



WIKIPEDIA
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

[What links here](#)

[Related changes](#)

[Upload file](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Wikidata item](#)

[Cite this page](#)

Print/export

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

Languages



Article [Talk](#)

[Read](#) [Edit](#)

[More▼](#)



Geometric hashing

From Wikipedia, the free encyclopedia

Not to be confused with [Geohashing](#).

In [computer science](#), **geometric hashing** is originally a method for efficiently finding two-dimensional objects represented by discrete points that have undergone an [affine transformation](#) (example below is based on similarity transformation), though extensions exist to some other object representations and transformations. In an off-line step, the objects are encoded by treating each pairs of points as a geometric [basis](#). The remaining points can be represented in an [invariant](#) fashion with respect to this basis using two parameters. For each point, its [quantized](#) transformed coordinates are stored in the [hash table](#) as a key, and indices of the basis points as a value. Then a new pair of basis points is selected, and the process is repeated. In the on-line (recognition) step, randomly selected pairs of data points are considered as candidate bases. For each candidate basis, the remaining data points are encoded according to the basis and possible correspondences from the object are found in the previously constructed table. The candidate basis is accepted if a sufficiently large number of the data points index a consistent object basis.

Geometric hashing was originally suggested in [computer vision](#) for [object recognition](#) in 2D and 3D,^[1] but later was applied to different problems such as [structural alignment](#) of [proteins](#).^[*citation needed*]

Contents [hide]

1 Geometric Hashing in Computer Vision

1.1 Example

1.1.1 Training Phase

1.1.2 Recognition Phase

1.2 Finding mirrored pattern

1.3 Geometric hashing in higher-dimensions

2 References

Geometric Hashing in Computer Vision [edit]

Geometric Hashing is a method used for object recognition. Let's say that we want to check if a model image can be seen in an input image. This can be accomplished with geometric hashing. The method could be used to recognize one of the multiple objects in a base, in this case the hash table should store not only the pose information but also the index of object model in the base.

Example [edit]

For simplicity, this example will not use too many [point features](#) and assume that their descriptors are given by their coordinates only (in practice [local descriptors](#) such as [SIFT](#) could be used for indexing).

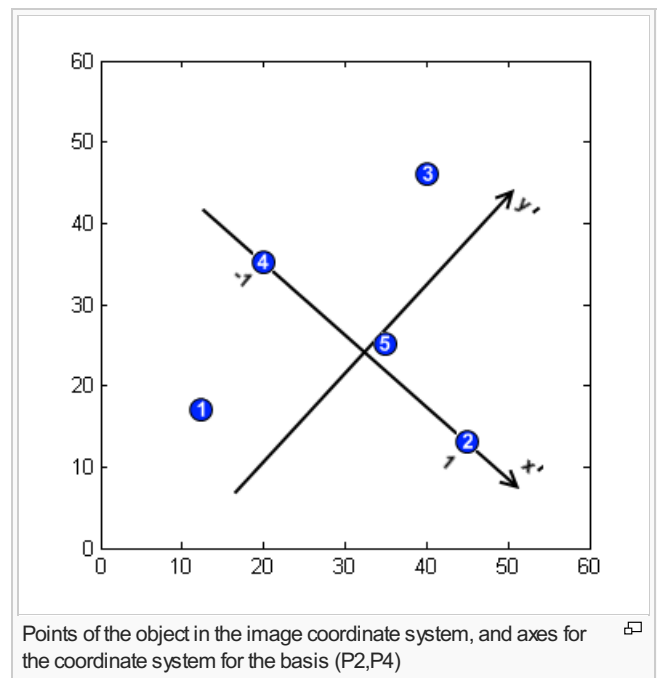
Training Phase [edit]

- Find the model's feature points. Assume that 5 feature points are found in the model image with the coordinates $(12, 17)$; $(45, 13)$; $(40, 46)$; $(20, 35)$; $(35, 25)$, see the picture.
- Introduce a basis to describe the locations of the feature points. For 2D space and [similarity transformation](#) the basis is defined by a pair of points. The point of origin is placed in the middle of the segment connecting the two points (P2, P4 in our example), the x' axis is directed towards one of them, the y' is orthogonal and goes through the origin. The scale is selected such that absolute value of x' for both basis points is 1.
- Describe feature locations with respect to that basis, i.e. compute the projections to the new coordinate axes. The coordinates should be discretised to make recognition [robust](#) to noise, we take the bin size 0.25. We thus get the coordinates $(-0.75, -1.25)$; $(1.00, 0.00)$; $(-0.50, 1.25)$; $(-1.00, 0.00)$; $(0.00, 0.25)$
- Store the basis in a [hash table](#) indexed by the features (only transformed coordinates in this case). If there were more objects to match with, we should also store the object number along with the basis pair.
- Repeat the process for a different basis pair (Step 2). It is needed to handle [occlusions](#). Ideally, all the

non-**colinear** pairs should be enumerated. We provide the hash table after two iterations, the pair (P1, P3) is selected for the second one.

Hash Table:

Vector (x' , y')	basis
$(-0.75, -1.25)$;	(P2,P4)
$(1.00, 0.00)$;	(P2,P4)
$(-0.50, 1.25)$;	(P2,P4)
$(-1.00, 0.00)$;	(P2,P4)
$(0.00, 0.25)$	(P2,P4)
$(1.00, 0.00)$;	(P1,P3)
$(0.00, 1.25)$;	(P1,P3)
$(-1.00, 0.00)$;	(P1,P3)
$(0.00, -0.25)$;	(P1,P3)
$(0.00, 0.50)$	(P1,P3)



Most hash tables cannot have identical keys mapped to different values. So in real life one won't encode basis keys $(1.0, 0.0)$ and $(-1.0, 0.0)$ in a hash table.

Recognition Phase [\[edit\]](#)

1. Find interesting feature points in the input image.
2. Choose an arbitrary basis. If there isn't a suitable arbitrary basis, then it is likely that the input image does not contain the target object.
3. Describe coordinates of the feature points in the new basis. Quantize obtained coordinates as it was done before.
4. Compare all the transformed point features in the input image with the hash table. If the point features are identical or similar, then increase the count for the corresponding basis (and the type of object, if any).
5. For each basis such that the count exceeds a certain threshold, verify the hypothesis that it corresponds to an image basis chosen in Step 2. Transfer the image coordinate system to the model one (for the supposed object) and try to match them. If succeed, the object is found. Otherwise, go back to Step 2.

Finding mirrored pattern [\[edit\]](#)

It seems that this method is only capable of handling scaling, translation, and rotation. However, the input Image may contain the object in mirror transform. Therefore, geometric hashing should be able to find the object, too. In fact, there are two ways to detect mirrored objects.

1. For the vector graph, make the left side as positive, and the right side as negative. Or multiplying the x position by -1 will give the same result.
2. Use 3 points for the basis. This allows detecting mirror images (or objects). Actually, using 3 points for the basis is another approach for geometric hashing.

Geometric hashing in higher-dimensions [\[edit\]](#)

Similar to the example above, hashing applies to higher-dimensional data. For three-dimensional data points, three points are also needed for the basis. The first two points define the x-axis, and the third point defines the y-axis (with the first point). The z-axis is perpendicular to the created axis using the right-hand rule. Notice that the order of the points affects the resulting basis

References [\[edit\]](#)

1. [^] A.S. Mian, M. Bennamoun, and R. Owens, [Three-dimensional model-based object recognition and segmentation in cluttered scenes](#), IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, Oct. 2006, pp. 1584-601.
- Wolfson, H.J. & Rigoutsos, I (1997). [Geometric Hashing: An Overview](#). IEEE Computational Science and Engineering, 4(4), 10-21.

Categories: [Geometric data structures](#) | [Search algorithms](#) | [Computer vision](#)

This page was last modified on 31 August 2015, at 13:15.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

