

Deep Learning for Computer Vision
Professor Vineeth N Balasubramanian
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad
Lecture 10
Scale Space, Image Pyramids and Filter Banks

Moving on from the last lecture, we get into Scale Space, Image, Pyramids and Filter Banks in this one. If you recall, one of the limitations of the Harris Corner Detector that we stated the last time was that it is not scale invariant. That is, what would have been a corner in an image could have been an edge in another image which is zoomed in. Let us try to quickly recall this before we move forward.




(Refer Slide Time: 00:49)

The slide content includes a 'Review' header, two side-by-side images of a mountain range, the NPTEL logo, and a video thumbnail of Professor Vineeth N Balasubramanian. A footer bar contains the text 'Vineeth N B. (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks'.


Once again, an application where you would want to detect key points or corners is when you have two different images or even more number of images. Let us say, if you want to stitch these images together, this is typically called image mosaicking or this panorama building. So, let us say we have these two images here, which correspond to the same scene from two different locations. We really do not know the camera movement between these two images, but we want to stitch these two images. How do we go about it?

(Refer Slide Time: 1:28)

Review




Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks











We typically detect key points in each of these images independently. How do we do that? An example would be the Harris Corner Detector. We are on the Harris Corner Detector on image one, we run the Harris Corner Detector on image two, and then we match which key point or set of key points in image one matches which set of key points in image two. How do we do the matching? We will talk about it a little later in this course, but our focus now is on finding those key points.

(Refer Slide Time: 2:04)


Review



"flat" region: no change in all directions	"edge": no change along the edge direction	"corner": significant change in all directions
		
		
"flat" region: I_x and I_y are small.	"edge": $I_x \gg I_y$ $I_y \gg I_x$	"corner": I_x and I_y are large $I_x \approx I_y$



Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



And one method that we spoke for finding those key points is the Harris Corner Detector. We said that in the Harris Corner Detector, we build something called the autocorrelation matrix

and then take eigen decomposition of the autocorrelation matrix. And then we said that when λ_1 one and λ_2 , which are the two eigenvalues of your autocorrelation matrix a , are small, then it means that the region is flat, there is no change.

When one of the eigenvalues is much greater than the other, we say it is an edge and when both the eigenvalues are large, it corresponds that that particular patch has a lot of changes in multiple directions and we call such a point a corner. And that is what is our methodology for coming up with the Harris Corner Detector.

(Refer Slide Time: 2:59)

Review

NPTEL

flat region: no change in all directions

edge: no change along the edge direction

corner: significant change in all directions

flat region: λ_1 and λ_2 are small.

edge: $\lambda_1 \gg \lambda_2$, $\lambda_2 \gg \lambda_3$

corner: λ_1 and λ_2 are large, $\lambda_1 \approx \lambda_2$

Vineeth N.B. (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

But some observations that we made towards the end is that the Harris Corner Detector is rotation invariant.

(Refer Slide Time: 3:09)

Review

"flat" region: no change in all directions
"edge": no change along the edge direction
"corner": significant change in all directions

"flat" region: λ_1 and λ_2 are small
"edge": $\lambda_2 \gg \lambda_1$, $\lambda_3 \gg \lambda_1$
"corner": λ_1 and λ_2 are large, $\lambda_3 = \lambda_2$

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

But the Harris Corner Detector is not necessarily scale invariant. Which means, what is a corner in one image need not be a corner in another image which is zoomed in where it could seem just like an edge.

(Refer Slide Time: 3:28)

Review

"flat" region: no change in all directions
"edge": no change along the edge direction
"corner": significant change in all directions

"flat" region: λ_1 and λ_2 are small
"edge": $\lambda_2 \gg \lambda_1$, $\lambda_3 \gg \lambda_1$
"corner": λ_1 and λ_2 are large, $\lambda_3 = \lambda_2$



Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

So, we are ideally looking for a setting where we can analyze both these image artifacts in different scales and be able to match them at the right scale and that is the way we would make the Harris Corner Detector scale invariant.


(Refer Slide Time: 3:45)

Scale-Invariant Interest Point Detection

- How can we independently select interest points in each image, such that detections are repeatable across different scales?
 - Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.



Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks





Before we go there, let us try to ask how we can independently select interest points in each image, such that the detections are repeatable across different scales. So, which means, there are two images at different scales. When we say scales, they are zoomed in differently, one of them is zoomed in a lot, one of them is say zoomed out. We ideally want to be able to detect a key point in both these images. Remember that if you take the patch size to be the same in both these cases, in one of those images which is zoomed in, a key point may just look like an edge when you zoom in a lot. How do we counter this?

A simple approach could be that we extract features at a variety of scales by using say, multiple resolutions in a pyramid and then we match features at the same level. That could be one of the simplest things that we can do.


(Refer Slide Time: 4:44)

Scale-Invariant Interest Point Detection

- How can we independently select interest points in each image, such that detections are repeatable across different scales?
 - Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
 - When does this work?



Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks





Where do you think this will actually work? If you thought carefully, you would find that as long as you match features at the same level, the properties of the Harris Corner Detector will only be compared at the same scale, but to be scale invariant, we ideally need to compare the Harris Cornerness measure, recall the Harris Cornerness measure. At a different scale in one image and the Harris Cornerness measure at a different scale in the next image. So, how do we do that?


(Refer Slide Time: 5:24)

Scale-Invariant Interest Point Detection

- How can we independently select interest points in each image, such that detections are repeatable across different scales?
 - Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
 - When does this work?
 - More efficient to extract features stable in both location and scale.



Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks




So, what we try to do now is to extract features that are stable in both location and scale and we are going to try to describe how we are going to do that over the next few minutes.

(Refer Slide Time: 5:39)


Scale-Invariant Interest Point Detection

- How can we independently select interest points in each image, such that detections are repeatable across different scales?
 - Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
 - When does this work?
 - More efficient to extract features stable in both location and scale.
 - Find scale that gives local maxima of a function f in both position and scale.



$$f(I_{b-a}(x, \sigma)) = f(I_{b-a}(x', \sigma'))$$

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks




So, if you have two images, notice again that we have two images now, which definitely differ in scale. In one of them, this inside artifact are paintings and one of them this insight artifact is quite small and then we are zooming into that artifact on the right image. We now ideally want to find a corner, which is indicated by the yellow cross there. We want to find the same corner in both these images irrespective of the scale differences. How do we go ahead and do that? So, we want to find a function f which gives you a maximum at both x and sigma, the sigma is denoted as the scale of the image in this context.

(Refer Slide Time: 6:28)


Automatic Scale Selection

Function responses for increasing scale (scale signature).



Credit: R Urtasun

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



And the way we are going to go about doing it is, we compute your scale signature, in this case, it could be the Harris Cornerness measure. At that particular point for a particular scale and let us say that particular point has a particular Harris Cornerness measure, which is plotted on a graph.

(Refer Slide Time: 6:50)

The slide is titled "Automatic Scale Selection" and contains the following elements:


- Title:** Automatic Scale Selection
- Text:** Function responses for increasing scale (scale signature).
- Visuals:** A video frame showing a person in a white shirt, a graph with a red circle highlighting a peak, and a red arrow pointing to a specific point on the graph.
- Logos:** NPTEL logo and IIT-H logo.
- Footnote:** Credit: R Urtasun, Vineeth N B. (IIT-H), §2.3 Scale Space, Image Pyramids, Filter Banks.
- Footer:** For accessing this content for free (no charge), visit : nptel.ac.in

We then change the scale. In our case, a simple way to change the scale is simply to take a larger patch for your autocorrelation window. So, if you take a larger patch for your autocorrelation window and now take your Harris Cornerness measure, you are going to get a slightly different value for the Harris Cornerness measure. So, remember in the x axis, we are measuring scale, so we have changed the scale, which is the size of your autocorrelation window and we now get a different Cornerness measure.

(Refer Slide Time: 7:23)

Automatic Scale Selection




Function responses for increasing scale (scale signature).



Credit: R Urtasun

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

For accessing this content for free (no charge), visit : npTEL.ac.in




And you do this for different scales. So, which means you again take a different patch size, compute the Cornerness measure for that patch size. Remember again, that from a definition of our Harris Cornerness measure, the autocorrelation matrix would change if the size of your patch changes. Remember again, that we did a summation with all the pixels and that will change now when the size of your patch changes.

(Refer Slide Time: 7:50)

Automatic Scale Selection




Function responses for increasing scale (scale signature).



Credit: R Urtasun

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

For accessing this content for free (no charge), visit : npTEL.ac.in



So, we do this for more scales, more scales, and you see that you will get such a graph when you do this for multiple scales. So, the takeaway from this graph is that we seem to be getting the maximum Cornerness measure or the maximum Cornerness response at a particular scale,

which is going to be important to us. At that particular scale is when the Cornerness measure is the highest for that particular key point in that given image. So, how do we take this forward?

(Refer Slide Time: 8:27)

Automatic Scale Selection

Function responses for increasing scale (scale signature).

Credit: R Urtasun

Vineeth N B. (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

For accessing this content for free (no charge), visit : npTEL.ac.in

We now take another image. So, in this case, as I said, that is the maximum, that is what we are showing in this particular slide.

(Refer Slide Time: 8:34)

Automatic Scale Selection

Function responses for increasing scale (scale signature).

Credit: R Urtasun

Vineeth N B. (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

For accessing this content for free (no charge), visit : npTEL.ac.in

Now, we take another image, which is the one on the right which is zoomed in version and perhaps there is a slight rotation of the first image. And now again at the first scale, compute

the Cornerness measure. At the second scale, compute the Cornerness measure and we repeat this process for all the different scale that we considered for the first image.

(Refer Slide Time: 9:02)

Automatic Scale Selection

Function responses for increasing scale (scale signature).

Credit: R Urtasun
Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

For accessing this content for free (no charge), visit : nptel.ac.in

And as we keep doing this, we are going to get another graph for the second image where the peak now is at a different scale.

(Refer Slide Time: 9:15)

Automatic Scale Selection

Function responses for increasing scale (scale signature).

Is there a better way to do this?

Credit: R Urtasun
Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

For accessing this content for free (no charge), visit : nptel.ac.in


The peak now is at a different scale which is denoted by that particular value. So which means now we have made some progress. We have been able to find what would be the Cornerness measure, the maximum Cornerness measure for a particular point, both in

location and scale. So, location would be the coordinate of that center of that patch and scale would be the scale at which we got the maximum cornerness measure. One question that we ask ourselves now is, is there a better way to implement this?

(Refer Slide Time: 9:51)

Automatic Scale Selection: Implementation


Instead of computing f for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid.



Sometimes need to create in-between levels, e.g., a $\frac{3}{4}$ -size image.

Credit: R Urtasun

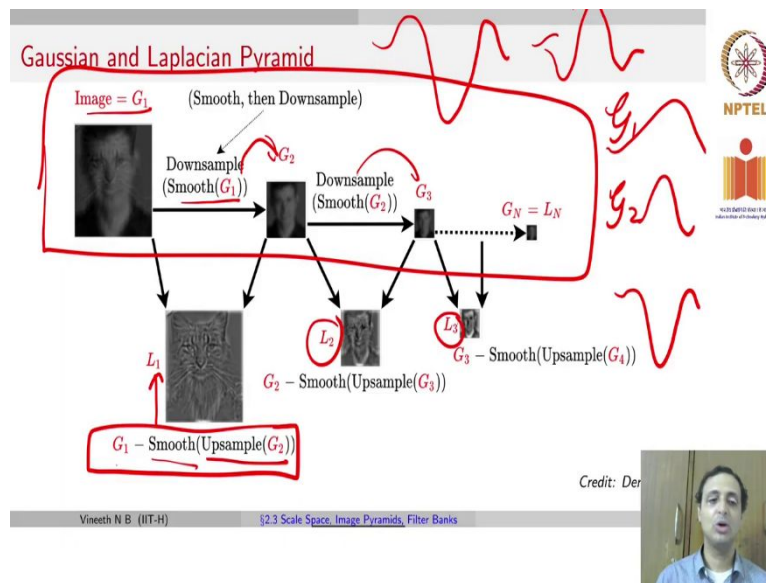
Vineeth N B. (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



The answer to that is to use what are known as image pyramids. Instead of changing your patch size in each of your images, you fix your patch size across any images you may encounter, but change your image size by doing a Gaussian pyramid, recall our discussion of Gaussian pyramids when we spoke about interpolation and frequencies. Remember a Gaussian pyramid is constructed by taking an image, Gaussian smoothing the image, then sub sampling the image and repeating this process again and again.

So, we do the same thing now and keep the patch size the same and construct a Gaussian pyramid. Keep in mind, that when you construct your Gaussian pyramid, it need not always be reduced by half each time, you can also reduce your sizes by say three fourth or by any other fraction by using interpolation methods.

(Refer Slide Time: 10:54)



When we consider an image pyramid, there are several kinds of pyramids that you can construct and use in practice. So, the Gaussian pyramid is what we have seen before, which is what the top part of this diagram shows, which is about taking the original image, let us call it G_1 . Then you smooth the image and then down sample it and then get a G_2 . Then you again smooth G_2 , down sample it, get a G_3 and you keep repeating this process.

There is also another way of getting another kind of a pyramid called Laplacian pyramid, which is obtained by, you take G_1 , which is your original image. Once you get your G_2 , which is your smoothed and down sample image you again up sample G_2 and again smooth it. Now, when you compute G_1 minus G_2 , that gives you a quantity called L_1 . The reason why we call L_1 a Laplacian pyramid is because Laplacian can be written as a difference of two Gaussians. Why so? Let us try to see it a little illustratively at this time.

So, recall the Laplacian filter that we discussed in the last lecture. A Laplacian filter could be drawn as something like this. This was one way of drawing it. Remember, we could also have drawn it the other way, where you have it as something like this. So, both are Laplacian pyramids depending on whether your central value is negative or positive. If you see purely from a graph perspective, this is a 1D Laplacian. If you will purely see from a graph perspective.

Such Laplacian can be written as the difference of one Gaussian, which is say wide, let us call that Laplacian some cursive g_1 and say another Gaussian, which is narrow, let us call it

Gaussian G_2 . When you subtract G_2 from G_1 , you will actually get a shape, which is similar to the Laplacian.

Clearly you will have to choose the variance for G_1 and G_2 appropriately to get the kind of Laplacian that you are looking for. And because in this particular example, G_1 minus the smoothed up sample version of G_2 turns out to be a difference of Gaussians, it effectively was done out to be some kind of a Laplacian which is why we call it a Laplacian pyramid.

And you repeat the Laplacian for every successive lower resolution representation in your Gaussian pyramid and get multiple L_2 s and L_3 s so on and so forth, to also get a Laplacian pyramid. For different applications, you may want to use a Gaussian pyramid or a Laplacian pyramid.

(Refer Slide Time: 14:02)

Image Pyramids: Uses

- Compression

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

But where do you use image pyramids in practice, in multiple applications? You could use it for compression because you may want to just transmit a low-resolution version of the original image and send some other information through other means and be able to reconstruct a high resolution from the low-resolution image.

(Refer Slide Time: 14:21)

Image Pyramids: Uses

- Compression
- Object detection
 - Scale search
 - Features

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

You could use an image pyramid for object detection. How and why? You could use it by doing a scale search and then doing some features. What we mean here is, you could look for an object firstly in a low resolution part of the pyramid and once you find the region of the image, where you get the object, then you go into the next high resolution, search in that region a bit more carefully, find where the object is and then, you can repeating this in high resolutions.

(Refer Slide Time: 14:55)

Image Pyramids: Uses

- Compression
- Object detection
 - Scale search
 - Features
- Detecting stable interest points
- Registration
 - Coarse-to-fine Image Registration

Image Pyramids, Filter Banks 7 / 17 Vineeth N B (IIT-H)

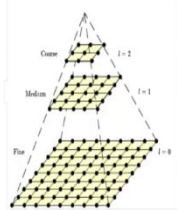
You can also use an image pyramid for stable interest points, which is what we have been discussing so far.

(Refer Slide Time: 15:02)

Image Pyramids: Uses

- Compression
- Object detection
 - Scale search
 - Features
- Detecting stable interest points
- Registration
 - Coarse-to-fine Image Registration




Coarse-to-fine Image Registration:



- Compute Gaussian pyramid.
- Align with coarse pyramid.
- Successively align with finer pyramids.
 - Search smaller range.

Credit: Derek Hoiem

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



Another application of image pyramids could be registration. In registration, is the process of aligning key points from two different images. How do you use image pyramids in registration? You can do what is known as coarse to fine image registration, where you start by constructing a pyramid for each image that you have.

So, you have a coarse level, a medium level and a fine level. And you first compute this Gaussian pyramid and then align features at the coarse pyramid level, just at this level to start with. Once you do that, you then continue successively aligning with final pyramids by only searching smaller ranges for that final match.




(Refer Slide Time: 15:59)

Texture in Images

Textures:

- Regular or stochastic patters caused by bumps, grooves and/or markings.

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



Moving on from the image pyramid, we will go to the third topic that we are covering in this lecture, which is the notion of textures, which is closely connected and built upon the other concepts that we are covering. What are textures to start with? Textures are regular or stochastic patterns that are caused by bumps, grooves and or markings, the way we literally call them textures.

(Refer Slide Time: 16:32)

Texture in Images

Textures:

- Regular or stochastic patterns caused by bumps, grooves and/or markings.
- Gives us information about the spatial arrangement of the colors or intensities in an image.

Credit: Derek Hoiem

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



The slide displays three pairs of texture images. The first pair, labeled 'Different Materials', shows a rough, fibrous texture on the left and a smooth, vertically striped texture on the right. The second pair, labeled 'Different Orientation', shows a vertically striped texture on the left and a horizontally striped texture on the right. The third pair, labeled 'Different Scales', shows a coarse, brick-like texture on the left and a fine, brick-like texture on the right. The NPTEL logo is in the top right, and a small portrait of a man is in the bottom right. Footer text includes 'Credit: Derek Hoiem', 'Vineeth N B (IIT-H)', and '§2.3 Scale Space, Image Pyramids, Filter Banks'.

So, these textures give us some information about the spatial arrangement of colors or intensities in an image. On the right side, you will see that textures can give you an idea of materials, textures can give you an idea of the orientation. Textures can also give you an idea of the scale that you are dealing with. So, textures contain significant information to be able to make higher level decisions or predictions from images.

(Refer Slide Time: 17:02)

Texture in Images

Conveys more information that can be exploited to match regions of interest in images.


Histogram conveys 50% white pixels and 50% black pixels

(Block Pattern) (Checkerboard Pattern) (Striped Pattern)

Drastically different textures

Credit: Linda G Shapiro

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



NPTEL

KJ Somaiya Institute of Technology

It is also important to keep in mind that even if you had a single image. Let us say you obtained a high-level statistic, such as the histogram of an image containing 50 percent white pixels and 50 percent black pixels. In this scenario, we could have images of multiple kinds, three samples of what you see on the slide. You could have the image to be something like this. You could have the image to be something like this, or you could have the image to be something like this.

In all these three cases, the histogram contains 50 percent white and 50 percent black, but the textures are vastly different. So, it is not only important to get global statistics, it is also important to get local texture information to be able to understand what is in images.

(Refer Slide Time: 17:59)

Texture in Images

Conveys more information that can be exploited to match regions of interest in images.

Histogram conveys 50% white pixels and 50% black pixels

How to represent textures?

(Block Pattern) (Checkerboard Pattern) (Striped Pattern)

Drastically different textures

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

The slide features a histogram at the top with a white bar on the left and a black bar on the right, representing a 50/50 pixel distribution. Below it, three patterns are shown: a solid black block, a checkerboard pattern, and a striped pattern. The slide also includes the NPTEL logo and a small video inset of the speaker.

So, how do we actually represent textures? Let me let you think for a moment. So far, we have seen edges, we have seen corners, we have seen corners at different scales. How do we represent textures? The answer is you put together whatever you have seen so far. And how do we put them together?

(Refer Slide Time: 18:26)

Texture in Images

Conveys more information that can be exploited to match regions of interest in images.

Histogram conveys 50% white pixels and 50% black pixels

How to represent textures?

- Compute responses of blobs and edges at various orientations and scales.
- Ways to process:
 - Record simple statistics (e.g., mean, std.) of absolute filter responses.
 - Take vectors of filter responses at each pixel and cluster them.

(Block Pattern) (Checkerboard Pattern) (Striped Pattern)

Drastically different textures

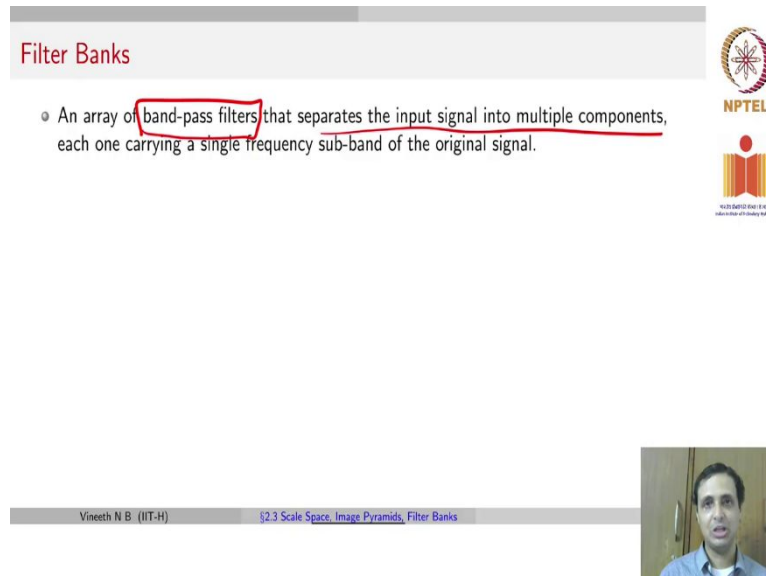
Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks

This slide is similar to the previous one but includes a list of methods for representing textures. The list includes computing responses of blobs and edges, and two ways to process these responses: recording simple statistics and taking vectors of filter responses for clustering. The slide also includes the NPTEL logo and a small video inset of the speaker.

We compute responses of blobs and edges at different orientations and scales and that is one way of getting textures. So, the way we process an image is we record simple statistics, such as mean and standard deviation of absolute filter responses of an image. And then we could take the vectors of filtered responses at each pixel and cluster them to be able to represent

your textures. There are multiple ways of doing this, but that could be the general process of capturing the textures and images. We will see a couple of examples of how texture can be captured in an image.

(Refer Slide Time: 19:11)



The slide is titled "Filter Banks" in red text. Below the title is a bullet point: "An array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal." The words "band-pass filters" are circled in red, and the rest of the sentence is underlined in red. To the right of the text are two logos: the NPTEL logo (a circular emblem with a book and a lamp) and the IIT Bombay logo (a stylized book icon). At the bottom of the slide, there is a grey bar with the text "Vineeth N.B. (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks" on the left and a small video inset of a man speaking on the right.

A simple way to do this is by what is known as filter banks. Filter banks are as the word says, a bank of filters. We are not going to use just a Sobel filter or a Harris Corner Detector or Laplacian to compute blobs, we are going to use a set of different filters, a bank of different filters. And what do each of these filters do?

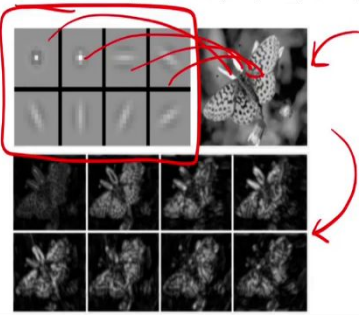
Each of these filters can be viewed as what are known as band-pass filters. This goes back to our discussion on extracting low-frequency components and high-frequency components in images. Band-pass filters are filters that allow a certain band of frequencies to pass through and get as output when you convolve a filter with the image.

So, remember we have seen examples of filters that extract high-frequency components, edge detection. We can also be opposite to get low-frequency components by doing Gaussian smoothing. At this point, with band-pass filters, we are saying that we want only certain set of frequencies to pass through, and we are going to use a bank of such filters to be able to separate the input signal into multiple components, each one carrying a certain sub band of your original signal image, and that can be used to represent the texture in your image.

(Refer Slide Time: 20:40)




Filter Banks

- An array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal.
- Process image with each filter and keep responses (or squared/abs responses).



Credit: Derek Hoiem

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks





Here is a visual illustration. So, you process an image with different filters. So, you see here eight different filters that you can come up with. This is your input image. So, you convolve each filter on the image with the image and these are the responses that you get when you convolve each of those filters with the input image. As you can see, each of these outputs capture different aspects of the texture or the content in that butterfly, and they all put together give you a sense of what is the texture in the image.


(Refer Slide Time: 21:23)

Gabor Filters

- Special classes of band pass filters (i.e., they allow a certain 'band' of frequencies and reject the others).



Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



We will talk about a more concrete example, which are known as Gabor filters. Gabor filters are a very popular set of band-pass filters. At a certain level, they are known to mimic or

mimic how the human visual system works. But they allow a certain band of frequencies and reject the others.

(Refer Slide Time: 21:46)

Gabor Filters


- Special classes of band pass filters (i.e., they allow a certain 'band' of frequencies and reject the others).
- A Gabor filter can be viewed as a sinusoidal signal of particular frequency and orientation, modulated by a Gaussian wave.

A 2-D Gaussian


A sinusoid oriented 30° with x-axis

A corresponding 2-D Gabor Filter


A 2-D Gabor filter obtained by modulating the sine wave with a Gaussian



NPTEL



NPTEL
National Program on Technology Enhanced Learning

Vineeth N B (IIT-H)
§2.3 Scale Space, Image Pyramids, Filter Banks


The way Gabor filters work is intuitively, they can be seen as a combination of a Gaussian filter and a sinusoidal filter. So, here is an example of a sinusoidal filter for certain orientation. Here is an example of a Gaussian filter. If you convolve a Gaussian filter and a sinusoidal filter, you would get something like this. Imagine superimposing your sinusoid on your Gaussian, you would get something like this.

(Refer Slide Time: 22:20)

2-D Gabor Filter


$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = e^{-\left(\frac{x'^2 + y'^2}{2\sigma^2}\right)} e^{i\left(2\pi\frac{y'}{\lambda} + \psi\right)}$$

where:


$x' = x \cos \theta + y \sin \theta$

$y' = -x \sin \theta + y \cos \theta$


θ – Orientation of the normal to the parallel stripes of Gabor function.



NPTEL



NPTEL
National Program on Technology Enhanced Learning

Vineeth N B (IIT-H)
§2.3 Scale Space, Image Pyramids, Filter Banks


Mathematically speaking, a 2D Gabor filter can be written as you have an x , y , you have a λ , θ , ψ , σ and γ , we will talk about each of them in a moment. And it is given by

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = e^{-\left(\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)} e^{i\left(2\pi\frac{x'}{\lambda} + \psi\right)}$$

We will talk about each of those quantities, we are not going to derive this in this particular course, that may be outside the scope.

But in this particular formula, $x' = x\cos(\theta) + y\sin(\theta)$, we will talk about what theta is. θ is the orientation of the normal to the parallel stripes of the Gabor. We saw that the sinusoid could be oriented in a particular direction and that is given by theta. So, $x' = x\cos(\theta) + y\sin(\theta)$, $y' = -x\sin(\theta) + y\cos(\theta)$



(Refer Slide Time: 23:24)


2-D Gabor Filter

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = e^{-\left(\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)} e^{i\left(2\pi\frac{x'}{\lambda} + \psi\right)}$$

where:

- $x' = x \cos \theta + y \sin \theta$
- $y' = -x \sin \theta + y \cos \theta$
- θ – Orientation of the normal to the parallel stripes of Gabor function.
- λ – Wavelength of the sinusoidal component.
- ψ – Phase offset of the sinusoidal function.
- σ – Standard deviation of the Gaussian envelope.
- γ – Spatial aspect ratio and specifies the ellipticity of the support of Gabor function.

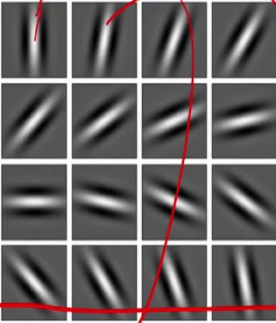
Vineeth N B (IIT-H)
§2.3 Scale Space, Image Pyramids, Filter Banks


λ is the wavelength of your sinusoidal component. Remember your sinusoid has a wavelength and a frequency. So, your λ wavelength, ψ is the phase offset of your sinusoidal function. Once again, recall our discussion on Mitch frequencies earlier. σ is the standard deviation of your Gaussian envelope.

And γ is a spatial aspect ratio and specifies the ellipticity of the support of your Gabor function. So, if you want to elongate it, all of them can be controlled in this particular context. Instead of having a circular Gaussian, you can use the gamma parameter to be able to control the elliptical nature of your Gabor response function.



(Refer Slide Time: 24:15)

Gabor Filter Banks



Bank of 16 Gabor filters at an orientation of 11.250 (i.e. if the first filter is at 00.00, then the second will be at 11.25, the third will be at 22.50, and so on.)

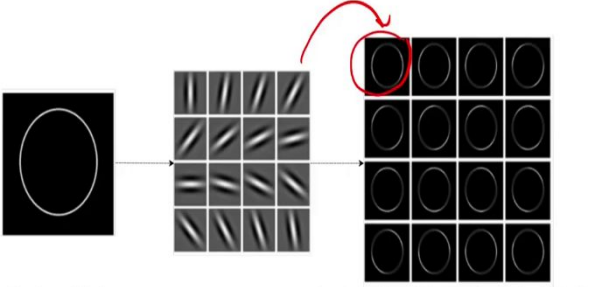
Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



So, this is a 2D Gabor filter. As you can see, it gives you an idea of certain textures. So, here is a filter bank of Gabor filters. So, this has 16 Gabor filters at an orientation of 11.25, which means, if your first filter has an orientation of 0, your next filter will be 11.25, the next filter will be 22.5, so on and so forth. And you can see the Gabor filter being rotated and you now have an entire bank of Gabor filters.



(Refer Slide Time: 24:48)

Gabor Filter Banks



Input Image of a Circle A bank of 16 Gabor filters Output Image of the circle after passed through individual Gabor filters

Vineeth N B (IIT-H) §2.3 Scale Space, Image Pyramids, Filter Banks



You can now take an image and convolve each of these filters with the image and you will get 16 different responses of the image to these 16 different filters. As you can see here, each of these responses capture a certain aspect of your original image. In case of a circle, they

simply seems to highlight a different perspective to the circle, but when you have more complex textures, each of these responses captures a certain dimension of that texture.

(Refer Slide Time: 25:24)

Steerable Filter Banks


Steerable Filters are a class of oriented filters that can be expressed as a linear combination of a set of basis filters.

- For an isotropic Gaussian filter, $G(x, y) = e^{-(x^2+y^2)}$


$$G_1^{\theta} = G_1^{0^\circ} \cos(\theta) + G_1^{90^\circ} \sin(\theta)$$

where G_1^{θ} is the first derivative of G at angle θ .


original




Y-axis




15°




X-axis




30°



60°



Vineeth N B. (IIT-H)
§2.3 Scale Space, Image Pyramids, Filter Banks



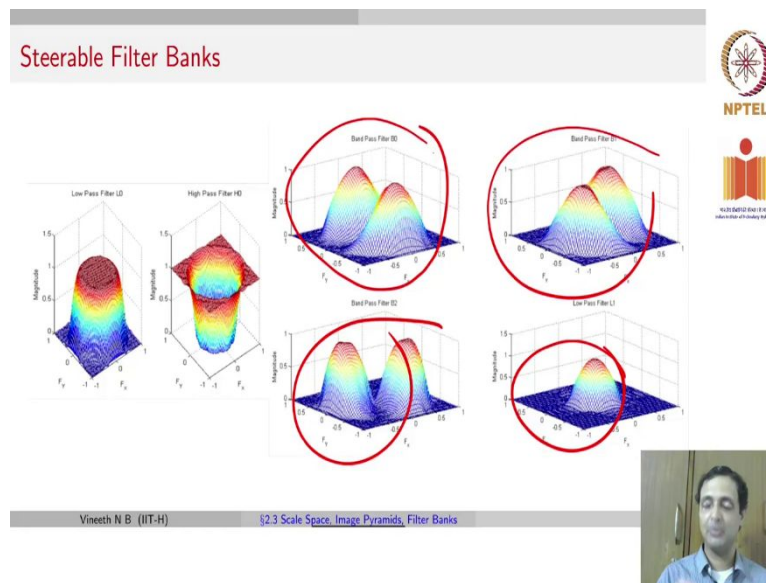
And putting these together gives us an overall response of the image to different set of orientations and frequencies. There has also been another popular set of filter banks called Steerable Filter Banks. Steerable filters are a class of oriented filters that can be expressed as a linear combination of a set of basis filters.

For example, if you have an isotropic Gaussian filter, $e^{-(x^2+y^2)}$, you can define a Steerable filter as you have $G_1^{\theta} = G_1^{0^\circ} \cos(\theta) + G_1^{90^\circ} \sin(\theta)$, where G_1^{θ} is the first derivative of G at a certain angle θ . For example, if you have an original image, you can now consider G_1 along the y axis to be the derivative at a particular angle.

You can consider G_1 of 15 degrees to be the derivative at a different angle and so on and so forth. So, now you can construct combinations of these two, of these different images to construct an overall response that you have. So, each of them is a Steerable filter where you can control the angle at which you are getting your response.

So, this is another, Gabor filter banks was one example that could be used to extract textures from images, Steerable filters banks are another example that could be used to extract textures from images.

(Refer Slide Time: 27:07)



Here is an example, another illustration of Steerable filter banks, where you can take a band-pass filter, B_0 . As you can see this band-pass filter allows a certain set of frequencies to pass through. Another band-pass filter B_1 , B_2 , so on and so forth. You can have a low-pass filter, so on and so forth.

Now, you can combine the responses of an image to all these kinds of filters and store some statistics at each pixel. So remember, you are going to get a value at each pixel, you can store the mean and standard deviation, you can cluster, you can do various things with those values that you get at each pixel across the filter banks, responses to the filter banks and be able to get a representation for your texture.

(Refer Slide Time: 27:53)

The screenshot shows a presentation slide with the following content:

- Homework Readings**
- Readings**
 - Chapter 2, Szeliski, *Computer Vision: Algorithms and Applications*
- Questions**
 - Why is camouflage attire effective? How?
 - How is texture different from noise?
 - Will scale-invariant filters be effective in matching pictures containing Matryoshka (or Russian nesting) dolls?

Logos for NPTEL and IIT-H are visible on the right side of the slide. Below the slide, there is an image of a set of Matryoshka dolls and a small video inset of a man speaking.

That concludes this lecture. Please do continue to read Chapter 2 in Szeliski's book. Some interesting questions for you to take away now, which you may not really answered, but it is something for you to think about is; From the discussions we have had in this lecture, why is a camouflage attire effective? Think about it. Obviously, it connects to our lecture, so think carefully on what we discussed and how you can extend it to understanding how a camouflage attire works.

Another question to ask here is, how is texture different from say a salt and pepper noise? A salt and pepper noise could also look like a texture. So, how is a texture different from a salt and pepper noise? Something for you to think about and read to understand. And a last question is, will scale invariant filters be effective in matching pictures, containing Matryoshka's dolls or I think we also have equivalence in India. Nesting dolls, can scale invariant filters be able to match pictures across these dolls? Think about these questions as your exercise for this lecture.