

Title: Geometric feature learning

URL: [https://en.wikipedia.org/wiki/Geometric\\_feature\\_learning](https://en.wikipedia.org/wiki/Geometric_feature_learning)

PageID: 34633465

Categories: Category:Applications of computer vision, Category:Feature detection (computer vision), Category:Machine learning

Source: Wikipedia (CC BY-SA 4.0). Content may require attribution.

-----

Geometric feature learning is a technique combining machine learning and computer vision to solve visual tasks. The main goal of this method is to find a set of representative features of geometric form to represent an object by collecting geometric features from images and learning them using efficient machine learning methods. Humans solve visual tasks and can give fast response to the environment by extracting perceptual information from what they see. Researchers simulate humans' ability of recognizing objects to solve computer vision problems. For example, M. Mata et al.(2002) applied feature learning techniques to the mobile robot navigation tasks in order to avoid obstacles. They used genetic algorithms for learning features and recognizing objects (figures). Geometric feature learning methods can not only solve recognition problems but also predict subsequent actions by analyzing a set of sequential input sensory images, usually some extracting features of images. Through learning, some hypothesis of the next action are given and according to the probability of each hypothesis give a most probable action. This technique is widely used in the area of artificial intelligence .

#### Introduction

Geometric feature learning methods extract distinctive geometric features from images. Geometric features are features of objects constructed by a set of geometric elements like points, lines, curves or surfaces. These features can be corner features, edge features, Blobs, Ridges, salient points image texture and so on, which can be detected by feature detection methods.

#### Geometric features

**Corners:** Corners are a very simple but significant feature of objects. Especially, Complex objects usually have different corner features with each other. Corners of an object can be extracted through Corner detection . Cho and Dunn used a different way to define a corner by the distance and angle between two straight line segments. This is a new way by defining features as a parameterized composition of several components.

**Edges:** Edges are one-dimensional structure features of an image. They represent the boundary of different image regions. The outline of an object can be easily detected by finding the edge using the technique of edge detection .

**Blobs:** Blobs represent regions of images, which can be detected using blob detection method.

**Ridges:** From a practical viewpoint, a ridge can be thought of as a one-dimensional curve that represents an axis of symmetry. Ridges detection method-see ridge detection

salient points-see Kadir–Brady saliency detector

image texture

#### Geometric composition

Geometric component feature is a combination of several primitive features and it always consists more than 2 primitive features like edges, corners or blobs. Extracting geometric feature vector at location  $x$  can be computed according to the reference point, which is shown below:

$x$  means the location of the location of features,  $\theta$   $\{\displaystyle \textstyle \theta\}$  means the orientation,  $\sigma$   $\{\displaystyle \textstyle \sigma\}$  means the intrinsic scale.

#### Boolean Composition

Boolean compound feature consists of two sub-features which can be primitive features or compound features. There are two type of boolean features: conjunctive feature whose value is the product of two sub-features and disjunctive features whose value is the maximum of the two sub-features.

Feature space

Feature space was firstly considered in computer vision area by Segen. He used multilevel graph to represent the geometric relations of local features.

Learning algorithms

There are many learning algorithms which can be applied to learn to find distinctive features of objects in an image. Learning can be incremental, meaning that the object classes can be added at any time.

Geometric feature extraction methods

Corner detection

Curve fitting

Edge detection

Global structure extraction

Feature histograms

Line detection

Connected-component labeling

Image texture

Motion estimation

Feature learning algorithm

1.Acquire a new training image "I".

2.According to the recognition algorithm, evaluate the result. If the result is true, new object classes are recognised.

recognition algorithm

The key point of recognition algorithm is to find the most distinctive features among all features of all classes. So using below equation to maximise the feature  $f_{\max}$

Measure the value of a feature in images,  $f_{\max}$  and  $f_{\max}$ , and localise a feature:

Where  $f_{\{p\}}(x)$  is defined as  $f_{\{p\}}(I) = \max \{ 0, f_{\{p\}}(T) \}$   
 $f_{\{p\}}(I) = \max \left\{ 0, \frac{f_{\{p\}}(T) - f_{\{p\}}(x)}{f_{\{p\}}(T) - f_{\{p\}}(x)} \right\}$

evaluation

After recognise the features, the results should be evaluated to determine whether the classes can be recognised, There are five evaluation categories of recognition results: correct, wrong, ambiguous, confused and ignorant. When the evaluation is correct, add a new training image and train it. If the recognition failed, the feature nodes should be maximise their distinctive power which is defined by the Kolmogorov-Smirno distance (KSD).

3.Feature learning algorithm

After a feature is recognised, it should be applied to Bayesian network to recognise the image, using the feature learning algorithm to test.

The main purpose of feature learning algorithm is to find a new feature from sample image to test whether the classes are recognised or not. Two cases should be consider: Searching for new feature of true class and wrong class from sample image respectively. If new feature of true class is detected and the wrong class is not recognised, then the class is recognised and the algorithm should terminate. If feature of true class is not detected and of false class is detected in the sample image, false class should be prevented from being recognised and the feature should be removed from Bayesian network.

Using Bayesian network to realise the test process

PAC model based feature learning algorithm

Learning framework

The probably approximately correct (PAC) model was applied by D. Roth (2002) to solve computer vision problem by developing a distribution-free learning theory based on this model. This theory heavily relied on the development of feature-efficient learning approach. The goal of this algorithm is to learn an object represented by some geometric features in an image. The input is a feature vector and the output is 1 which means successfully detect the object or 0 otherwise. The main point of this learning approach is collecting representative elements which can represent the object through a function and testing by recognising an object from image to find the representation with high probability.

The learning algorithm aims to predict whether the learned target concept  $f_T(X)$  belongs to a class, where  $X$  is the instance space consists with parameters and then test whether the prediction is correct.

Evaluation framework

After learning features, there should be some evaluation algorithms to evaluate the learning algorithms. D. Roth applied two learning algorithms:

SNoW-Train Initial step: initial the set of features  $F_t = \phi$  which linked to target  $t$  for all  $t \in T$ .  $T$  is a set of object targets whose elements are  $t_1$  to  $t_k$ . If each target object in set  $T$  belongs to a list of active features, link feature to target and set initial weight at the same time. Evaluate the targets : compare targets  $\sum_{i \in e} w_{it}$  with  $\theta_t$ , where  $w_{it}$  is the weight on one position connecting the features  $i$  to target  $t$ .  $\theta_t$  is the threshold for the target not  $t$ . Update weight according to the result of evaluation. There are two cases: predicted positive on negative example ( $\sum_{i \in e} w_{it} > \theta_t$ ) and targets are not in the list of active features) and predicted negative on positive example ( $\sum_{i \in e} w_{it} \leq \theta_t$  and targets are in the list of active features).

Initial step: initial the set of features  $F_t = \phi$  which linked to target  $t$  for all  $t \in T$ .  $T$  is a set of object targets whose elements are  $t_1$  to  $t_k$ .

If each target object in set  $T$  belongs to a list of active features, link feature to target and set initial weight at the same time.

Evaluate the targets : compare targets  $\sum_{i \in e} w_{it}$  with  $\theta_t$ , where  $w_{it}$  is the weight on one position connecting the features  $i$  to target  $t$ .  $\theta_t$  is the threshold for the target not  $t$ .

Update weight according to the result of evaluation. There are two cases: predicted positive on negative example ( $\sum_{i \in e} w_{it} > \theta_t$ ) and targets are not in the list of active features) and predicted negative on positive example ( $\sum_{i \in e} w_{it} \leq \theta_t$  and targets are in the list of active features).

SNoW-Evaluation Evaluate the each target using same function as introduced above Prediction:  
Make a decision to select the dominant active target node.

Evaluate the each target using same function as introduced above

Prediction: Make a decision to select the dominant active target node.

The main purpose of SVM is to find a hyperplane to separate the set of samples  $(x_i, y_i)$  where  $x_i$  is an input vector which is a selection of features  $x \in \mathbb{R}^N$  and  $y_i$  is the label of  $x_i$ . The hyperplane has the following form:  $f(x) = \text{sgn}(\sum_{i=1}^l y_i \alpha_i \cdot k(x, x_i) + b) = \{1, \text{positive inputs} - 1, \text{negative inputs}\}$   

$$f(x) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i \cdot k(x, x_i) + b\right) = \begin{cases} 1, & \text{positive inputs} \\ -1, & \text{negative inputs} \end{cases}$$

$k(x, x_i) = \phi(x) \cdot \phi(x_i)$  is a kernel function

Both algorithms separate training data by finding a linear function.

Applications

Landmarks learning for topological navigation

Simulation of detecting object process of human vision behaviour

Learning self-generated action

Vehicle tracking

References