Page No.
Date

For Orientation Assignment

* SURF uses wavelet responses in horizontal and vertical direction for a neighbourhood of size 6s. Adequate gaussian weights are also applied to it.

* The dominant orientation is estimated by calculating the sum of all responses can be found out using integral images very easily at any scale.

* The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. Wavelet response could be found out using integral images very easily at any scale.

* For many applications, rotation invariance is not needed; hence, no needed of finding the orientation, which speeds up the process.

* SURF provides such a functionality called Upright-SURF or U-SURF. It improves speed and is robust of upto ±15°.

Page No.
Date

* For Feature Description,

→ SURF uses wavelet response in horizontal and vertical direction. A neighbourhood of size 20s x 20s is taken around the keypoint where s is the size.

→ It is divided into 4x4 subregions. For each subregions, horizontal and vertical wavelet responses are taken and a vector is formed like this,

$$v = (\Sigma d_x, \Sigma d_y, \Sigma |d_x|, \Sigma |d_y|)$$

→ This when represented as a vector will give SURF feature descriptor 64 features or dimensions.
Hence, lower the dimension, higher the speed of computation and matching but provide better distinctiveness of features.

* Another improvement over SIFT approach is that, the use of sign of Laplacian (trace of Hessian Matrix) for underlying interest point. It adds no computation cost since it is already computed during detection.

Page No.
Date

→ The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. In the matching stage, we only compare features if they have the same type of contrast. This allows for faster matching, without reducing the descriptor's performance.

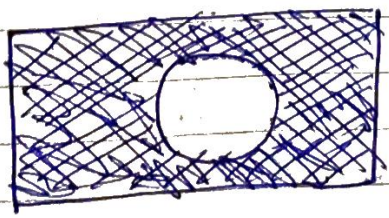
* SURF is 3 times faster than SIFT while performance is comparable to SIFT. SURF is good at handling images with blurring and rotation, but not good at ~~bad~~ handling viewpoint change and illumination change.

* Histogram of Oriented Gradience (HOG)

→ used for Object detection

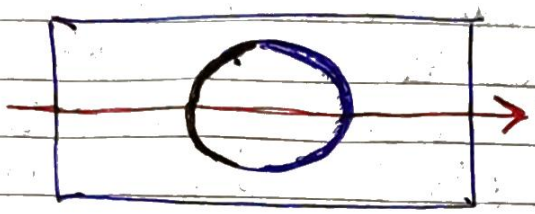
Intuition:

we have an image, with white circle in between, and the rest black.



Page No.
Date

Now we, want the horizontal gradient of the image.



The part marked in black becomes brighter.

The white color, indicates a positive gradient, i.e., change from 0 (black) to 255 (white).

The boldly marked blue part is dark in the image, this tells us the negative gradient, that is bright to dark.

So, now let's assume that in an image, there is a pixel, (the center one)

	100	
70	✓	120
	50	

So the pixels around it are 100, 70, 120, 50 (assumption)

Page No.
Date

* Gradient in x direction:

⇒ 120 - 70 = 50

* Grad in y direction: 100 - 50 = 50

* Putting it together, [50, 50] feature vector

* After this, the magnitude and dir^n are calculated as follows:

Grad. magnitude = $\sqrt{(50)^2 + (50)^2} = 70.1$

Grad. Angle = $\tan^{-1}(50/50) = 45^\circ$

Now, we have a oriented gradient from the surrounding pixels.

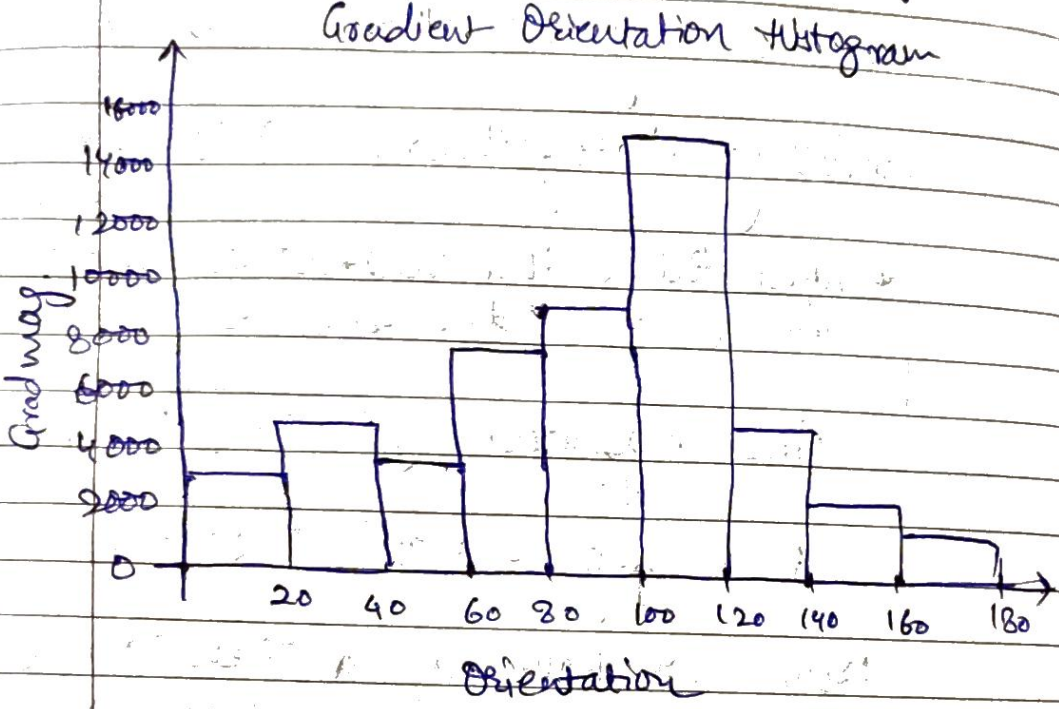
Oriented gradient is basically gradients of with directions.

Thus, we have oriented gradients, now we want histogram of this, so we follow the following steps:

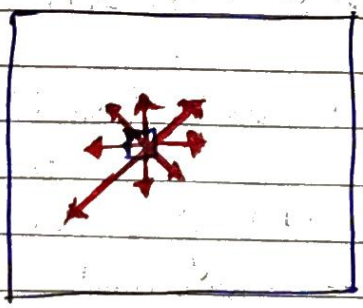
→ Assuming we have an image, we use 8x8 pixel cell, compute gradient mag/direction from SOBEL-X (grad in x dir^n) and SOBEL-Y (grad in y dir^n)

→ Each cell is split into angular bins,

each bin corresponding to a gradient direction (9 bins, 0-180° (20° each bin)
→ This reduces 64 vectors to just 9 values



So, now we can compact the above histogram in a single vector representation.



Hence, this makes a kind of dotted image of diff. grad with diff. mag & angles and that's how our machines sees an image.

Page No.
Date

To summarize,

→ HOGs are a feature descriptor used for object detection.
→ HOGs along with SVM classifiers work well for obj. detection.
→ HOG applied a sliding window detector over an image and HOG descriptor is computed for each position.
→ HOG takes care of the scaling issues by changing image scale (pyramiding).