

Object Recognition Contest

- Students project IPCV Budapest 2006 -

Dietrich Paulus, Wolfram Hans

paulus@uni-koblenz.de

hans@uni-koblenz.de

Institute of Computational Visualistics
University Koblenz-Landau

21. August 2006

Goal

Given an image

→

Recognize object

Restrictions

- ▶ Small set of possible objects (approx. 12)
- ▶ Only planar objects (2D processing)
- ▶ Homogeneous background

Problems

- ▶ size / orientation changes
- ▶ illumination changes
- ▶ different background

Goal

Given an image



Recognize object

Restrictions

- ▶ Small set of possible objects (approx. 12)
- ▶ Only planar objects (2D processing)
- ▶ Homogeneous background

Problems

- ▶ size / orientation changes
- ▶ illumination changes
- ▶ different background

Goal

Given an image



Recognize object

Restrictions

- ▶ Small set of possible objects (approx. 12)
- ▶ Only planar objects (2D processing)
- ▶ Homogeneous background

Problems

- ▶ size / orientation changes
- ▶ illumination changes
- ▶ different background

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance“ of features to all known feature sets
- ▶ select „closest“ set

Details will be given in the lectures

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance“ of features to all known feature sets
- ▶ select „closest“ set

Details will be given in the lectures

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance” of features to all known feature sets
- ▶ select „closest” set

Details will be given in the lectures

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance“ of features to all known feature sets
- ▶ select „closest“ set

Details will be given in the lectures

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance“ of features to all known feature sets
- ▶ select „closest“ set

Details will be given in the lectures

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance” of features to all known feature sets
- ▶ select „closest” set

Details will be given in the lectures

Some Details on the Strategy

Given an image ...

- ▶ try to isolate image from background
- ▶ compute so-called *features* (e.g. mean of the object region)

Given several images of one object type (a so-called *class*) ...

- ▶ compute features of each image
- ▶ store set of features somehow for that class

Given one images and a set of known classes

- ▶ compute features for the image
- ▶ compute „distance” of features to all known feature sets
- ▶ select „closest” set

Details will be given in the lectures

Organization

- ▶ Work in small groups (6 Students)
- ▶ Groups should be multi-national
- ▶ Approach is up to you – we only give hints
- ▶ During the two weeks
 - ▶ you will get to know nice features in the lectures
 - ▶ implement them in the exercises and project work
 - ▶ create a tool box for feature extraction
- ▶ Contest on last friday

Sample images (1)



Sample images (2)



Challenges

- ▶ object size & orientation may vary
- ▶ image size may be different
- ▶ background may be different or textured

Hint: start working on the simple cases!

Schedule

- ▶ Mo: build groups – view sample images
- ▶ Tu-Fr: start implementing feature extractors
- ▶ Fr: create image database
- ▶ Mo-Tu: work on the database
- ▶ We: Define test framework
- ▶ Th: learn about color histograms, integrate them
- ▶ Fr: contest

Hints:

- ▶ Start to work *today!*
- ▶ Images will be in color
- ▶ First tasks:
 - ▶ separate object from background or identify background
 - ▶ compute simple features to test your system (e.g. mean color)
 - ▶ plot the feature vectors (e.g. with gnuplot) to see whether
 - ▶ match for object of one class
 - ▶ they differ for different classes