrics
  - Generalized cylinders
  - Deformable solids

Generalized cylinder



Parametric Shape Primitives

- Using parametric models, a wide variety of shapes can be represented from measurements (e.g. laser rangefinder data, contours from grayscale image).
- This approach is especially useful if parameters are object-centered.

Parametric Shape Primitives

- Can be used in a parts-based representation, where data are segmented into parts and parametric models fit to each part.
- Parameters of unknown object are then matched to parts of object in database.

Parametric Shape Primitives

- Very few recognition schemes based on volumetric models.
- One reason has been the shortage of stable, efficient bottom-up systems capable of building stable representations for complex objects.
- This is due to the inherent difficulty in each inference stage, especially the segmentation task.

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

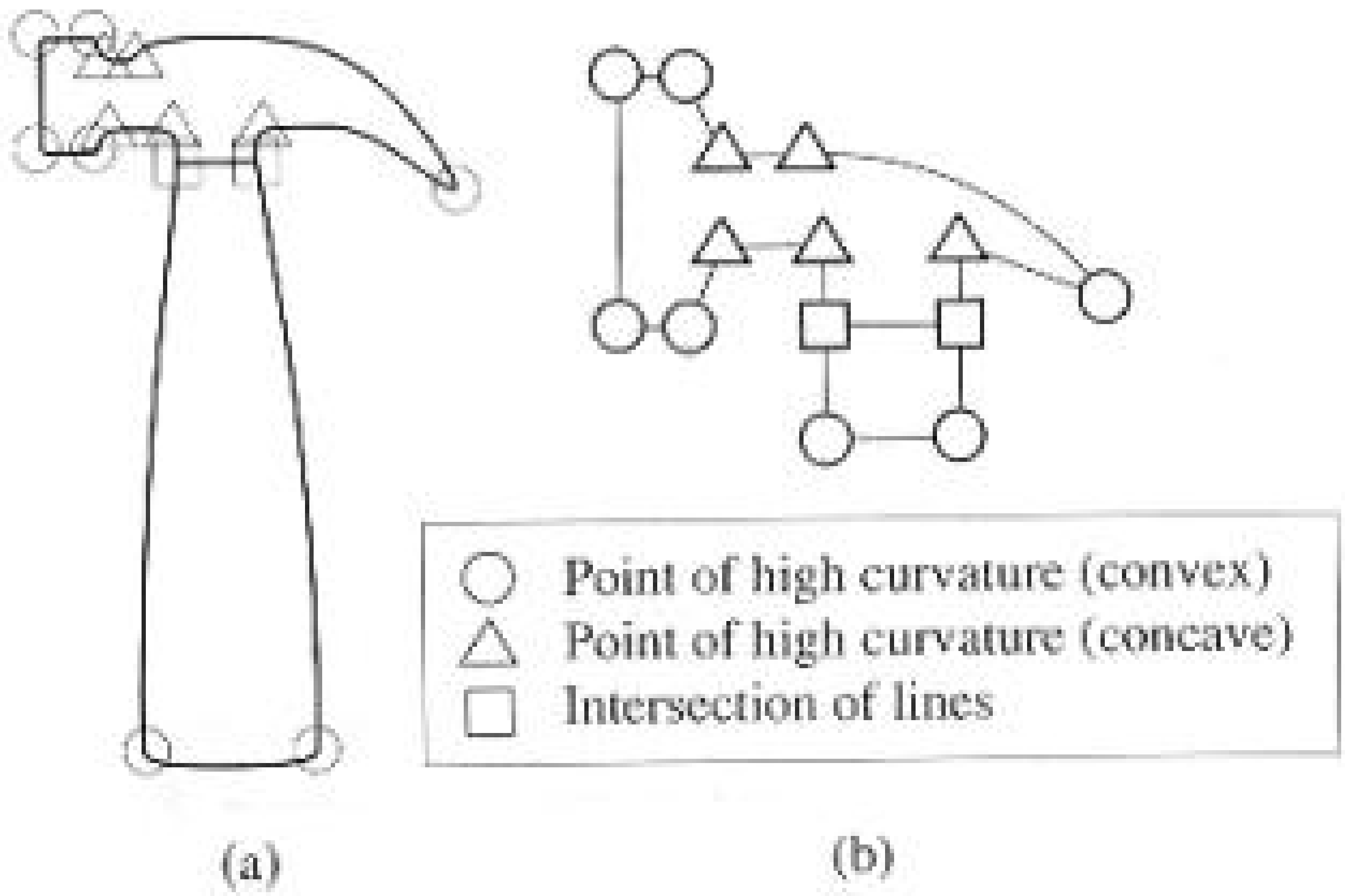
Graph Matching

- Once a set of reliable features are extracted from the images, one can represent objects in the form of a 2D graph:
  - Nodes represent object features.
  - Arcs represent the geometric relationship between object features (e.g. distance, angle).

Graph Matching

- The literature has focused on graphical representation of 2D shapes such as silhouettes of objects.
- Other approaches use medial skeletons and shock graphs.

Graph Matching



http://www.netnam.vn/unescocourse/computervision/84.htm

Graph Matching

- The object recognition problem is then considered as a graph matching problem:

  Given a graphical representation of an object in the scene, find the best match among the database of other graphs.

Graph Matching

Algorithm:

– Given 2 graphs G_1 and G_2 containing nodes Nij, where *i* and *j* denote the graph number and the node number, respectively.

– Let R_{ij} denote the relations between the nodes *i* and *j*.

– Define a similarity measure for the graphs that considers the similarities of all nodes and functions.

Graph Matching

- One can formulate the problem as matching to a sub-graph in the database.
- This permits matching under occlusion.

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Geometric Hashing

- A major problem facing object recognition schemes has been the enormous complexity involved in searching the database to select the possible candidates.
- Many methods have been introduced to reduce the computational complexity.
- One such method has been the *geometric hashing* scheme.

Geometric Hashing

- In these schemes, a hash table containing feature constraints for all the models in the database is constructed.
- Feature measurements are derived from the scene, and the corresponding values are located in the appropriate entry in the table.
- This results in many possible matches, which are resolved by a using set predefined rules.

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Tree-search matching

- One general category of matching schemes has been the *tree-search* approach.
- After object features are extracted, a tree of possible model-to-object feature matches is built.
- Each path from root to leaf represents one possible solution to the correspondence problem.

Tree-search matching

- The idea is to search the path that would ensure a consistent matching between object and model in the database.
- Work has been done in order to reduce the search time. These include constraining:
  - the range of unary feature values (e.g. the length of an edge),
  - the range of binary features describing the inter-relationships between unary features (e.g. the angle between normal vectors).

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Functional Classification

- A different approach to the recognition problem developed, based on the conjecture that objects can be more robustly identified based on their high-level functionality.
- Humans can classify objects even with wide variations in form:
  - Example: A chair is for sitting on.

Functional Classification

Functional Classification

- Functional representations were first developed by Stark and Bowyer[1].
- The goal was to recognize objects by reasoning about their intended function rather than matching to a database of 2D appearances or 3D object shapes.
- Function was extracted from a set of structural features.

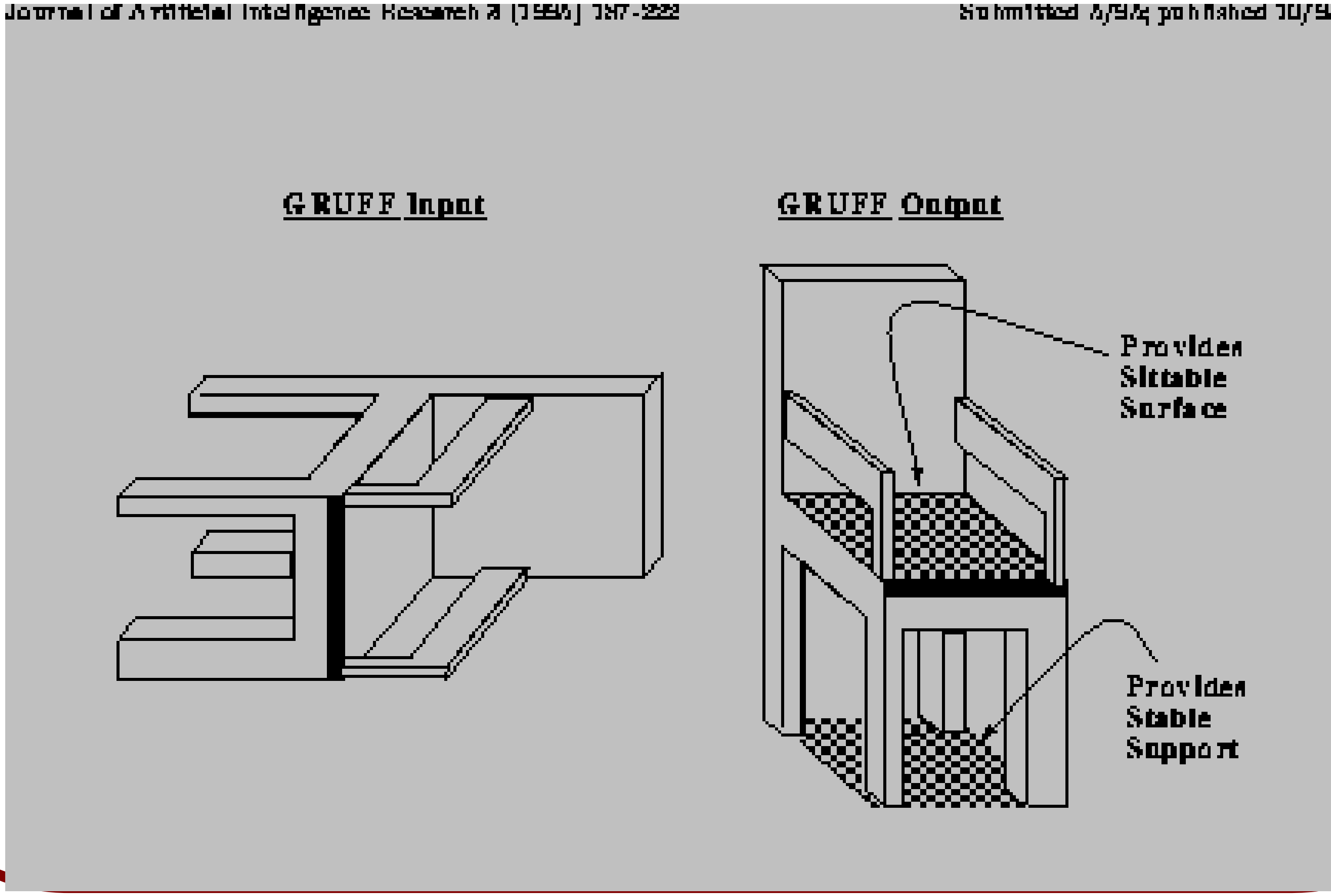
[1] L. Stark and K.W. Bowyer, “Achieving generalized object recognition through reasoning about association of function to structure”, PAMI 13(10), pp. 1097-1104, 1991.

Functional Classification

- This required matching of features and relations between them to function.
- Chair example: Relationship between parts led to high-level functional description of chair based on knowledge primitives:
  – Relative orientation between normals of surfaces
  – Dimensions of surfaces
  – Proximity in terms of qualitative relations between elements (e.g. above, below)
  – Clearance (unobstructed free space relative to a part of shape)
  – Stability (e.g. check that shape is stable when placed on a surface)

Functional Classification

http://seraphim.csee.usf.edu/omlet/node3.html



Functional Classification

- Good for a small set of predefined object classes.
- Generalizing this approach to a wide set of object classes based on function would be difficult in practice.

Traditional Approaches

- Aspect graphs
- Feature matching
- Parametric shape primitives
- Graph matching
- Geometric hashing
- Tree-search matching
- Functional classification
- Appearance-based techniques

Appearance-based Techniques

- Model-based approaches to recognition are difficult in cases where:
  - the scene is complex,
  - the relationship between the data and model less-obvious,
  - the requirements are more general.
- Appearance-based techniques were developed for these cases.

Appearance-based Techniques

- Representations are *viewer-centered* and the data/model relationship made explicit through empirical evidence gathered during training.
- These techniques (Kirby and Sirovich, Nayar and Murase, Turk and Pentland) became popular due to their:
  - simplicity
  - fast indexing time
  - Accuracy in constrained environments
  - Relative insensitivity to small perturbations in the image

Appearance-based Techniques

- Off-line, images of each object in the database are acquired.
- A lower dimensional subspace is build based on the entire set of raw images acquired using Principal Components Analysis (PCA).
- The lower dimensional space in which the images are represented is referred to as an *appearance manifold*.

Appearance-based Techniques

- Recognition or indexing is based on projecting images acquired on-line onto the manifold and finding the closest stored image.
- The major drawback is that tight control over the image formation parameters (e.g. lighting, background) has to be enforced to ensure repeatable appearance.

Application to Robotic Vision Tasks

- One of the first application areas for appearance based recognition was in the area of industrial robotics.
- Many difficult problems in robotics are simplified when effective vision processes are available.

Application to Robotic Vision Tasks

- Some of these problems include:
  - visual position control (visual servoing)
  - visual object tracking (for controlling moving objects)
  - object inspection and recognition
- The appearance-based approaches of Nayar et al have been shown to be effective in the controlled, real-time, conditions of robotics.

(S. K. Nayar, S. A. Nene, and H. Murase, "Subspace Methods for Robot Vision," CUCS-06-95, Technical Report, Department of Computer Science, Columbia University, New York, February 1995)

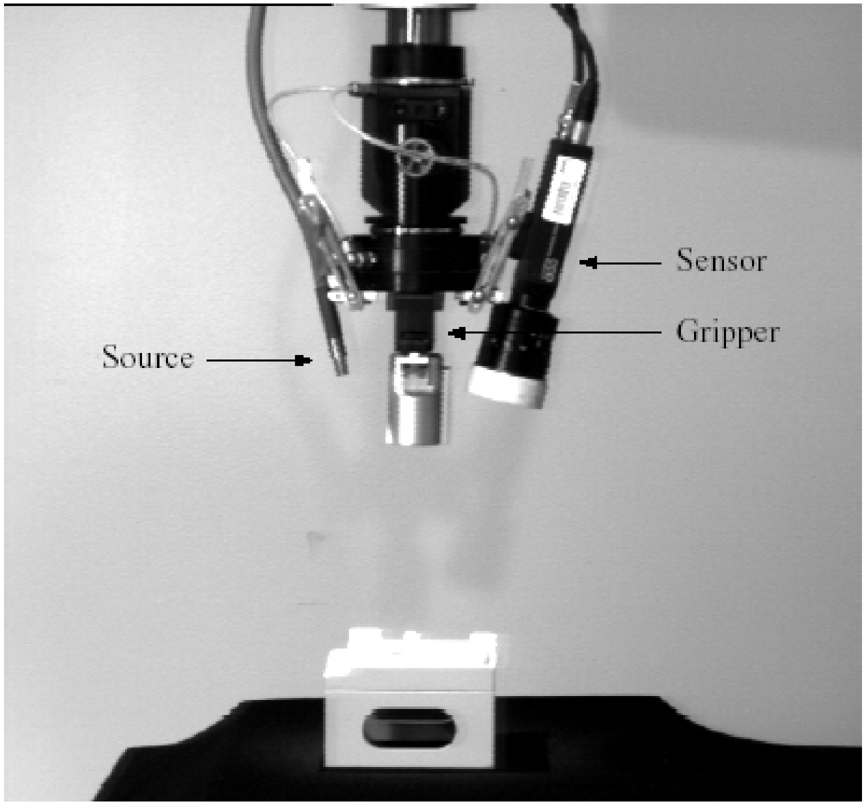


Figure 1: The hand-eye system used for visual positioning, tracking and inspection. The end-effector includes a gripper and an image sensor. In applications where ambient lighting is not diffuse, a light source may be mounted on the end-effector to achieve illumination invariance.

Application to Robotic Vision Tasks

- Nayar et. al. acquire images of an object over a sampling of positions in the *workspace* of a robotic manipulator, where the camera is fixed to the manipulator.
- A low-dimensional eigenspace is created from the principal components of all of these images.

Application to Robotic Vision Tasks

- Since the images of the object taken from neighboring points in the manipulator workspace tend to be highly correlated, they will tend to have neighboring representations in the eigenspace as well.
- Thus there is usually a smooth relationship between spatial coordinates in the robot's manipulator workspace and the eigenspace coordinates.

Application to Robotic Vision Tasks

- This relationship between the visual eigenspace and the manipulator workspace is very useful for *visual servoing* tasks.
- The relationship is easily computed.
- No feature extraction is needed, nor does the camera need to be calibrated.

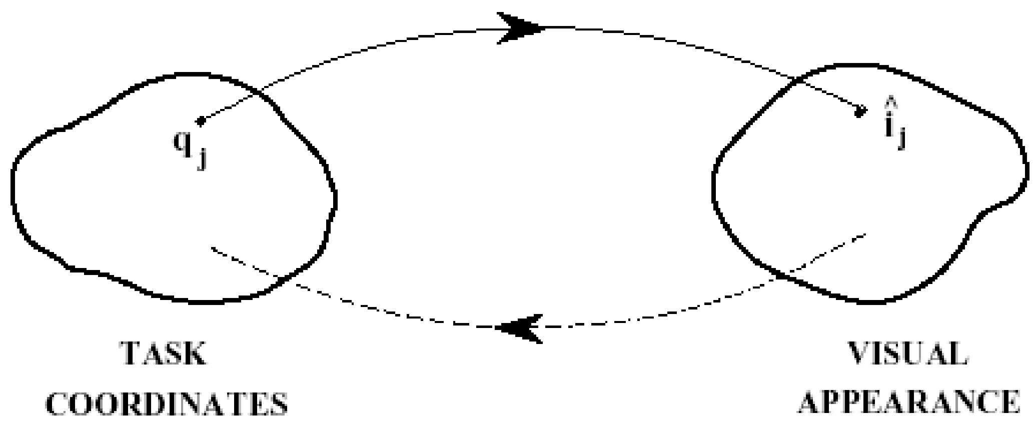
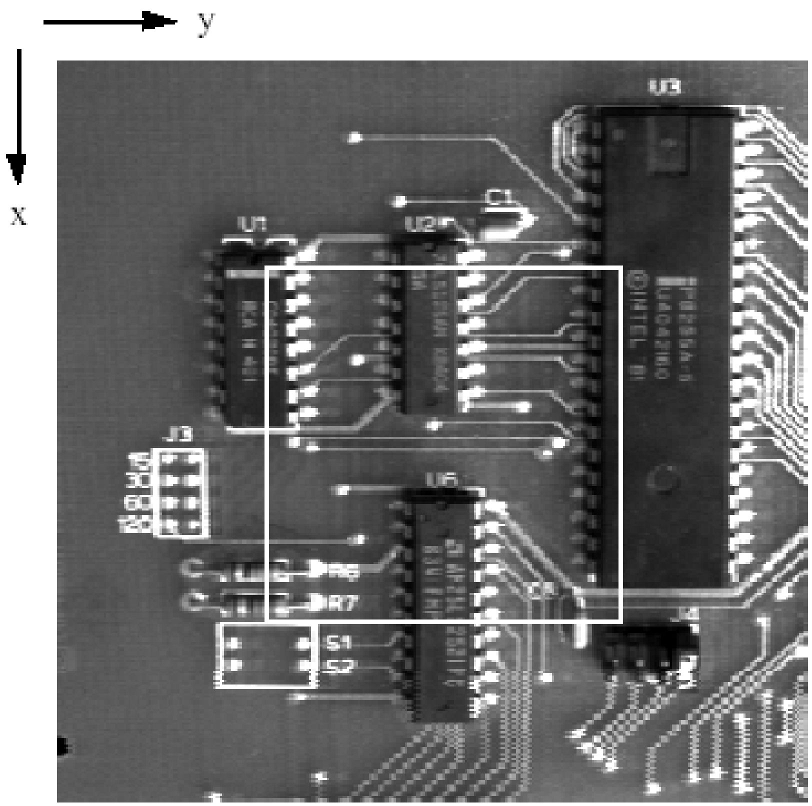


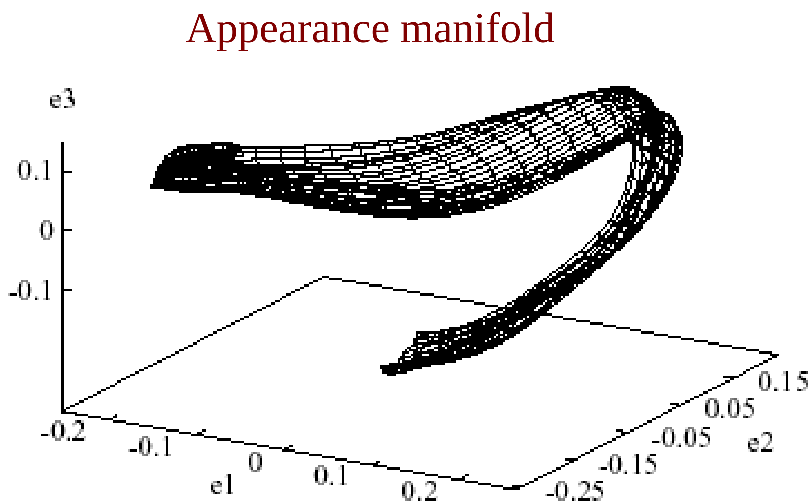
Figure 3: Estimation of the task coordinates (for instance, robot's displacement from its desired position in servoing) is based on the observation that each task coordinate can be forced to produces a unique visual appearance.

Visual Positioning Tasks

Every position of the camera gives a different image, which then defines a point in the eigenspace. Thus, every camera position corresponds to a point in the eigenspace (and vice versa).

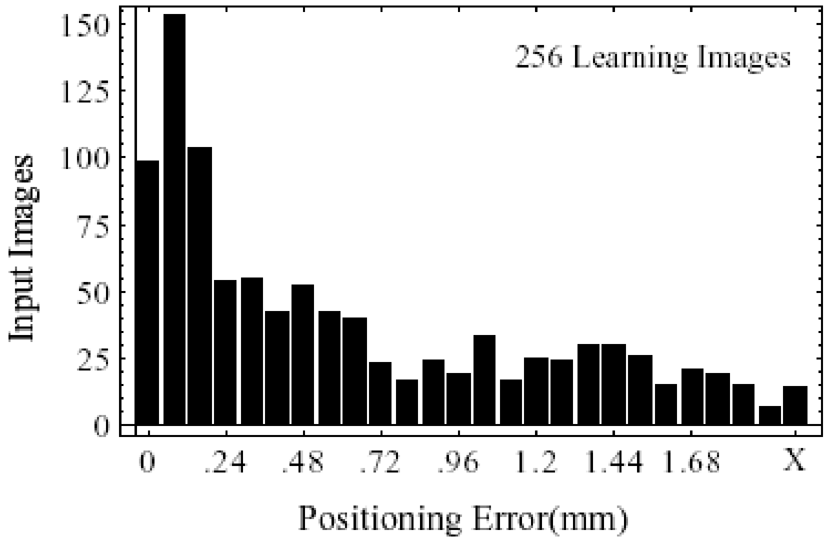


(a)

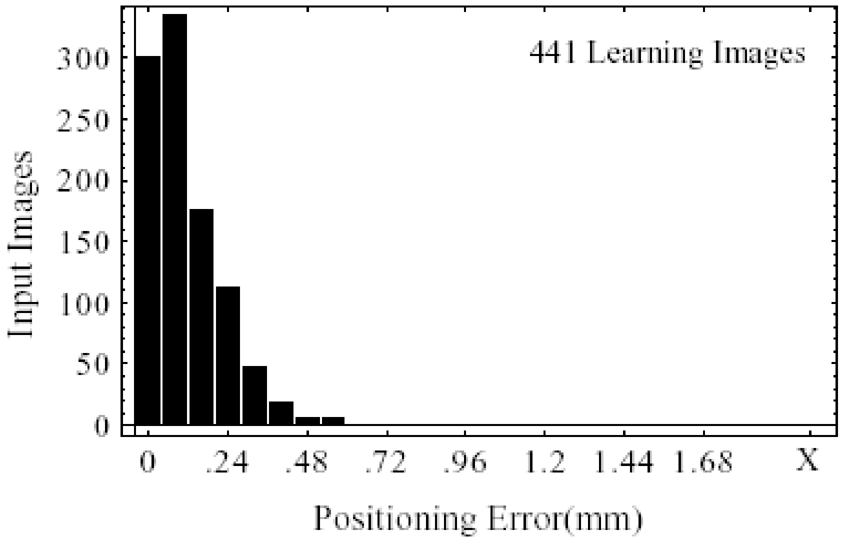


(b)

The applications use a 15-dimensional eigenspace - only a 3-D subspace is shown here.



(c)

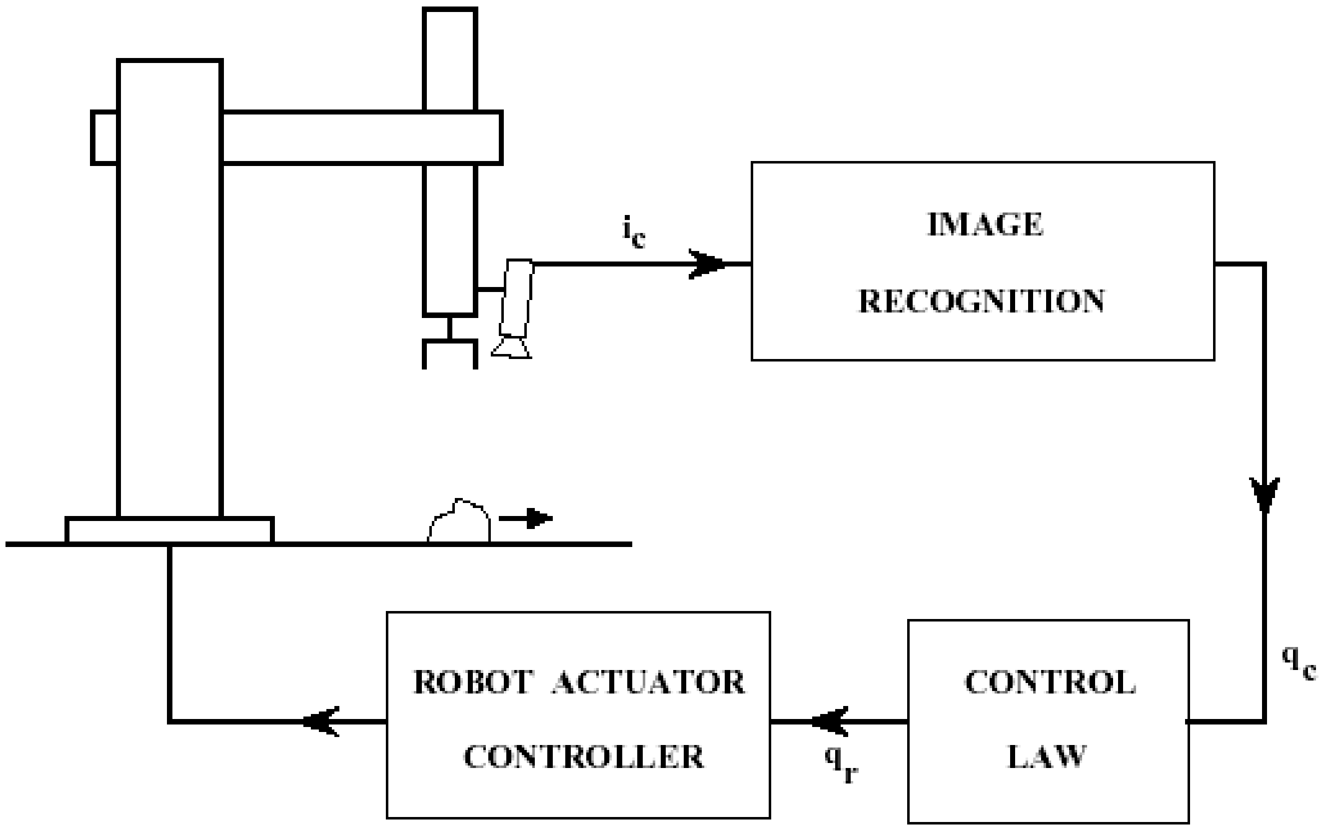


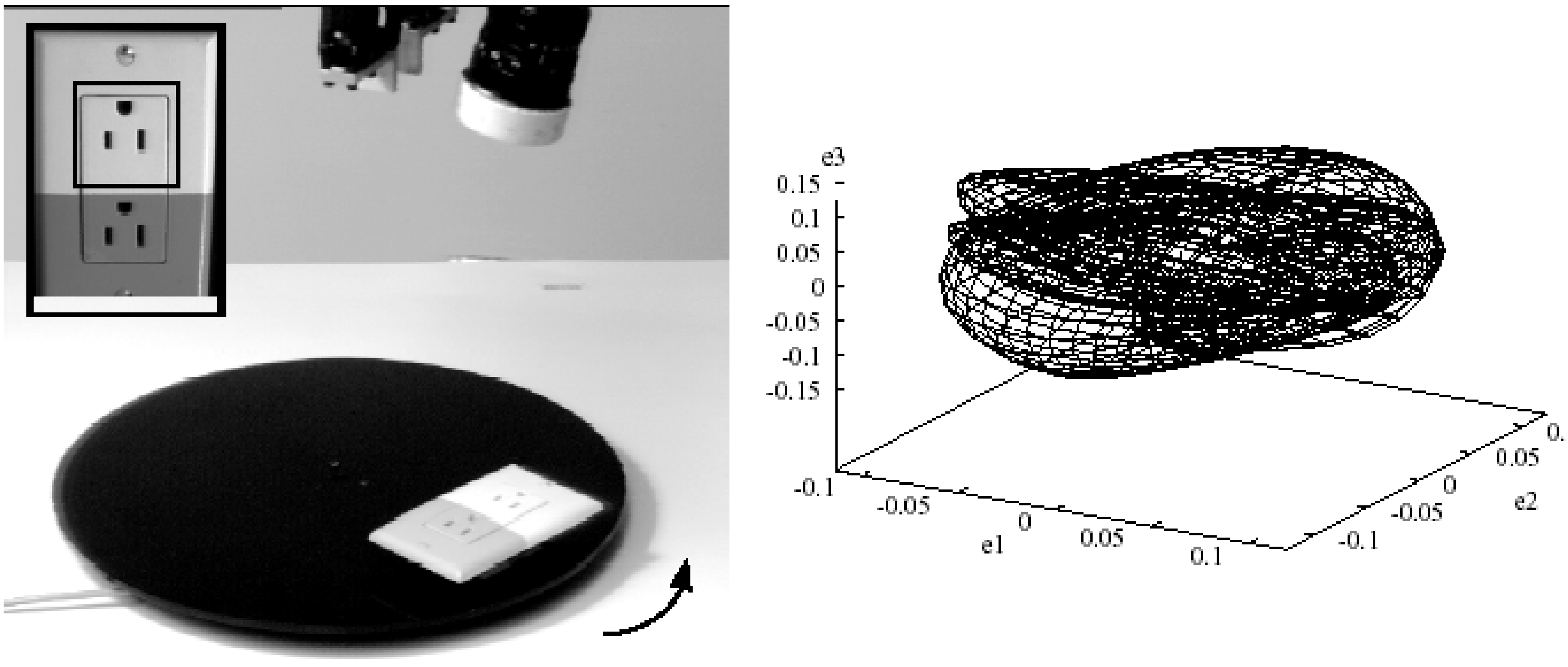
(d)

Figure 6: Visual positioning experiment: printed circuit board. (a) Image window used for learning and positioning. (b) Parametric eigenspace representation of visual workspace displayed in 3-D. Displacements are in two dimensions (x and y). Histograms of absolute positioning error (in mm) for (c) 256 learning images and (d) 441 learning images.

Visual Tracking Tasks

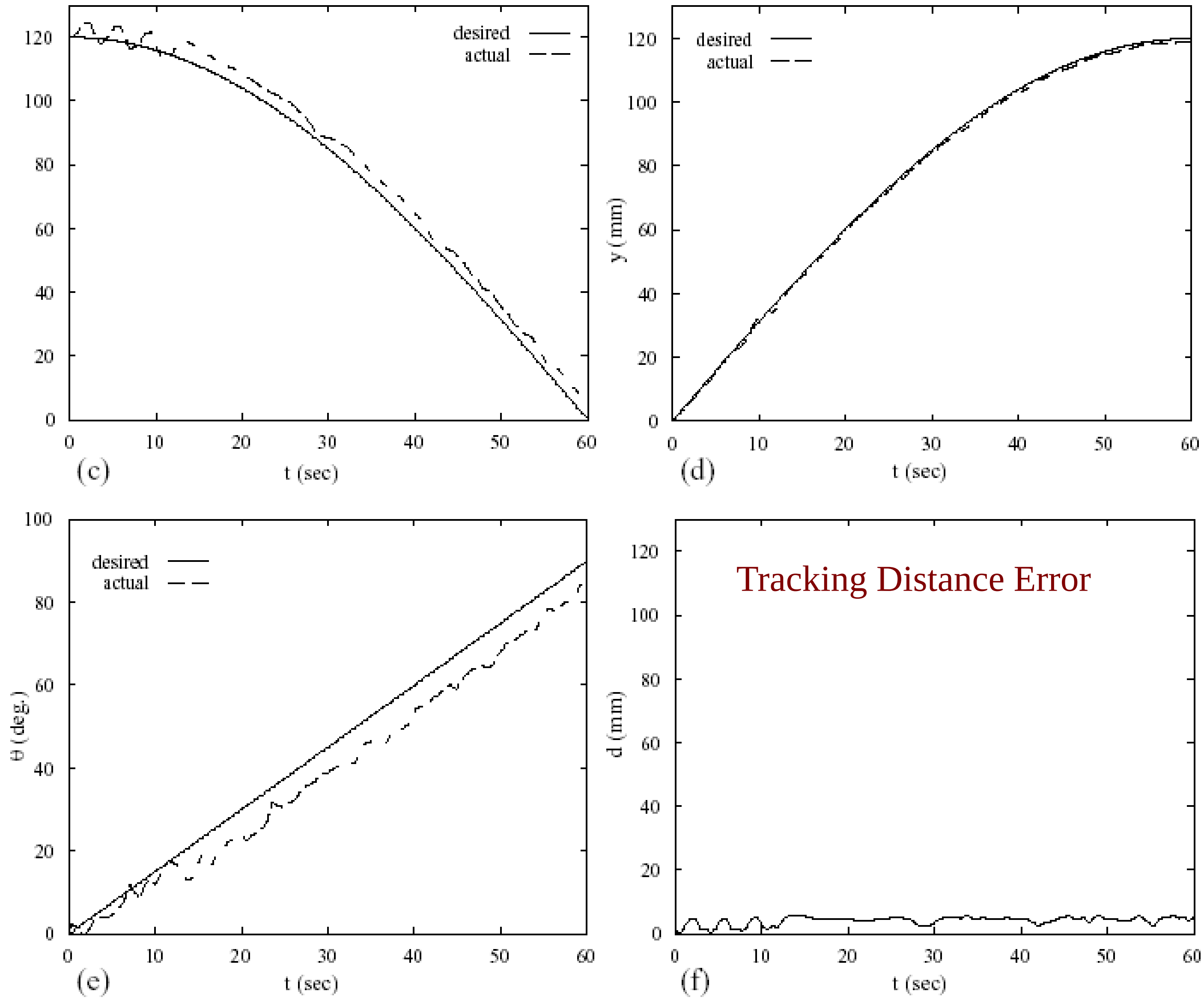
The technique used for position control of a robot can also be used for tracking tasks. The position determined by the eigenspace method is used as a feedback signal in a servo loop.





(a) (b)

Desired and Actual Coordinates (x,y,θ)



Inspection Tasks

- Inspection of objects can be easily implemented with appearance-based eigenmethods.
- Nayar et. al. describe a system which passes a camera in a known trajectory over a good object (one with no defects).
- The images obtained over this trajectory are projected onto a low dimensional eigenspace and the eigenspace trajectory is stored.

Inspection Tasks

- Then, an object to be inspected is presented and the camera passes over it in the same trajectory.
- The images are again projected onto the eigenspace, and the distances between points in the inspected object eigenspace trajectory and points in the model object trajectory are computed.

Inspection Tasks

- A defect in the object under inspected will show up as a large distance in the eigenspace between the two trajectories at the location of the defect.

Image of a good circuit board

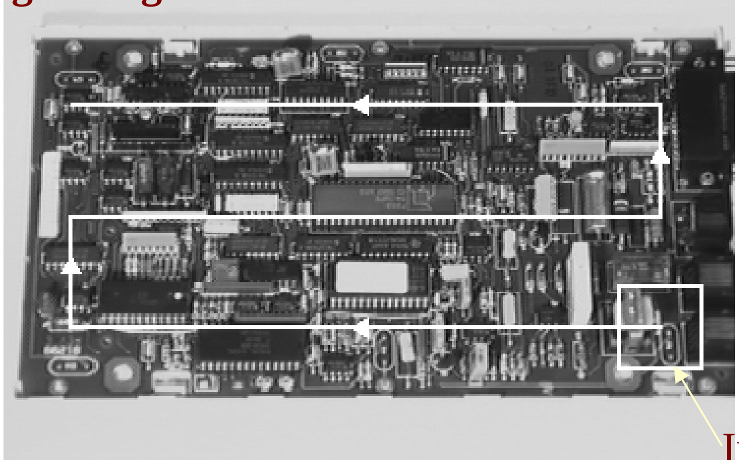


Image window

Trajectory in eigenspace

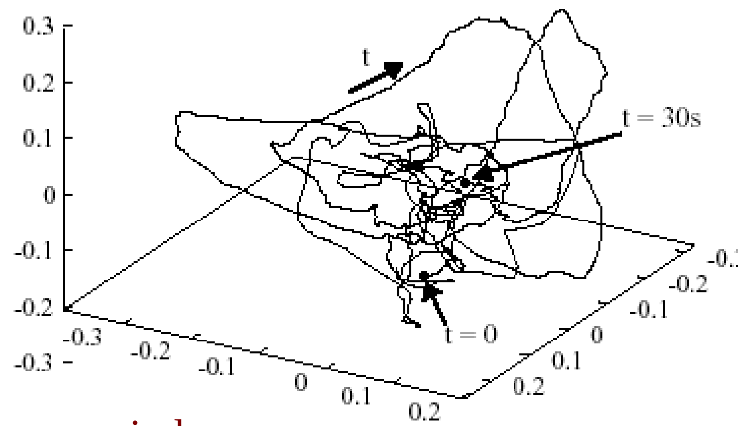
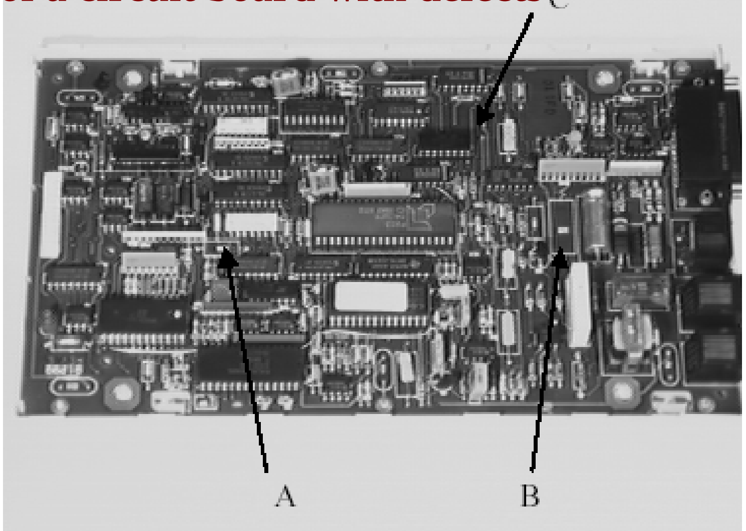
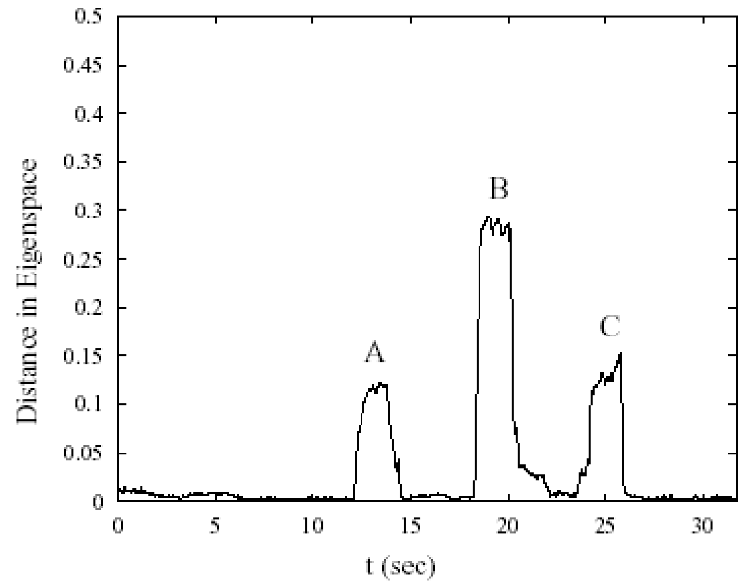


Image of a circuit board with defects



(c)



(d)