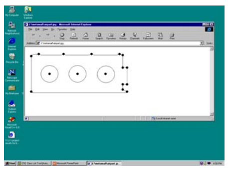
DS.G.1

## Application: Geometric Hashing

Model-based object recognition is the area of computer vision that tries to recognize and locate known objects in images.

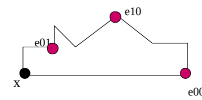
A geometric model of an object is a precise model of the kind produced by CAD systems that specifies the full geometry of the object in terms of the points, lines, surfaces that define it.



DS.G.2

Let M be an ordered set of points in a plane that constitutes the 2D model of an object.

Select 3 noncollinear points of M
e00, e10, e11
to be an affine basis set that defines a coordinate system on the object.




Then any point **x** in M can be represented in affine coordinates (ξ,η) where

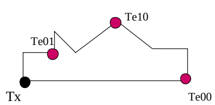
$$\mathbf{x} = \xi(\mathbf{e10} - \mathbf{e00}) + \eta(\mathbf{e01} - \mathbf{e00}) + \mathbf{e00}$$

DS.G.3

If $\mathbf{x} = \xi(\mathbf{e10} - \mathbf{e00}) + \eta(\mathbf{e01} - \mathbf{e00}) + \mathbf{e00}$

and we apply an affine transform T to x we get

$$\mathbf{Tx} = \xi(\mathbf{Te10} - \mathbf{Te00}) + \eta(\mathbf{Te01} - \mathbf{Te00}) + \mathbf{Te00}$$




Thus **Tx** has the same affine coordinates (ξ,η) with respect to the transformed basis **(Te00, Te01, Te10).**

DS.G.4

Affine transforms include:

- Translation
- Rotation
- Scaling
- Skewing

So if I select a 3-point basis E for object M
E = (e00, e01, e10)
and I transform M by transformation T to M'
and also transform my basis E to TE
TE = (Te00, Te01, Te10)
then if the affine coordinates of point x of M are **(ξ,η),** the affine coordinates of corresponding point Tx of M' are the same **(ξ,η).**

We can use quantized (ξ,η) as two-dimensional hash table indexes for object recognition.

DS.G.5

**Offline Preprocessing**

Let D be a (large) database of models.

Let H be an initially empty hash table.

```
procedure GH_Preprocessing(D, H)
{
for each model M
  {
  Extract the feature point set FM of M;
  for each noncollinear triple E of points in FM
    for each other point x of FM
      {
      Calculate (ξ,η) for x wrt E;
      Store (M,E) in H in bin indexed by (ξ,η) ;
      }
  }
}
```

DS.G.6

**Online Recognition**

Let H be the hash table.
Let A be an accumulator array for voting.

```
procedure GH_Recognition(H,A)
{
Initialize A to all zeroes;
Extract feature points FP from image;
for each basis triple F of FP
  for each other point v
    {
    Calculate (ξ,η) for v wrt F;
    Retrieve the list L of model-basis pairs
        from the hash table H at index (ξ,η);
    for each pair (M,E) of L
      A[M,E] = A[M,E] + 1;
    }
for each peak (M,E) in accumulator array A
  {
  Calculate T such that F = TE;
  if (verify(T, M, FP) return T;
  }
}
```

DS.G.7

**Complexity**

Preprocessing:
For s models and n points in each

$T(s,n) = s * n^3 * n = O(sn^4)$

(s: models; $n^3$: bases; n: points in a model)

Matching:

Best Case. You try one triple and it works.

Worst Case. You try every possible triple on the image and none of them work.

DS.G.8

s: models
n: points per model
a: average length of a list in the hash table
h: number of high-valued accumulators

Best Case:

$a * n + h * n$
(voting: $a * n$; verification: $h * n$)

Worst Case:

$a * n^4 + h * n^4$

The worst case would be to try all bases, to get some high-valued accumulators for all of them, and to try to verify all of them.