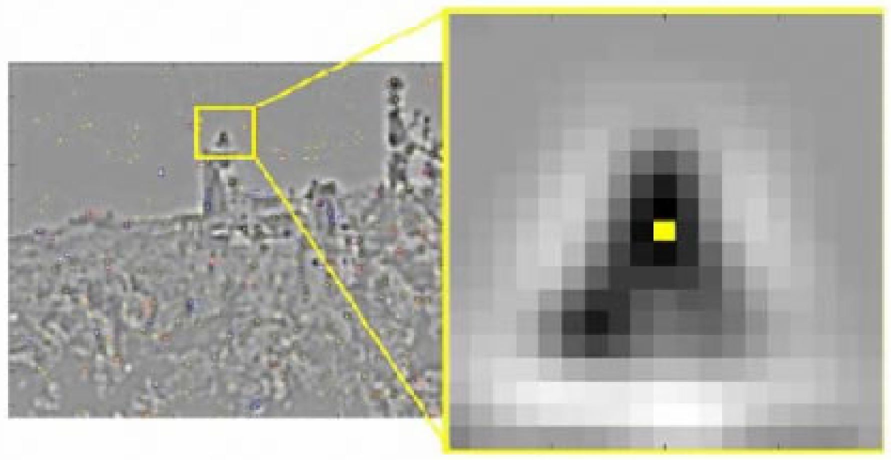
SIFT: Keypoint orientations

After step 4 (/tutorials/sift-scale-invariant-feature-transform-eliminate-low-contrast/), we have legitimate key points. They've been tested to be stable. We already know the scale at which the keypoint was detected (it's the same as the scale of the blurred image). So we have scale invariance. The next thing is to assign an orientation to each keypoint. This orientation provides rotation invariance. The more invariance you have the better it is. :P

The idea

The idea is to collect gradient directions and magnitudes around each keypoint. Then we figure out the most prominent orientation(s) in that region. And we assign this orientation(s) to the keypoint.

Any later calculations are done relative to this orientation. This ensures rotation invariance.

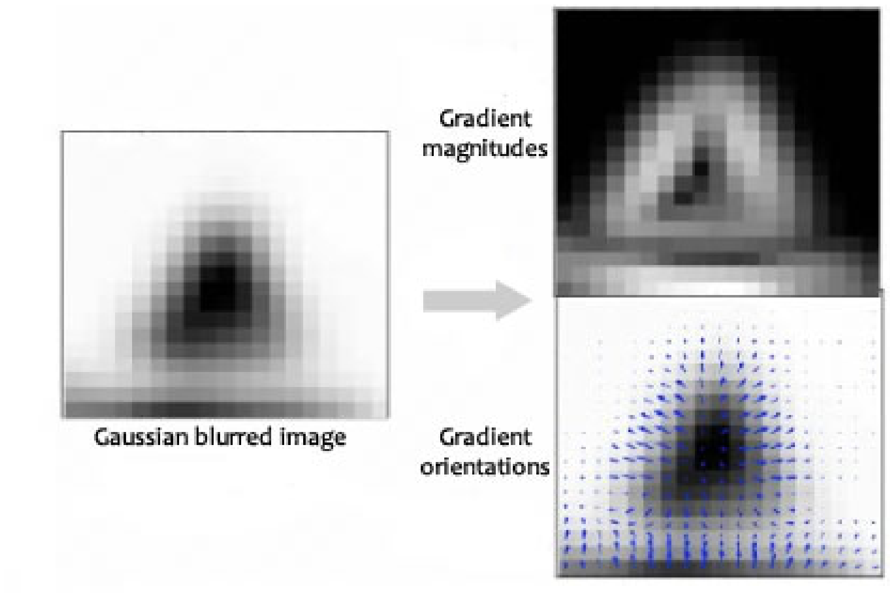


A keypoint

The size of the "orientation collection region" around the keypoint depends on it's scale. The bigger the scale, the bigger the collection region.

The details

Now for the little details about collecting orientations.



Gradient magnitudes and orientations are calculated using these formulae:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

The magnitude and orientation is calculated for all pixels around the keypoint. Then, A histogram (/tutorials/histograms-from-simplest-to-the-most-complex/) is created for this.

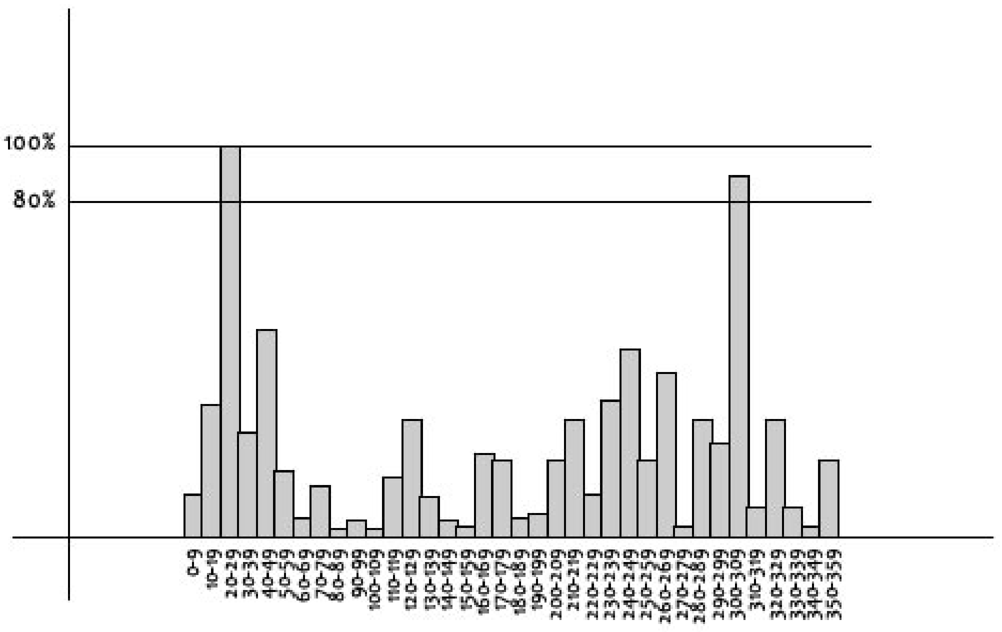
In this histogram, the 360 degrees of orientation are broken into 36 bins (each 10 degrees). Lets say the gradient direction at a certain point (in the "orientation collection region") is 18.759 degrees, then it will go into the 10-19 degree bin. And the "amount" that is added to the bin is proportional to the magnitude of gradient at that point.

Once you've done this for all pixels around the keypoint, the histogram will have a peak at some point.

Above, you see the histogram peaks at 20-29 degrees. So, the keypoint is assigned orientation 3 (the third bin)

Also, any peaks above 80% of the highest peak are converted into a new keypoint. This new keypoint has the same location and scale as the original. But it's orientation is equal to the other peak.

So, orientation can split up one keypoint into multiple keypoints.



The Technical Details

Magnitudes

Saw the gradient magnitude image above? In SIFT, you need to blur it by an amount of 1.5*sigma.

Size of the window

The window size, or the "orientation collection region", is equal to the size of the kernel for Gaussian Blur of amount 1.5*sigma.

Summary

To assign an orientation we use a histogram and a small region around it. Using the histogram, the most prominent gradient orientation(s) are identified. If there is only one peak, it is assigned to the keypoint. If there are multiple peaks above the 80% mark, they are all converted into a new keypoint (with their respective orientations).

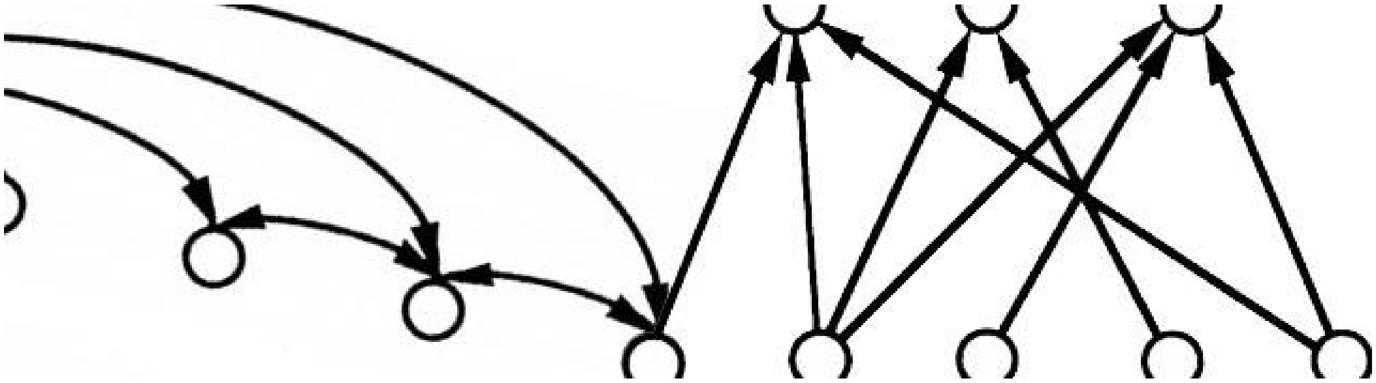
Next, we generate a highly distinctive "fingerprint" for each keypoint. Here's a little teaser. This fingerprint, or "feature vector", has 128 different numbers.

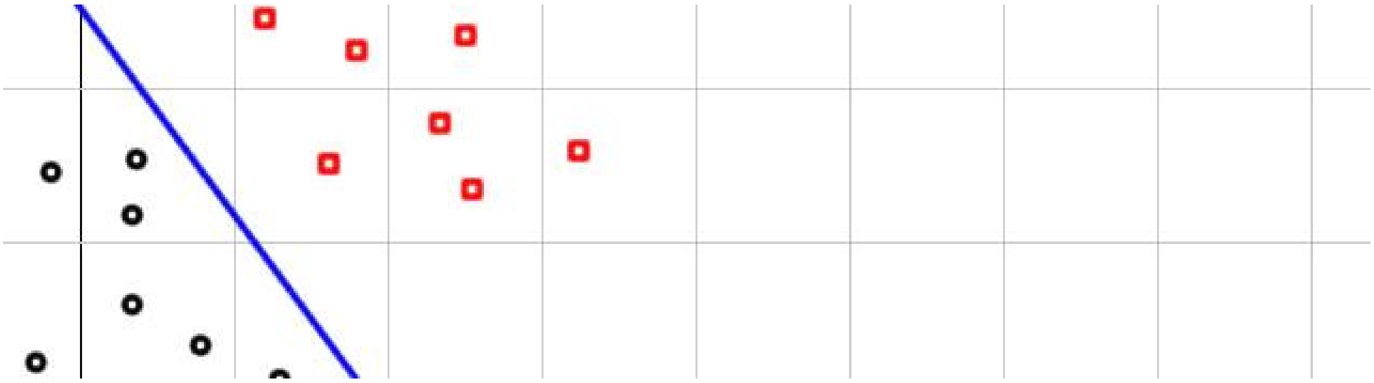
This tutorial is part of a series called **SIFT: Theory and Practice**:

1. SIFT: Introduction (/tutorials/sift-scale-invariant-feature-transform-introduction/)
2. SIFT: The scale space (/tutorials/sift-scale-invariant-feature-transform-scale-space/)
3. SIFT: LoG approximations (/tutorials/sift-scale-invariant-feature-transform-log-approximation/)
4. SIFT: Finding keypoints (/tutorials/sift-scale-invariant-feature-transform-keypoints/)
5. SIFT: Getting rid of low contrast keypoints (/tutorials/sift-scale-invariant-feature-transform-eliminate-low-contrast/)
6. **SIFT: Keypoint orientations**

7. SIFT: Generating a feature (/tutorials/sift-scale-invariant-feature-transform-features/)

Related posts


7 unique neural network architectures (/tutorials/7-unique-neural-network-architectures/)


A single neuron Dictomizer (/tutorials/a-single-neuron-dictomizer/)

Capturing images with DirectX (/tutorials/capturing-images-with-directx/)

(http://utkarshsinha.com/)

Utkarsh Sinha created AI Shack in 2010 and has since been working on computer vision and related fields as a hobby! Currently he works as a writer of software in Bangalore, India.

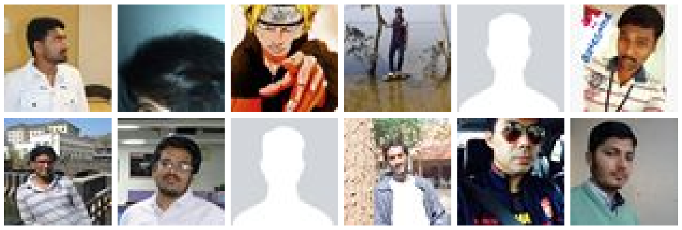
The bottom sidebar

AI Shack on Facebook

AI Shack

Like

2,756 people like AI Shack.



Facebook social plugin

Contact

You can contact me through the internet over the following channels:

Facebook (http://facebook.com/aishack/)
Twitter (http://twitter.com/aishack)
Github (http://github.com/aishack)
Email (mailto:utkarsh@utkarshsinha.com)

Where to start?

So you want to get started with AI but you're not sure where. Here are some links to get you started:

- Get started with OpenCV (/tracks/opencv-basics/)
- Track a specific color on video (/tutorials/tracking-colored-objects-in-opencv/)
- Learn basic image processing algorithms (/tracks/image-processing-algorithms-level-1/)
- How to build artificial neurons? (/tutorials/a-single-neuron-dictomizer/)
- Look at some source code (http://github.com/aishack/)

(/)