Fast Pattern Detection using Orbit Decomposition

Yacov Hel-Or

The Interdisciplinary Center

On Sabbatical at HP Labs

joint work with

Hagit Hel-Or

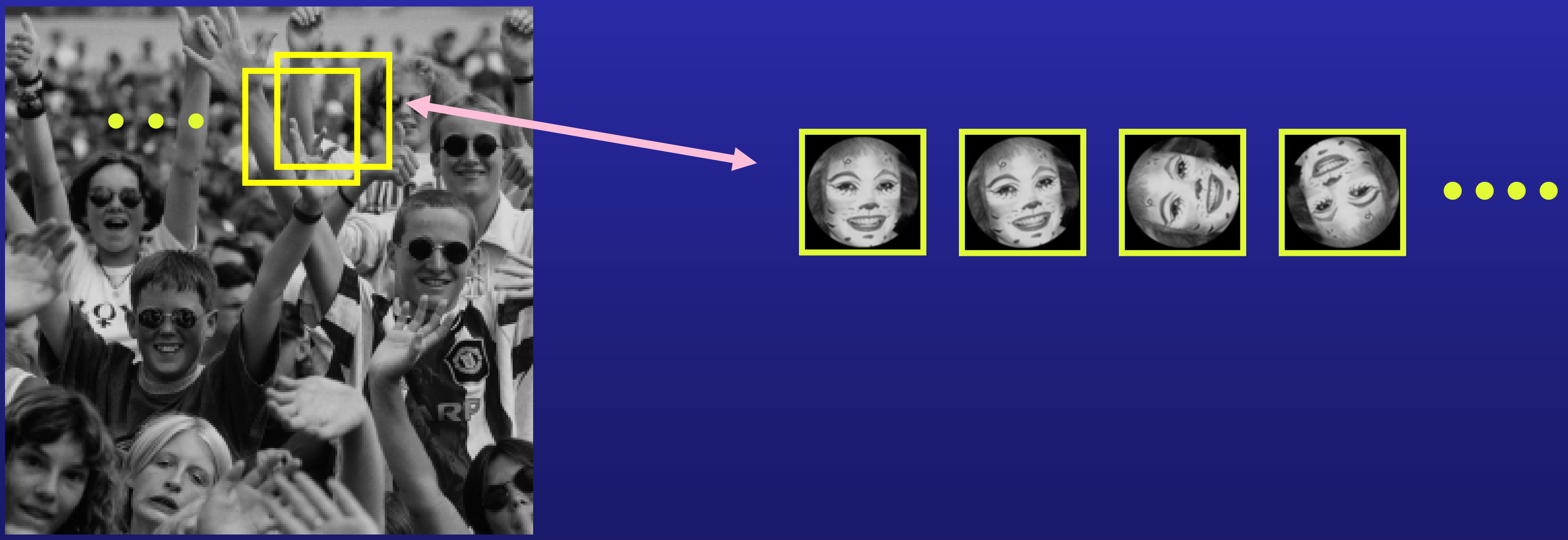
Haifa University

On Sabbatical at Stanford U.

Pattern Detection

A given pattern is sought in an image.

- The pattern may appear at any location in the image.
- The pattern may be subject to any transformation (within a given transformation group).



Example

Face detection in images

Why is it Expensive?
The search in Spatial Domain

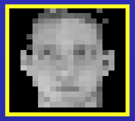
Searching for faces in a 1000x1000 image, is applied 1e6 times, for each pixel location.

A very expensive search problem

Why is it difficult?

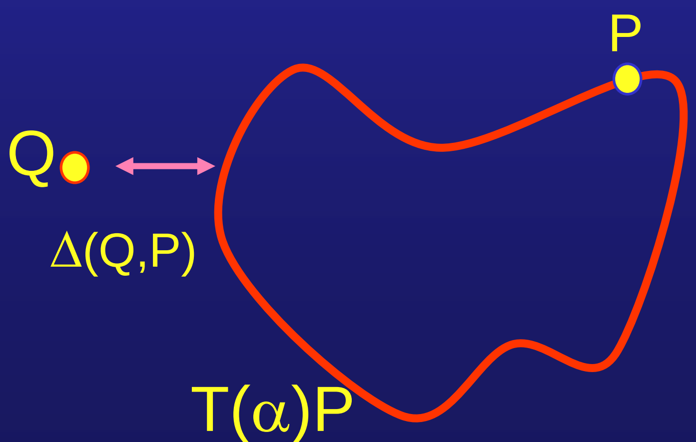
The Search in Transformation Domain

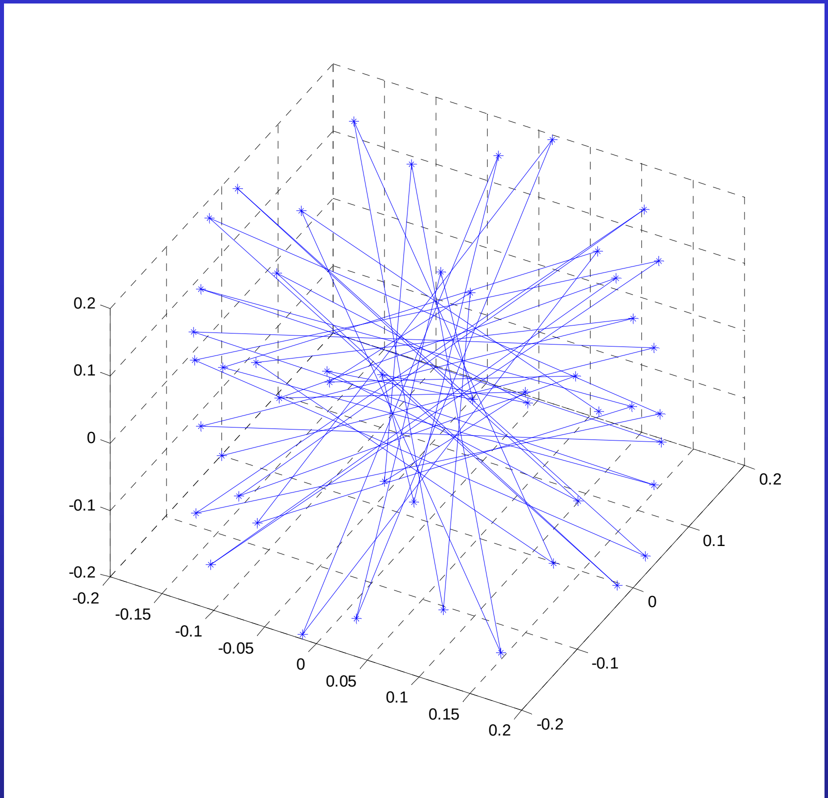
- A pattern under transformations draws a very complex manifold in “pattern space”:

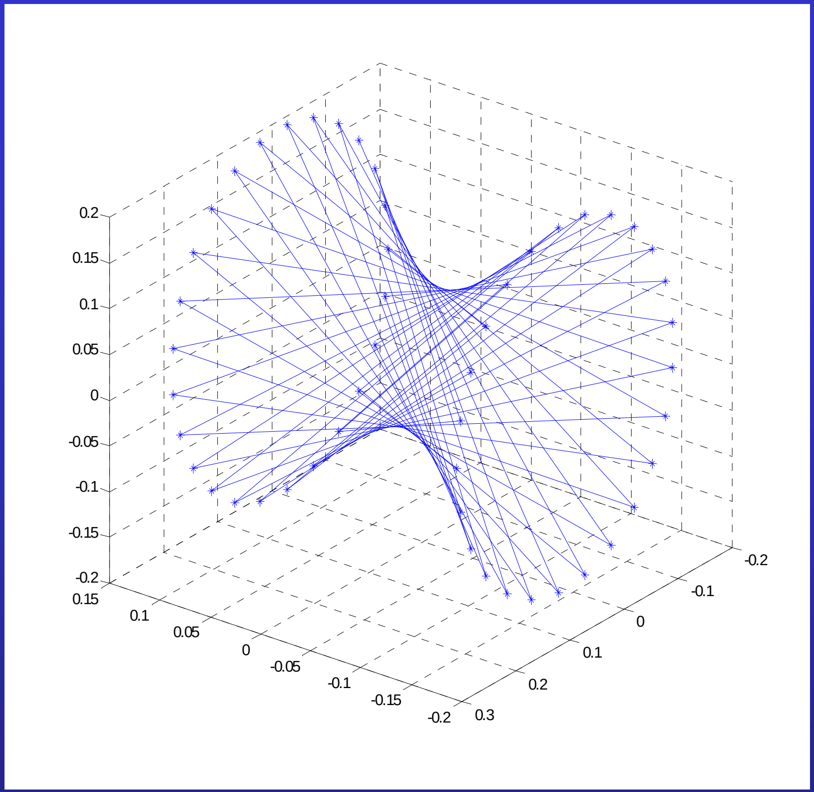


$\longleftrightarrow \quad P \in \Re^{kxk}$

  - In a very high dimensional space.
  - Non convex.
  - Non regular (two similarly perceived patterns may be distant in pattern space).

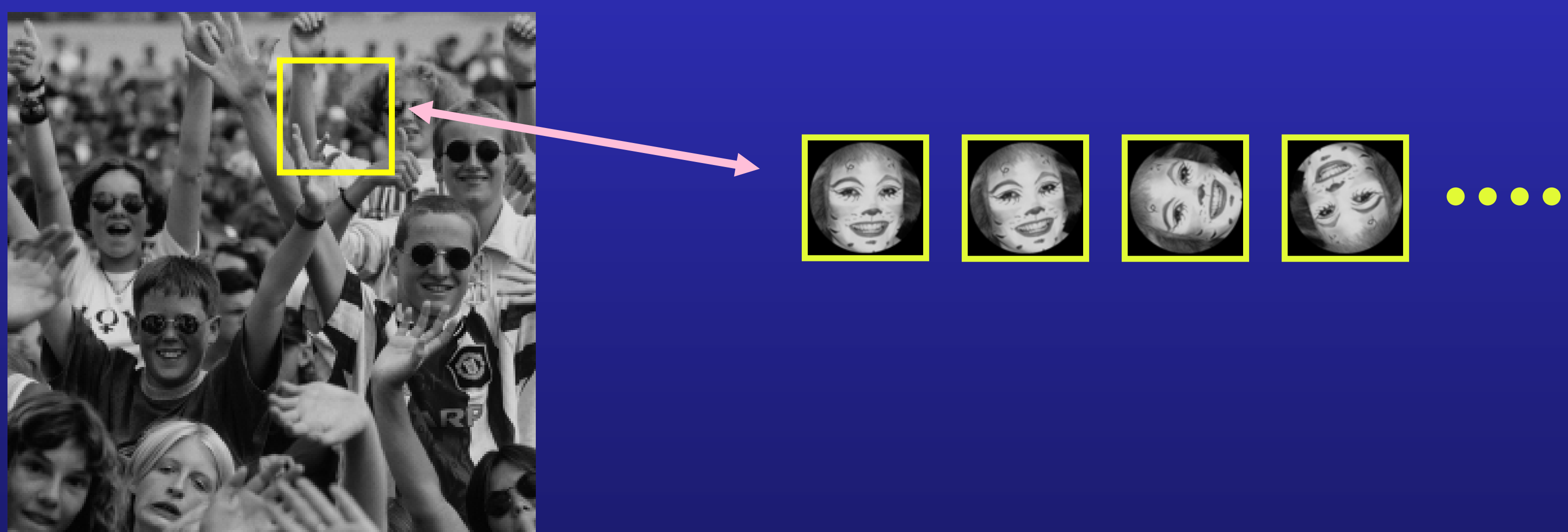






A rotation manifold of a pattern drawn in “pattern-space”
The manifold was projected into its three most significant components.

Efficient Search in the Transformation Domain

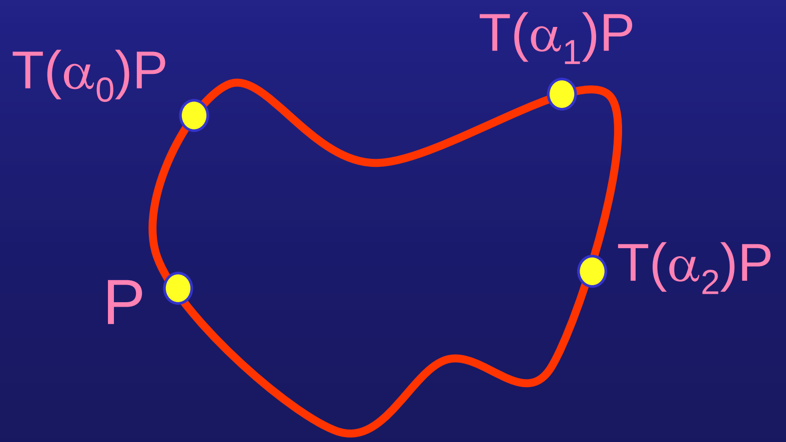


Transformation Manifold

A pattern P can be represented as a point in $\Re^{kxk}$

$T(\alpha)P$ is a transformation $T(\alpha)$ applied to pattern **P**.

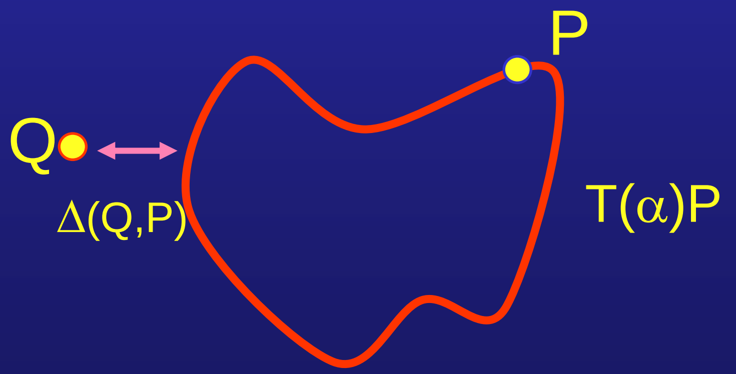
$T(\alpha)P$ for all α forms an orbit in $\Re^{kxk}$



Fast Search in Group Orbit

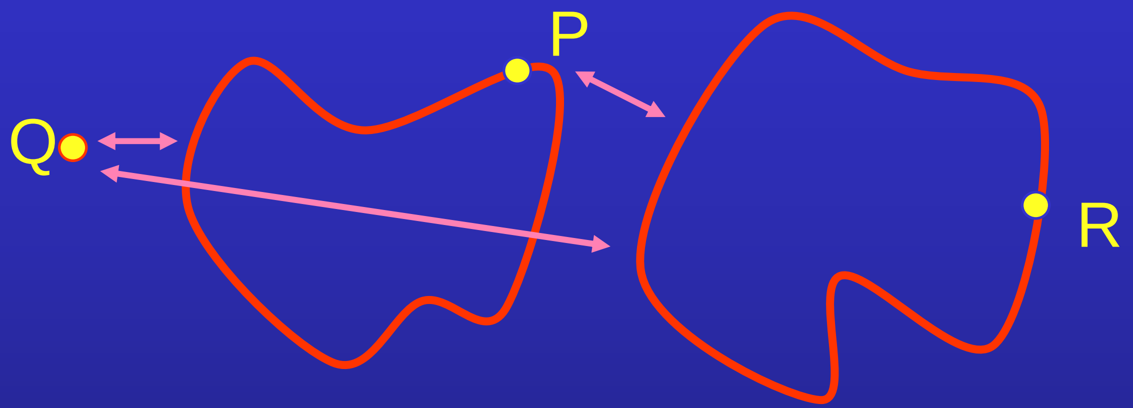
- Assume d(Q,P) is a distance metric.
- We would like to find

$$\Delta(Q,P)=\min_{\alpha} d(Q, T(\alpha)P)$$



Fast Search in Group Orbit (Cont.)

- In the general case $\Delta(Q,P)$ is not a metric.

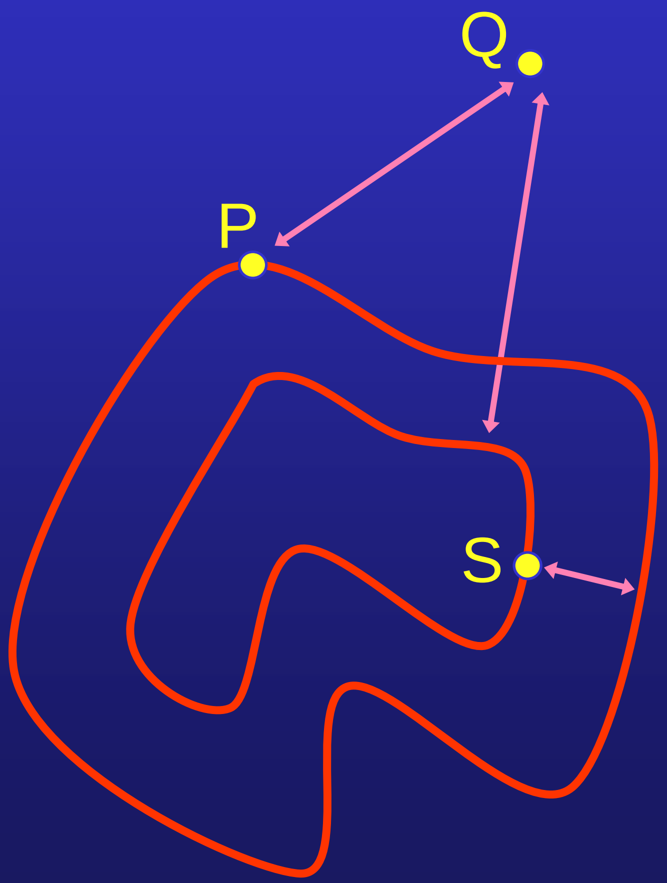


- Observation: if $d(Q,P) = d(T(\alpha)Q, T(\alpha)P)$

- $\Delta(Q,P)$ is a **metric**
- Point-to-Orbit dist. = Orbit-to-Orbit dist.

Fast Search in Group Orbit (Cont.)

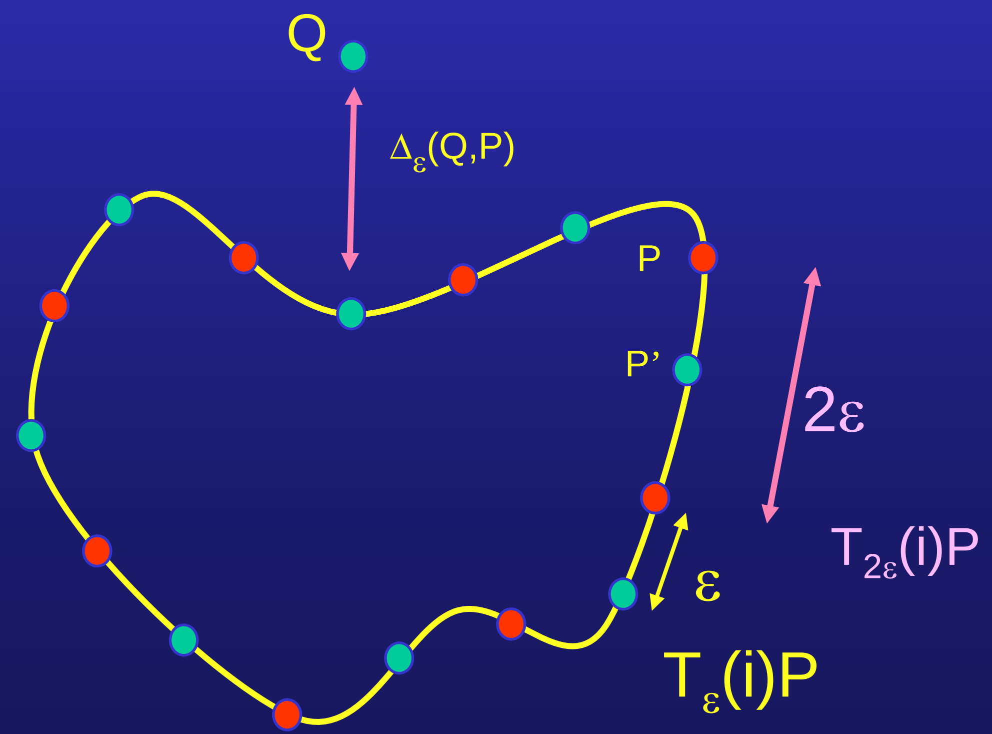
The metric property of $\Delta(Q,P)$ implies triangular inequality on the distances.



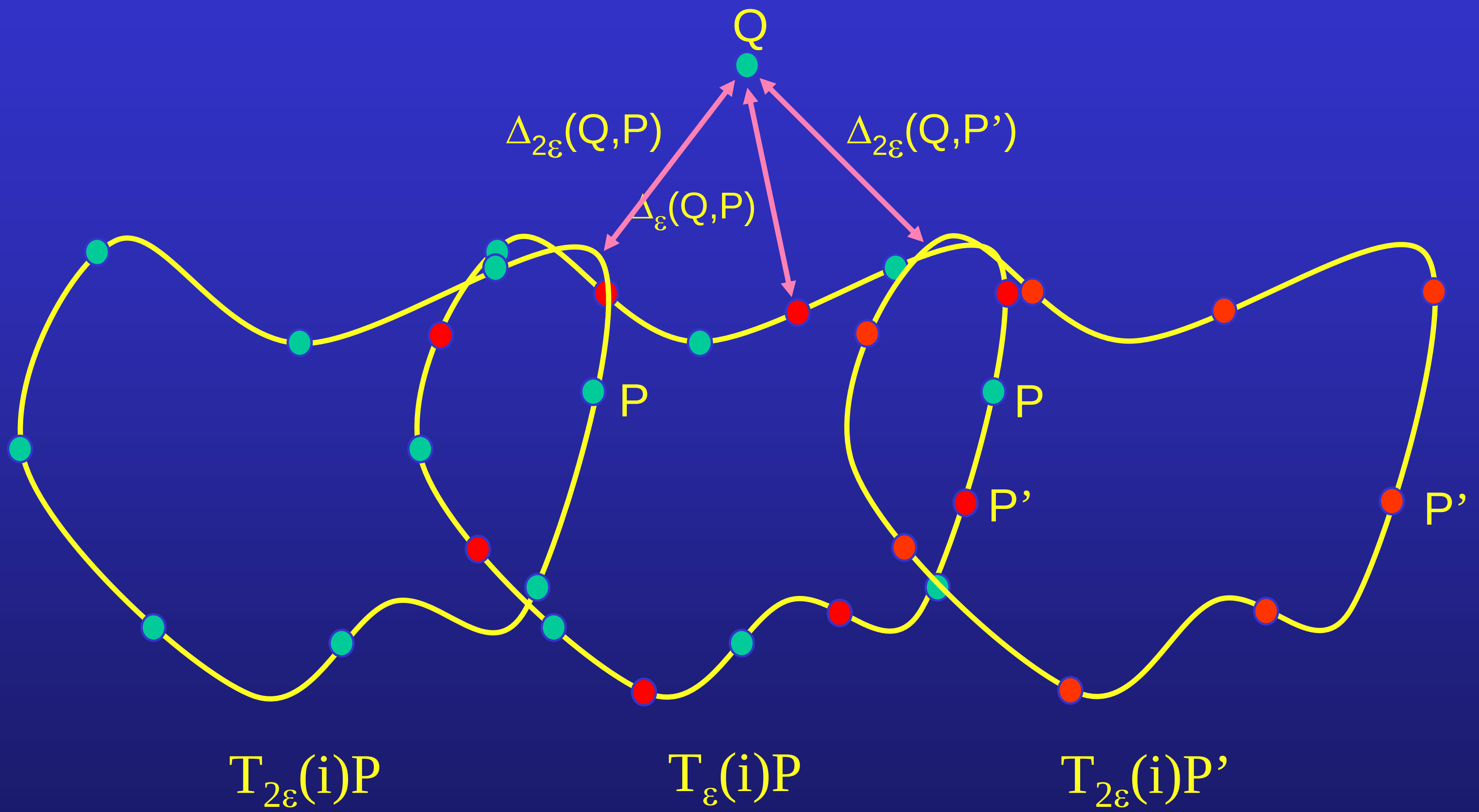
Orbit Decomposition

- In practice $T(\alpha)$ is sampled into $T(\varepsilon i)=T_{\varepsilon}(i)$, $i=1,2,\ldots$
- We can divide $T_{\varepsilon}(i)P$ into two sub-orbits:

$T_{2\varepsilon}(i)P$ and $T_{2\varepsilon}(i)P'$ where $P'= T_{\varepsilon}(1)\,P$

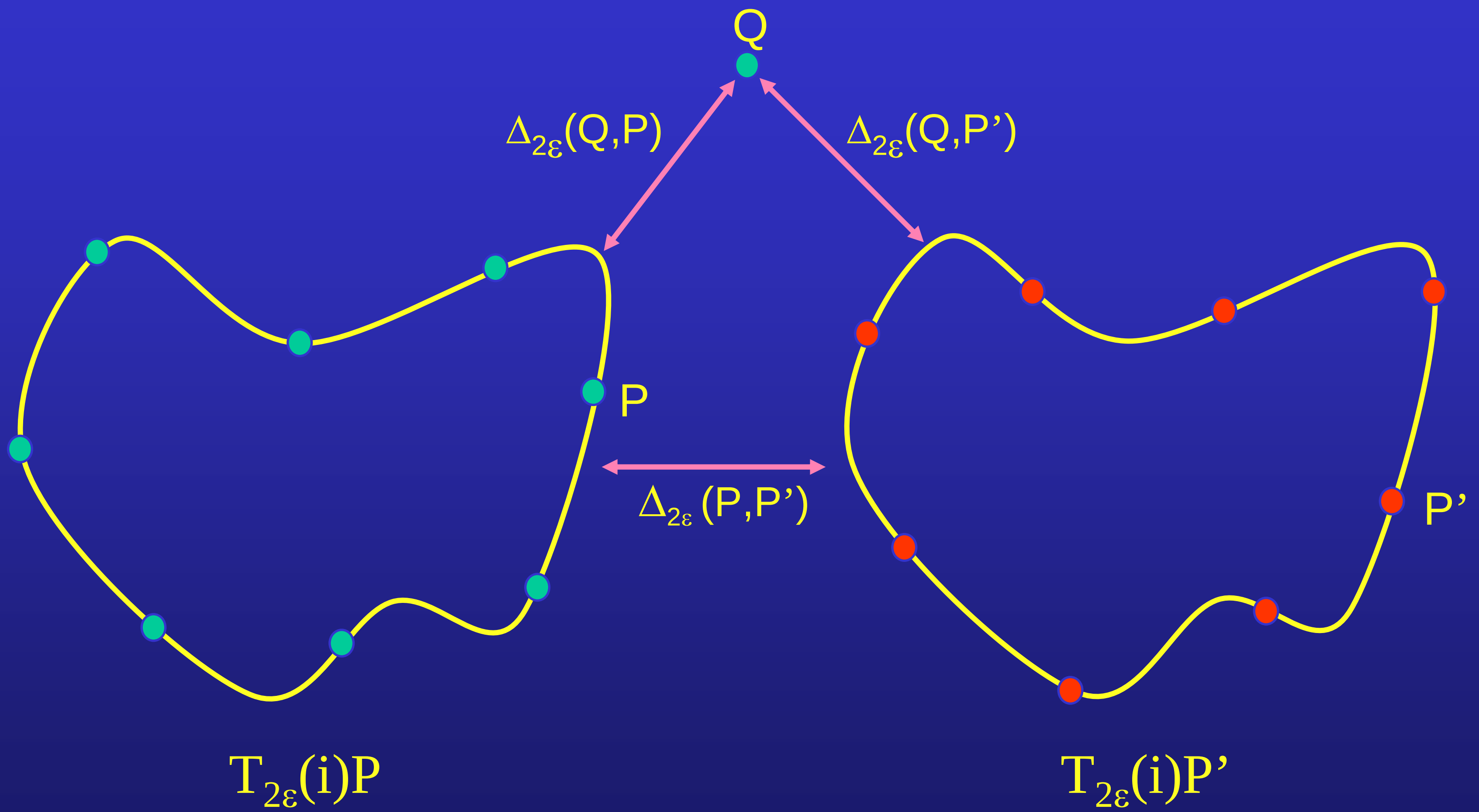


Orbit Decomposition (Cont.)



$$\Delta_\varepsilon(Q,P) = Min\{\Delta_{2\varepsilon}(Q,P), \Delta_{2\varepsilon}(Q,P')\}$$

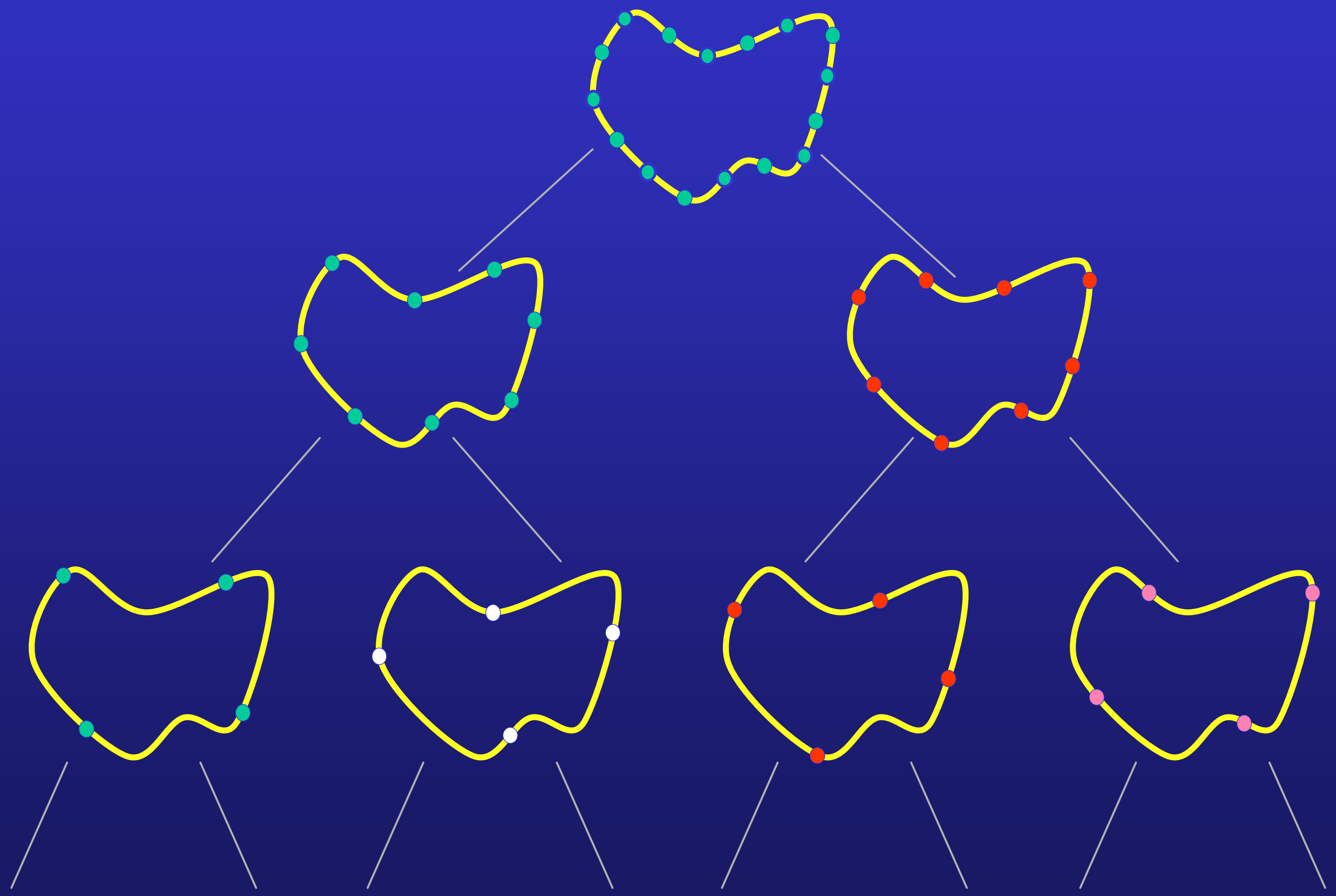
Orbit Decomposition (Cont.)



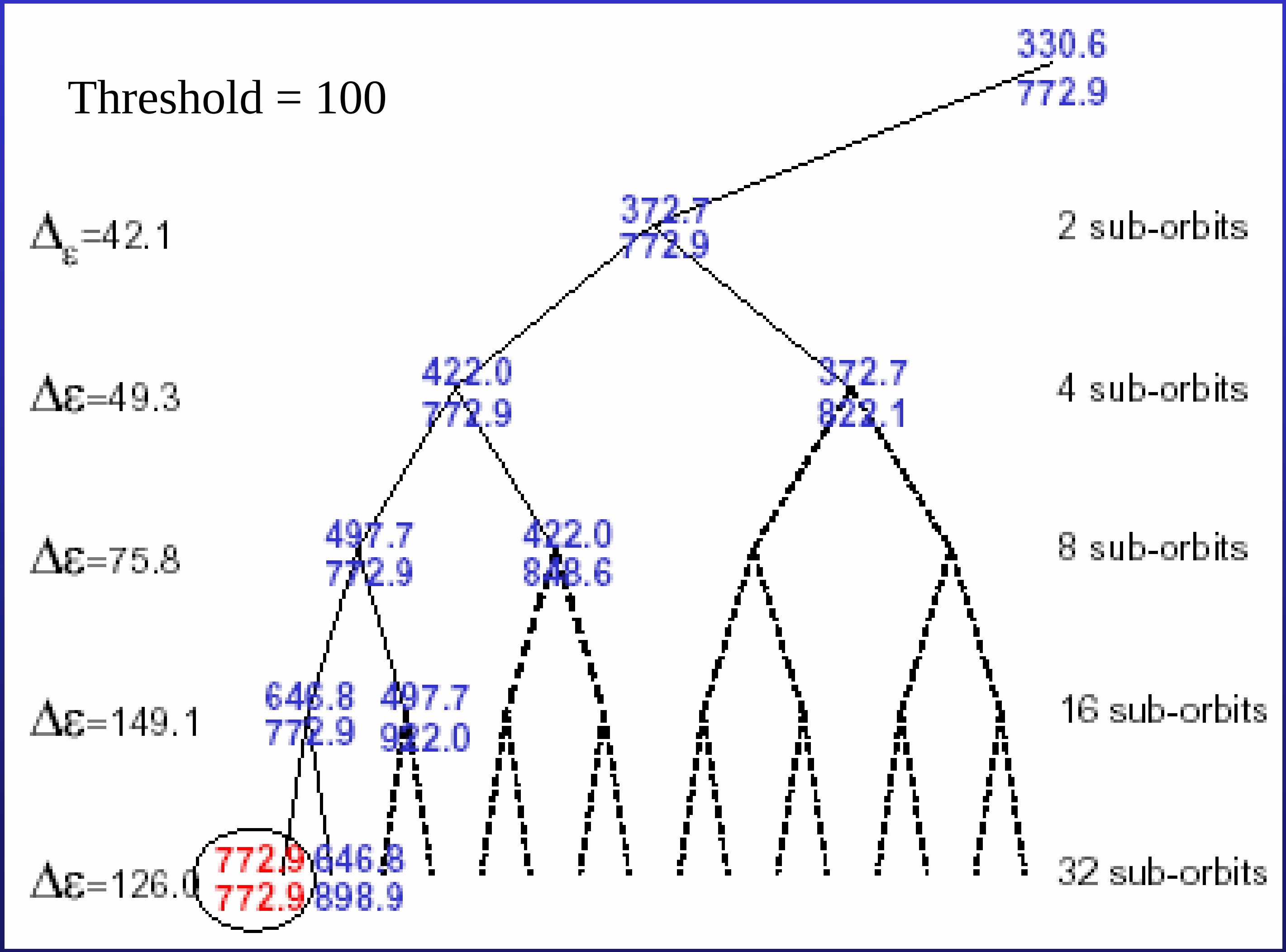
Since $\Delta_{2\varepsilon}$ is a metric and $\Delta_{2\varepsilon}$(P,P') can be calculated in advance we may save calculations using the triangle inequality constraint.

The Orbit Tree

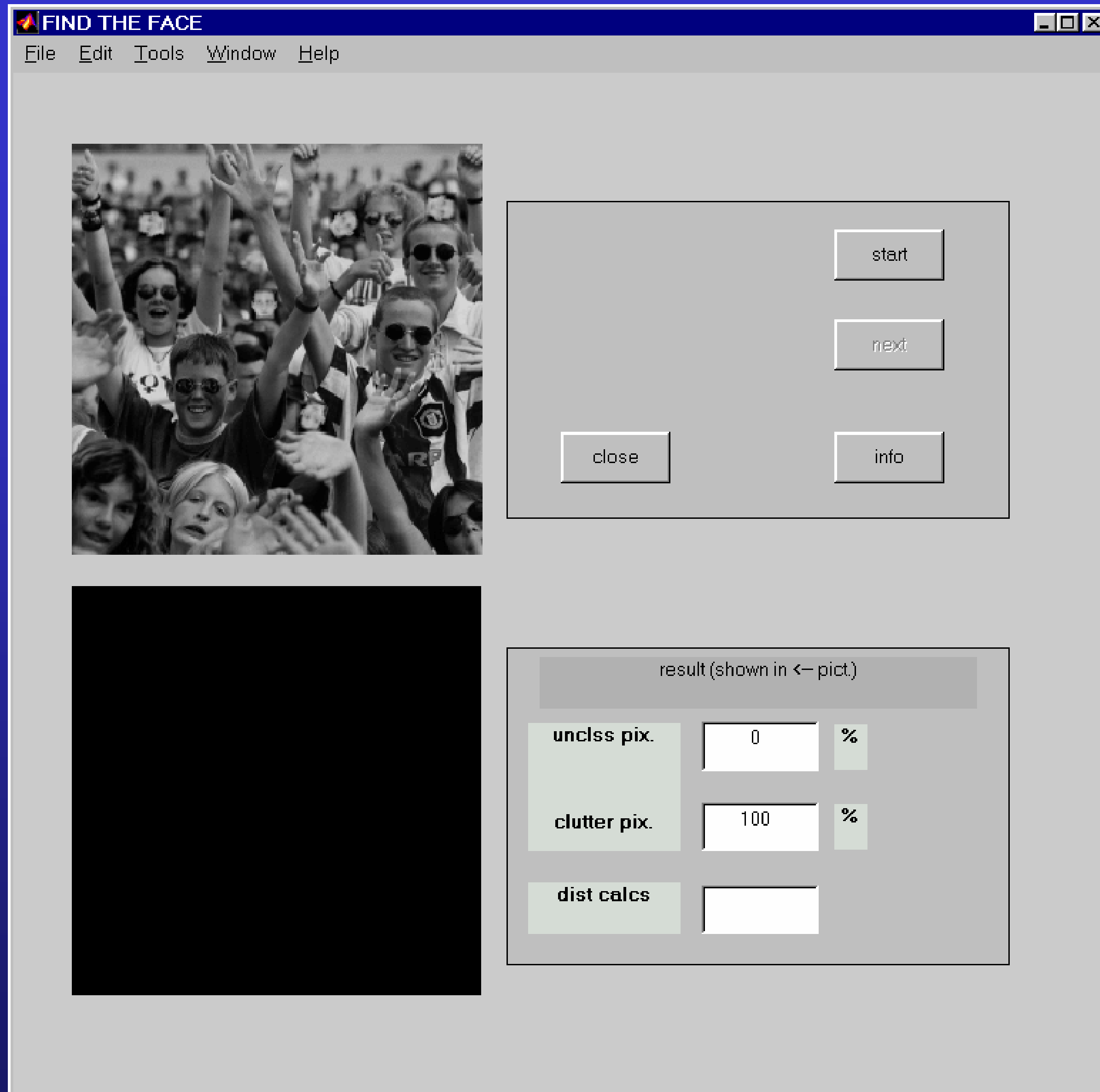
- The sub-group subdivision can be applied recursively.

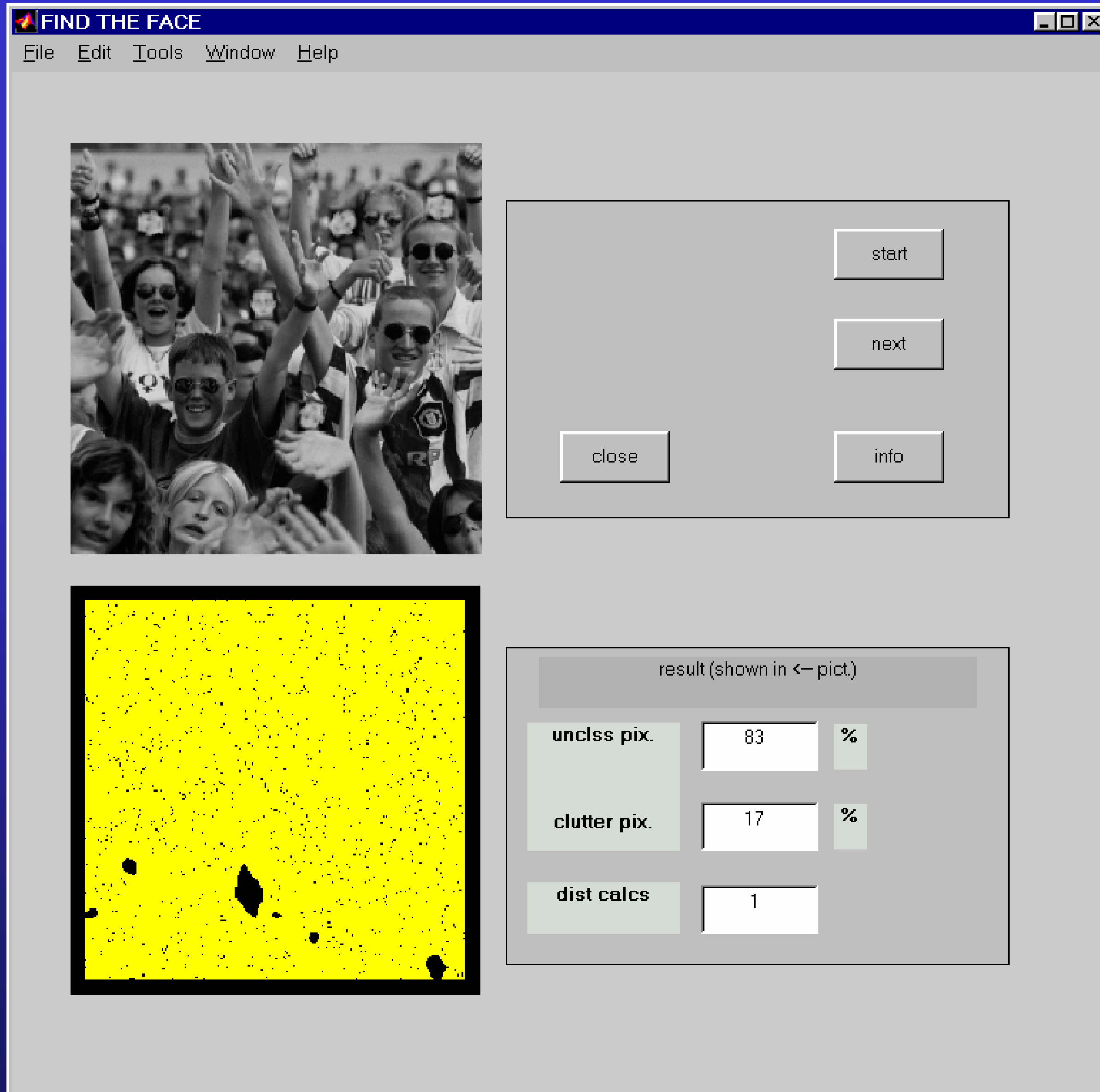


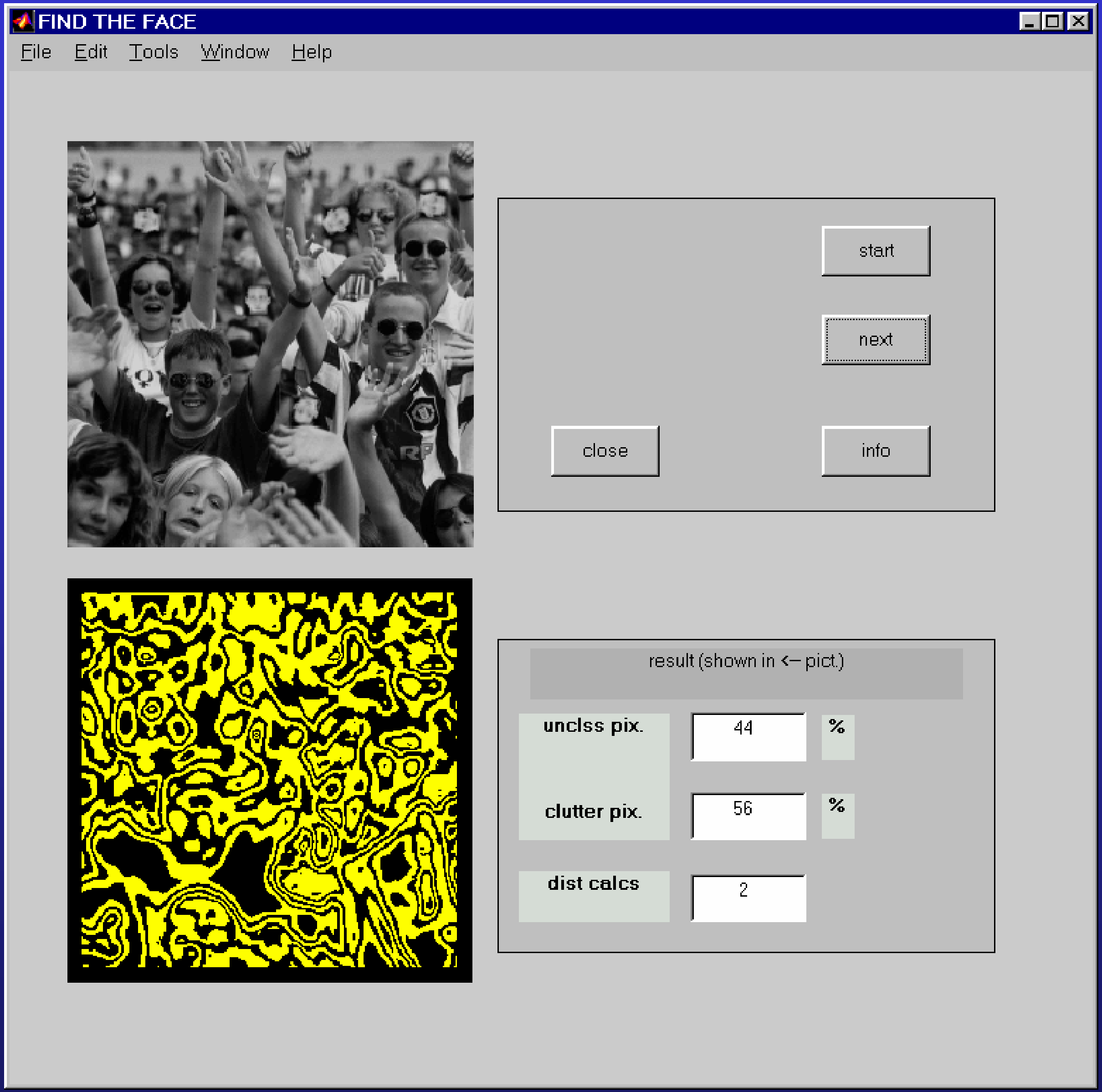
Orbit Tree



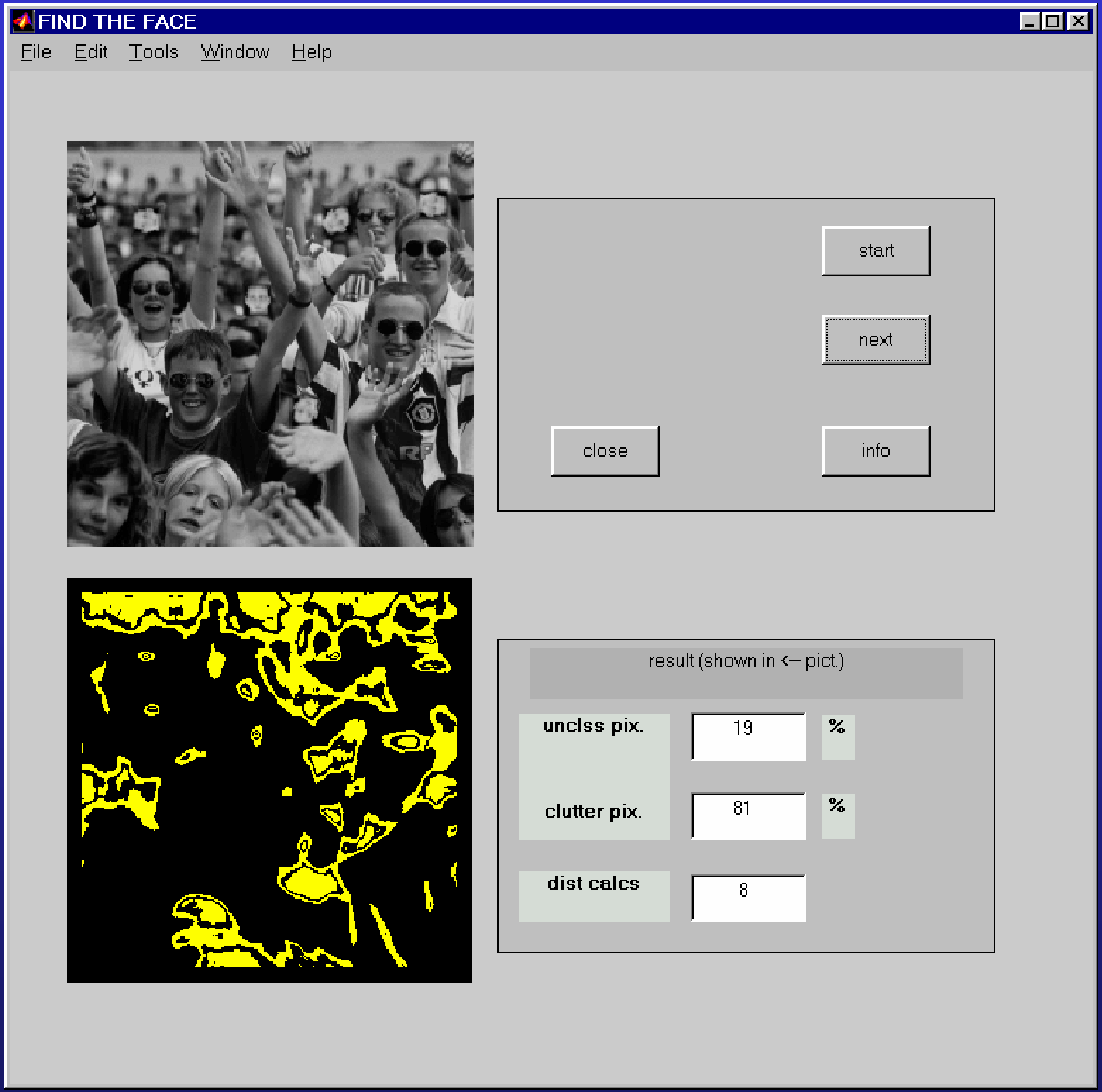
Threshold = 100
330.6
772.9
Δ_ε=42.1
372.7
772.9
2 sub-orbits
$\Delta\varepsilon$=49.3
422.0
772.9
372.7
822.1
4 sub-orbits
$\Delta\varepsilon$=75.8
497.7
772.9
422.0
848.6
8 sub-orbits
$\Delta\varepsilon$=149.1
646.8
772.9
497.7
922.0
16 sub-orbits
$\Delta\varepsilon$=126.0
772.9
772.9
646.8
898.9
32 sub-orbits

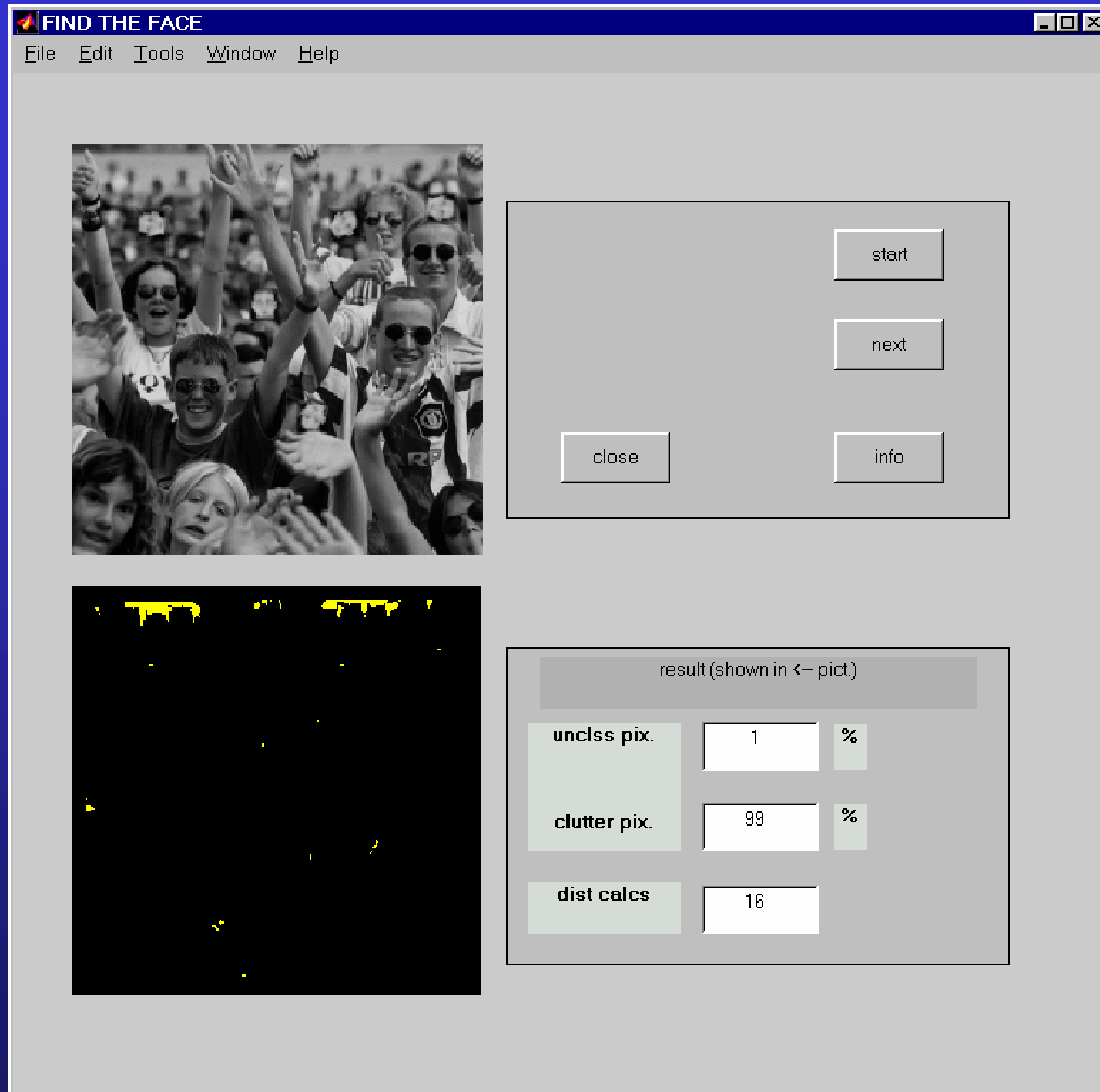

FIND THE FACE
File
Edit
Tools
Window
Help
start
next
close
info
result (shown in <– pict.)
unclss pix.
0
%
clutter pix.
100
%
dist calcs

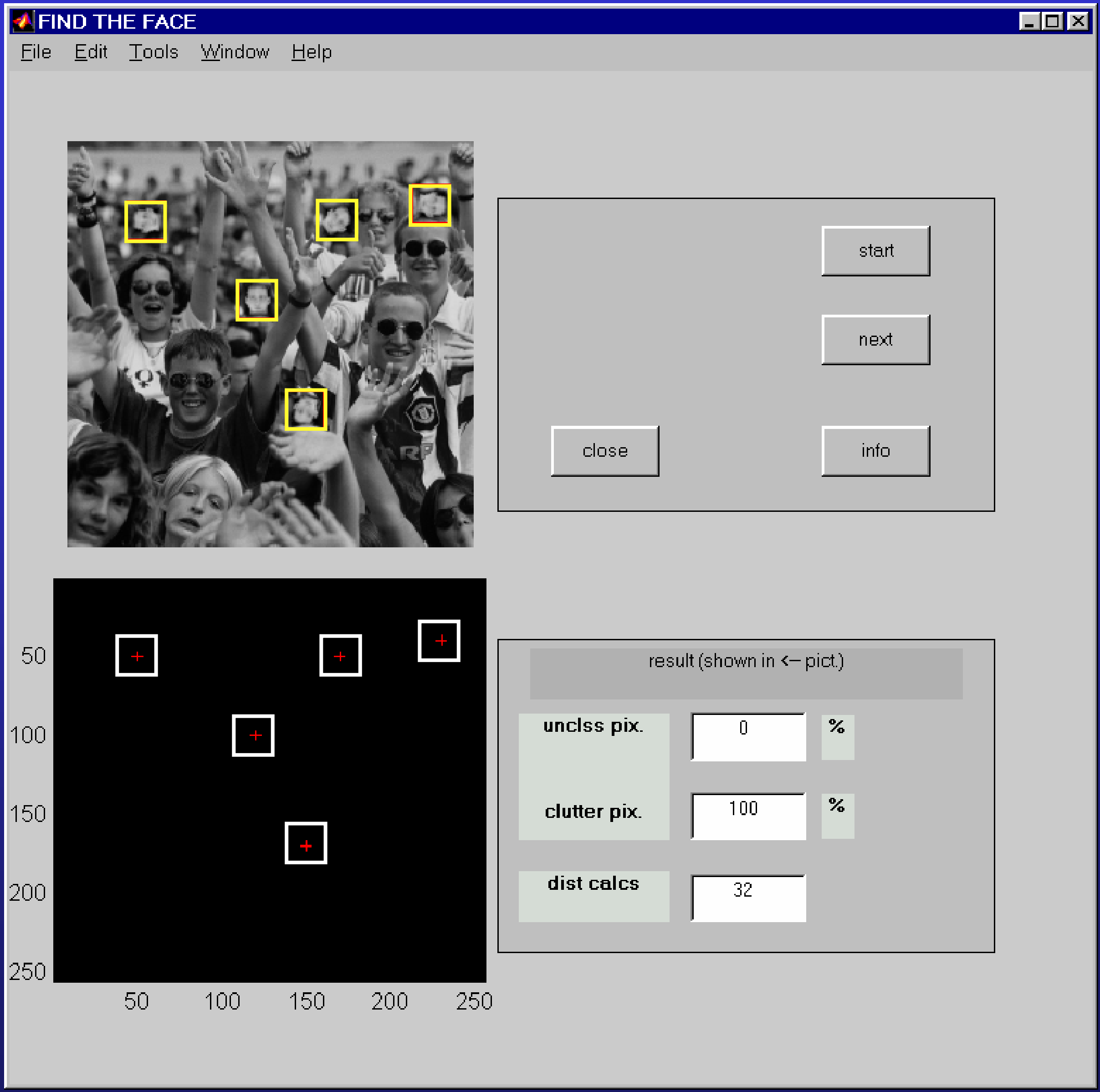

FIND THE FACE
File Edit Tools Window Help
start
next
close
info
result (shown in <– pict.)
unclss pix.
83
%
clutter pix.
17
%
dist calcs
1


FIND THE FACE
File
Edit
Tools
Window
Help
start
next
close
info
result (shown in <-- pict.)
unclss pix.
44
%
clutter pix.
56
%
dist calcs
2

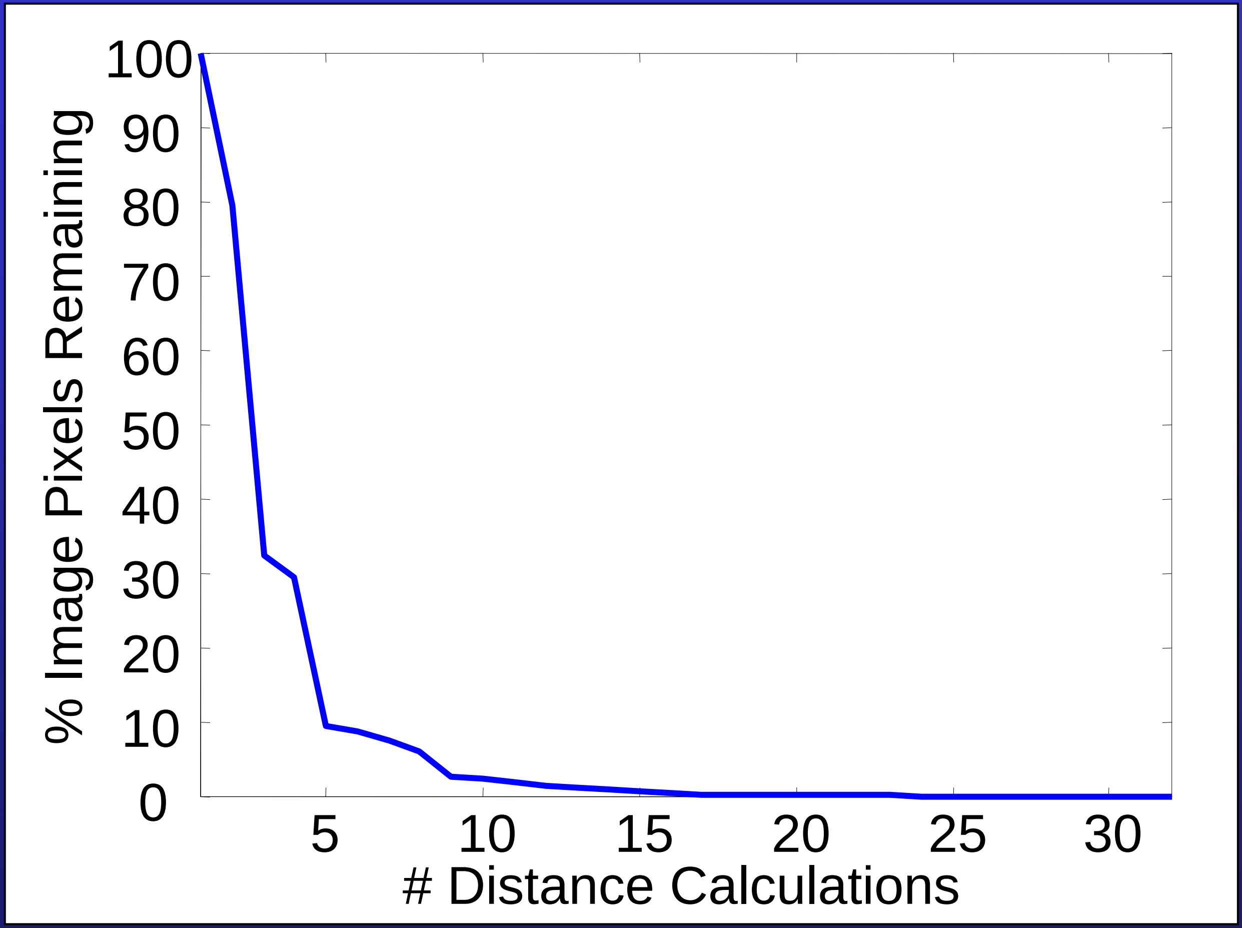

FIND THE FACE
File Edit Tools Window Help
start
next
close
info
result (shown in <– pict.)
unclss pix.
35
%
clutter pix.
65
%
dist calcs
4


FIND THE FACE
File Edit Tools Window Help
start
next
close
info
result (shown in <– pict.)
unclss pix.
19
%
clutter pix.
81
%
dist calcs
8


FIND THE FACE
File Edit Tools Window Help
start
next
close
info
result (shown in <– pict.)
unclss pix.
1
%
clutter pix.
99
%
dist calcs
16


FIND THE FACE
File
Edit
Tools
Window
Help
start
next
close
info
50
100
150
200
250
50
100
150
200
250
result (shown in <– pict.)
unclss pix.
0
%
clutter pix.
100
%
dist calcs
32

Rejection Rate



Average number of distance computations per pixel is 2.868

Fast Search in Group Orbit: Conclusions

- <u>Observation 1</u>: Orbit distance is a metric when the point distance is transformation invariant.
- <u>Observation 2</u>: Fast search in orbit distance space can be applied using recursive orbit decomposition.
- Distant patterns are rejected fast.
- **<u>Important</u>**: Can be applied to metric spaces (Non Euclidean).