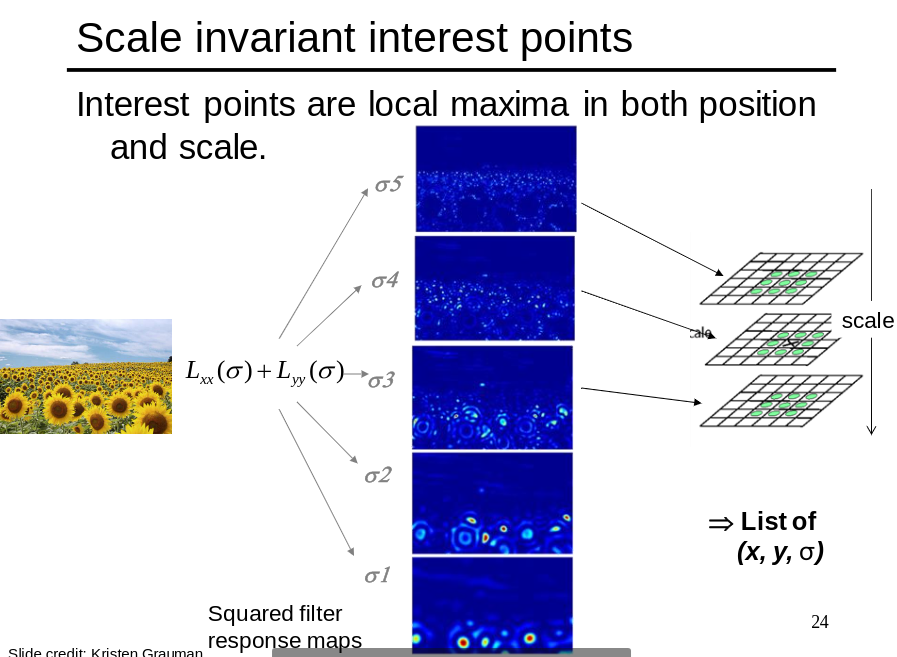
Collin Avidano

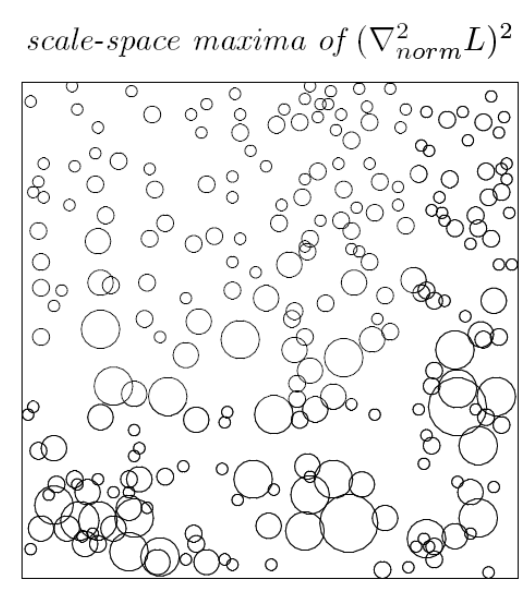
FRQs

1.1

When performing interest point detection with the Laplacian of Gaussian, how would the results differ if we were to (a) take any positions that are local maxima in scale-space, or (b) take any positions whose filter response exceeds a threshold? Specifically, what is the impact on repeatability or distinctiveness of the resulting interest points?



As noted in the slide above when we choose points of interest using the scale space local maxima we are not only finding the position at which there is a maximal response but also the scale that causes the maximal response for that location. This results in locations paired with the best resulting scale for that location which can be visualized as the image below:



Now let's consider the effects of using thresholding to determine the best scale for a particular location in the image. Looking back on the steps of the scale space blob detector we would run multiple varying sigma laplacians of gaussian filters over the image. Generally these levels are discrete values of sigma and only one size will have the best response for a given location, however other values of sigma near that best sigma value will still have high responses. Because of this, unless your threshold is perfect to the point it would cut out the next best sigma value response, that value of sigma would also be chosen. This would result in many concentric circles of expanding size at one location representing any sigma whose response for that region exceeds the threshold. In other words thresholding would lead to very non distinct detections of points of interest vs taking the local max.

You may reason that one could simply set the threshold higher to prevent such behavior but then you have another issue. Since the magnitude of a maximal response would vary throughout the image any threshold you set may successfully remove some redundant values of sigma for a particular position but may lead to a region with a lower response to any of the values of sigma to not have any points of interest chosen. In other words, using threshold holding is a bad idea because unless all the responses that you are interested in lie in a very narrow band you will not be able to get rid of every redundant value of sigma and may cut out some points of interest in the process. Additionally if you changed the discrete values of sigma used in filtersing to catch more unique points of interest you would need to retune your threshold value for that new set of sigmas, which leads to major issues in repeatability.

1.2

What is an “inlier” when using RANSAC to solve for the epipolar lines for stereo with uncalibrated views, and how do we compute those inliers?

In this case the inliers are the pairs of corresponding points which agree with an F calculated as part of an initial 8 point svd guess. This agreement line in normal ransac is the value of a distance function given the guessed F and a particular correspondence. For example you could use the reprojection error as your distance metric. Basically take the initial guessed F times the point see where it projects as a line onto the other image and calculate the distance from that line to the actual known correspondence x '. The inliers then become the correspondence pairs whose value under this op is below d. Then you use the original iterative process of ransac to find what F has the most correspondences pairs with a reprojection error under d.

1.3

Name and briefly explain two possible failure modes for dense stereo matching, where points are

matched using local appearance and correlation search within a window.

Texture-less or smooth surfaces and occlusion are two possible failure modes for stereo matching.

Occlusion: one point may not exist in the other camera because it is occluded because of their difference in perspective on the scene. Imagine it as when you are up close to a pole and you alternate which eye is open vs close you can see around the pole with one eye but not with the other and you lose accurate depth perception of things on the other side of the pole (because both eyes cant see the point).

Textureless or smooth surfaces: When trying to find correspondences on a smooth surface your ssd for the whole area is going to be similar and thus you cant reliably identify a pixel coordinate in such a region in between two images.

1.4

What exactly does the value recorded in a single dimension of a SIFT keypoint descriptor signify?

A sift descriptor is formed as a histogram of the gradient directions for a specific window size around a point. For example in a 8 bin sift descriptor a pixel with a gradient of 90 degrees and one with 88 degrees will both go into the same bin. Each one of these bins is a dimension of the sift descriptor. Therefore a single dimension of a SIFT keypoint descriptor describes the number of points in that window that have gradient direction angles within the bounds of that bin. In other words how many pixels in the region approximately have a gradient magnitude in that bins direction (you can represent the direction of the bin as the mid point of its thresholds).

1.5

If using SIFT with the Generalized Hough Transform to perform recognition of an object instance, what is the dimensionality of the Hough parameter space? Explain your answer.

With the Generalized Hough Transform and SIFT, we have subpatches that represent individual parts of the object we are trying to identify and we are trying to find where it is in the image or in other words what is the transform (scale, rotation, translation) to place the object in the image. Assuming that we are only estimating uniform scaling and a scalar for rotation we have 4 dimensions to our hough space. These are the x and y translation of the image in the scene, the rotation in radians, and the scale of the object. Each sift descriptor votes on the overall into this hough space knowing which patch it is in the original model image and the transform in that image to that patch and votes at the location for the center (or top left corner of the image depending on how the model was setup) in the new image.

2.1 Raw descriptor matching





2.2 Visualizing the vocabulary

Explanation:

On the left we see multiple different patches that have the highest similarity to our visual words. Each column represents one of the visual words and its best matches.

As we can see the visual word on the left is most representative of patches that have this one high gradient splitting the image (black bar) and where there is a darkened corner in the patch going diagonally across this region.

As for the visual word on the right its most representative of patches with two high gradient “sticks” into a triangle which converges to a point in the middle of the image



2.3

Frame 40

similarity\_scores [0.79848641 0.79031106 0.78404761 0.77399358 0.75873418]



Frame 540

similarity\_scores [0.71727457 0.71327592 0.7076043 0.55556159 0.54627556]





Frame 3000

similarity\_scores [0.58891826 0.56861297 0.50848761 0.50515182 0.50309436]



Explanation:

In many of the scenes this method correlated to very similar frames. These were the frames that came before or after the query frames. However as you can see there eventually are some very extraneous results. The fact they were not at least similar is probably because there weren't 5 frames that went in a continuous order from the same angle and contained the same scene and people. Additionally we see the similarity scores are rather low for many of these query frames. This is because of the issues with sift descriptors histograms we talked about in class. Since we are only mapping the number of sift descriptors that are close to a particular word vs another image, this approach lacks a sense of spatial awareness between the different features. Which harms our ability to recognize objects as a whole since our descriptors are generally edges.

2.4

Frame 80

indices [80 82 81 79 78]

similarity\_scores [0.41870402 0.35968592 0.35861702 0.33823529 0.33517463]





Frame 800

indices [ 800 803 801 802 5034]

similarity\_scores [0.8260542 0.63065943 0.62680887 0.60373953 0.4885677 ]





Frame 1200

indices [1200 1201 1198 1199 4664]

similarity\_scores [0.54074308 0.43217662 0.4176572 0.40643931 0.39772844]





Frame 4800

indices [4800 4798 4799 6348 3340]

similarity\_scores [0.66221368 0.46793717 0.46650802 0.44382423 0.44178103]





Explanation

As we saw by selecting a region this allowed us to match only a subset of the descriptors that would be common to the same scene in multiple angles or points in time. By only comparing the distributions of frames against the distributions of our vocabulary in a smaller region of interest we can target descriptors specific to an object we feel is unique. Consider my first and second queries above. The region targeted contained very sharp edges and text (which also has sharp edges). This leads to very unique descriptors which also would make their distribution fairly unique and at some points overlap.

However when we expand much further than just a single object of interest the same issues we saw with the whole scene comparison start to come back.

Consider the 4th query image frame 4800. In this image I selected the arrangement of objects on a table but this arrangement is in a way a clutter. That is unlike my first example the sift descriptors likely to be pulled are not unique / uniformly square, or could be associated with many different words.

Additionally the man's leg is included in the region of interest and as his leg moves out of the image we move to an entirely different scene. The reason I bring this up is it highlights the inability of the bag of words representation to understand the focus of the scene or even of a region of interest. The few descriptors that were pulled from his jeans change the histogram distribution of the region entirely. Which is certainly not stable.