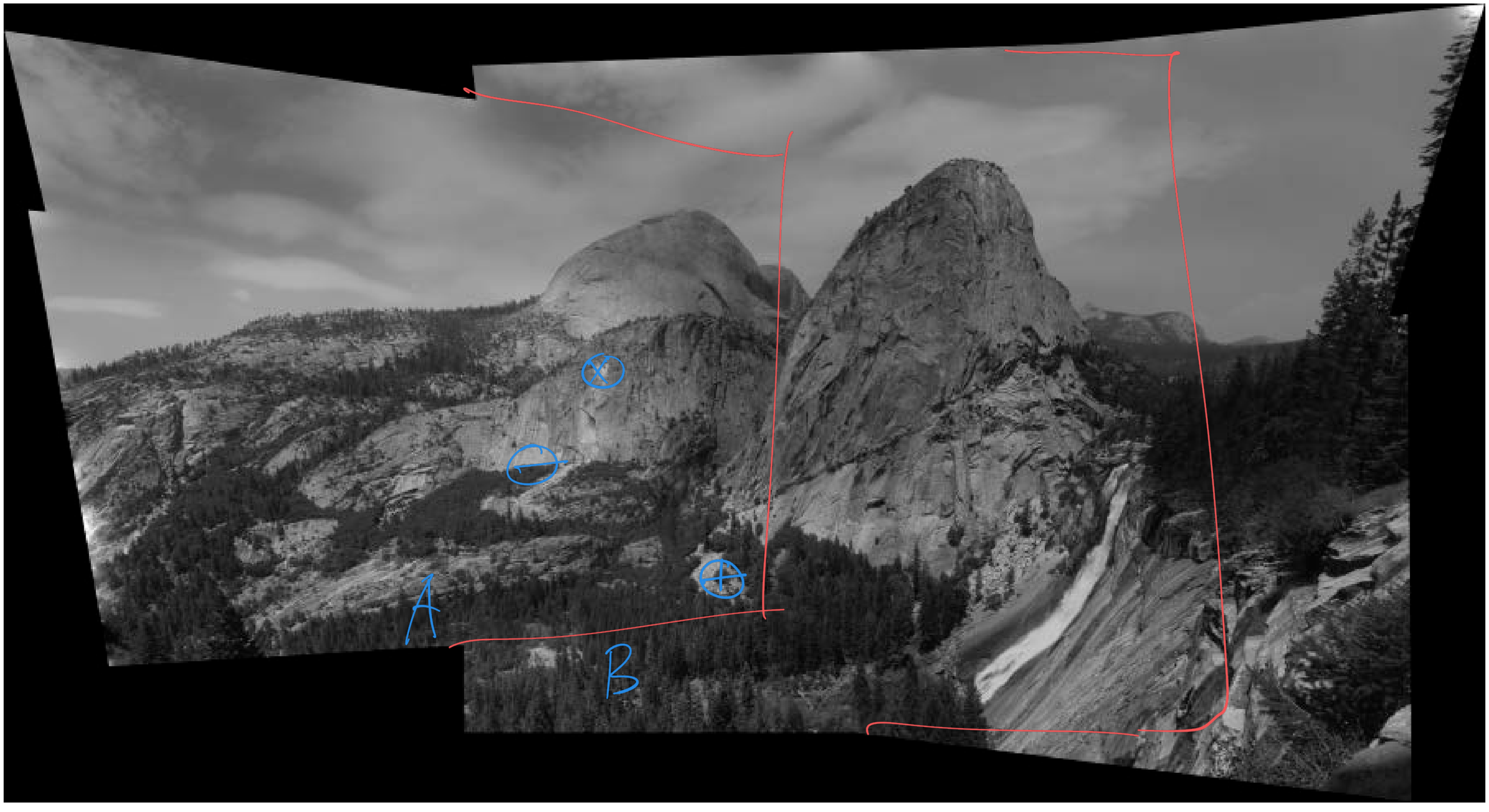
SIFT plus Scale space

Leo Dorst

Mosaicing Exercise



Distinctive Image Features from Scale-Invariant Keypoints

SIFT

David G. Lowe

- Famous paper from 2004 that changed the field of computer vision. Has 46,000+ citations.
- Good exercise for reading a scientific paper.
- Not too well written, but that happens.

How to Read a Paper B.1.1

- **Pass 0:** Read Abstract, Introduction, Conclusion to get the perspective (and decide if you care to read all)
- **Pass 1:** Read it all through once, trying to understand with ordinary effort, gloss over difficulty. Gives structure, focus, confirmation.
- **Pass 2:** In second pass, really understand relevant parts. Use the margins, references, annotate.
- **Pass 3 (optional):** If truly relevant, revisit a few weeks later. Everything will fall into place.

Personalize: Summarize on front in 5 bullets with + or -.

How to Read Lowe's Paper

- See Course Materials/Lecture Notes/how_to_read_lowe.pdf

Pass 0

- Read Abstract, Introduction and Conclusion
- 10 minutes vanaf 9:10
- Then we discuss

Abstract

This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

Introduction

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

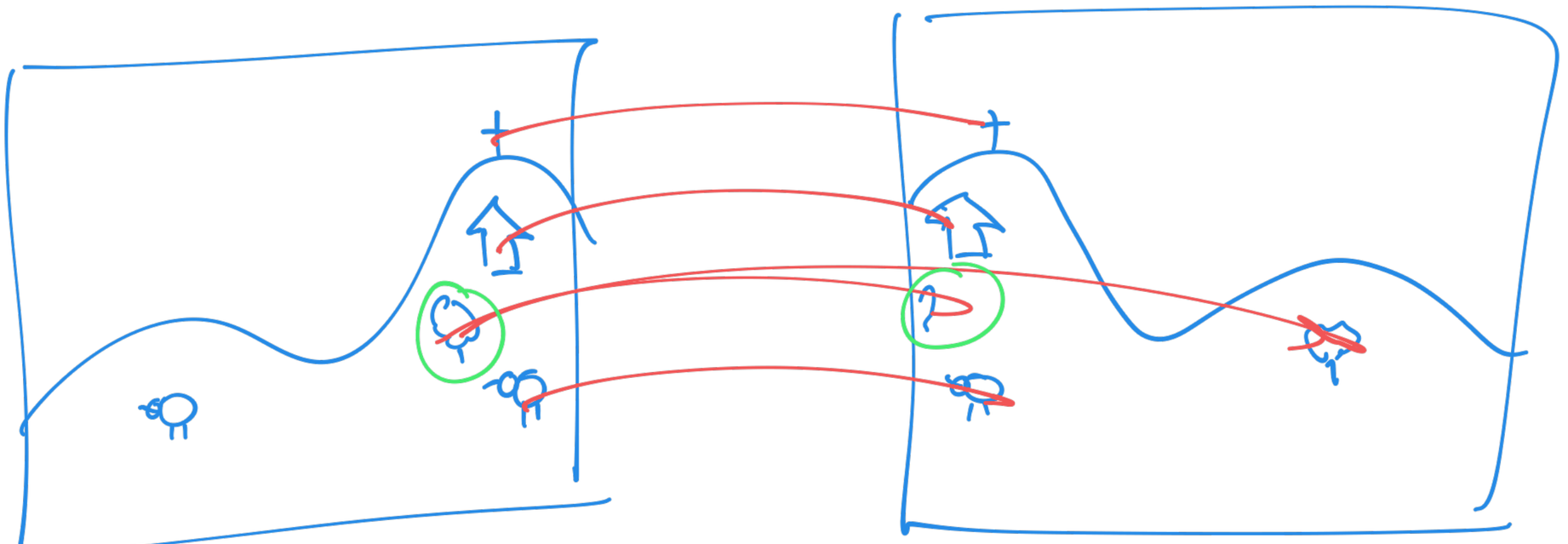
3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Lindeberg: http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform

Conclusions

- Works, has invariances
- Efficient, so practical
- Some applications
- More to be done



- Hough
- RANSAC

Random Sample Consensus

Pass 1, Section 3

- Let us skip 2 Related Research for now (gives context of quest for Invariant Features)
- Read 3.0 Detection of Scale Space Extrema till 3.1
- Take about 10 minutes (first fast, then slowly), then we'll talk. vanaf 9:30

From how_to_read_lowe.pdf

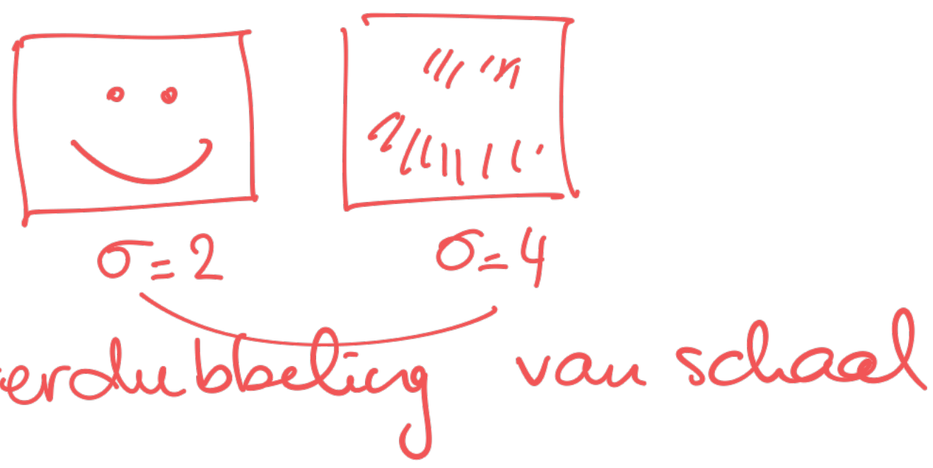
DoG

3. Detection of scale-space extrema

And we're off! Scale space is a bit new to you at this point, look it up or come to the lecture. But you know Gaussian convolution!

Your first pass of reading should tell you that the order of this section is messy. He want to use his own method (extrema of difference of Gaussians) because it is efficient. Oh, and actually it is a good approximation the Laplacian (look up what that is!). That is of course the really fundamental relationship to local structure. After having given (1), he derives it (he could/should have done that first), but does not quite return to 'his' D. A lab question asks you to close the gap.

What is an octave? Why does he 'need $s + 3$ images in the stack of blurred images for each octave'?



Scale and Diffusion

The Gaussian is Green's function for the heat equation.

$$\frac{\partial \phi(\mathbf{r},t)}{\partial t} = \sigma \nabla^2 \phi(\mathbf{r},t),$$

tijds afgeleide

ruimtelijke tweede afgeleide

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \nabla^2$$

nabla

Scale-space representation $L(x,y;t)$ at scale $t = 0$, corresponding to the original image f

Scale-space representation $L(x,y;t)$ at scale $t = 1$

Scale-space representation $L(x,y;t)$ at scale $t = 4$

Scale-space representation $L(x,y;t)$ at scale $t = 16$

Scale-space representation $L(x,y;t)$ at scale $t = 64$

Scale-space representation $L(x,y;t)$ at scale $t = 256$

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
Svenska

Article Discussion

Scale space

From Wikipedia, the free encyclopedia

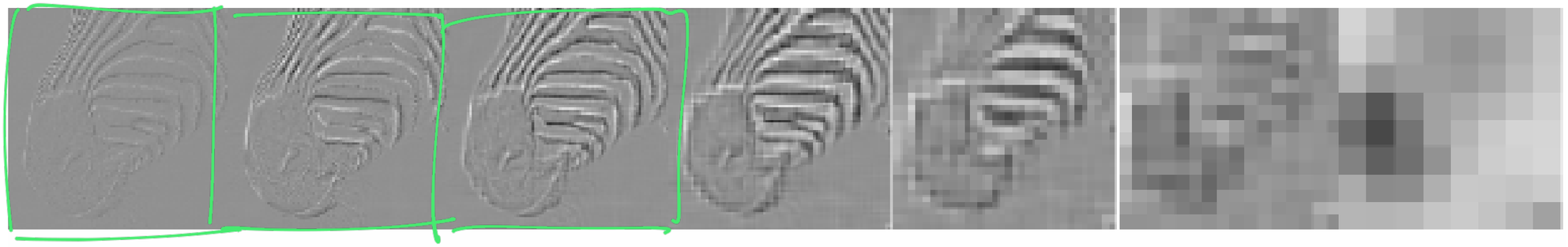
Scale-space theory is a framework for multi-scale signal representation developed by the handling image structures at different scales, by representing an image as a one-parame structures.[1][2][3][4][5][6][7] The parameter t in this family is referred to as the *scale param*

The main type of scale-space is the *linear (Gaussian) scale-space*, which has wide appli a theory for Gaussian derivative operators, which can be used as a basis for expressing a which is necessary for dealing with the size variations that may occur in image data, bec circumstances.[8]

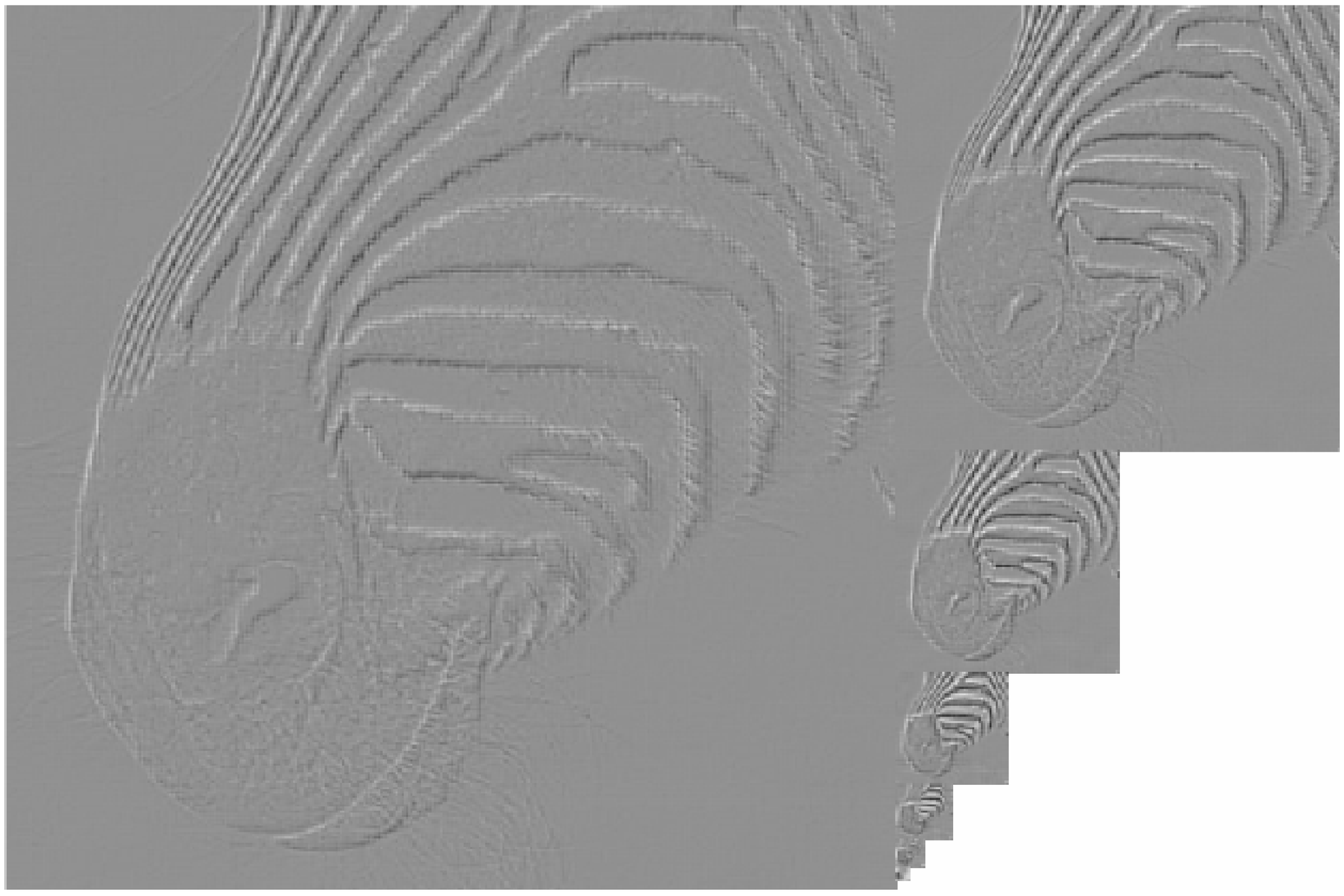
$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} = & f_{yy} \end{bmatrix}$$

Contents [hide]

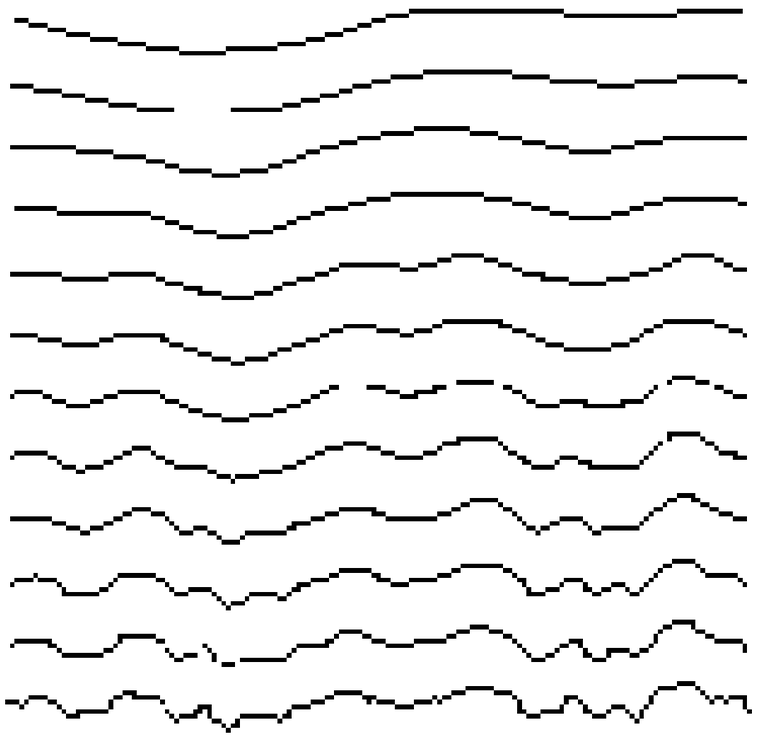
$$f_{xx} = \frac{\partial^2}{\partial x^2} f$$

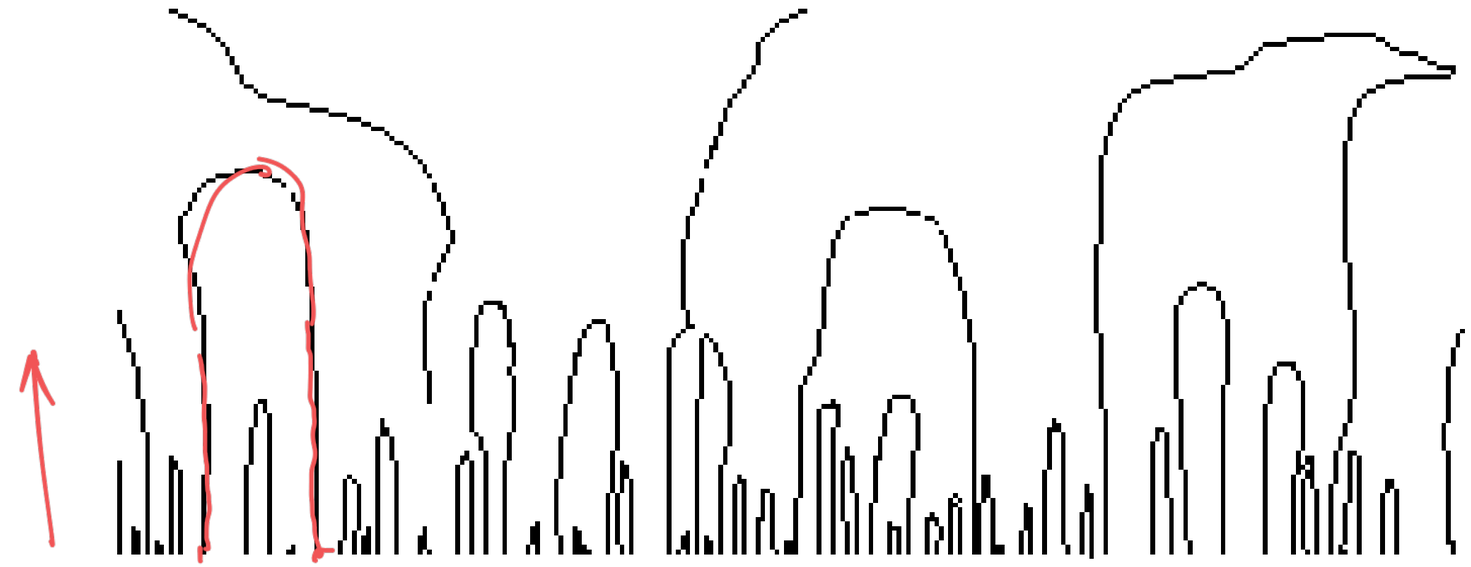


512 256 128 64 32 16 8



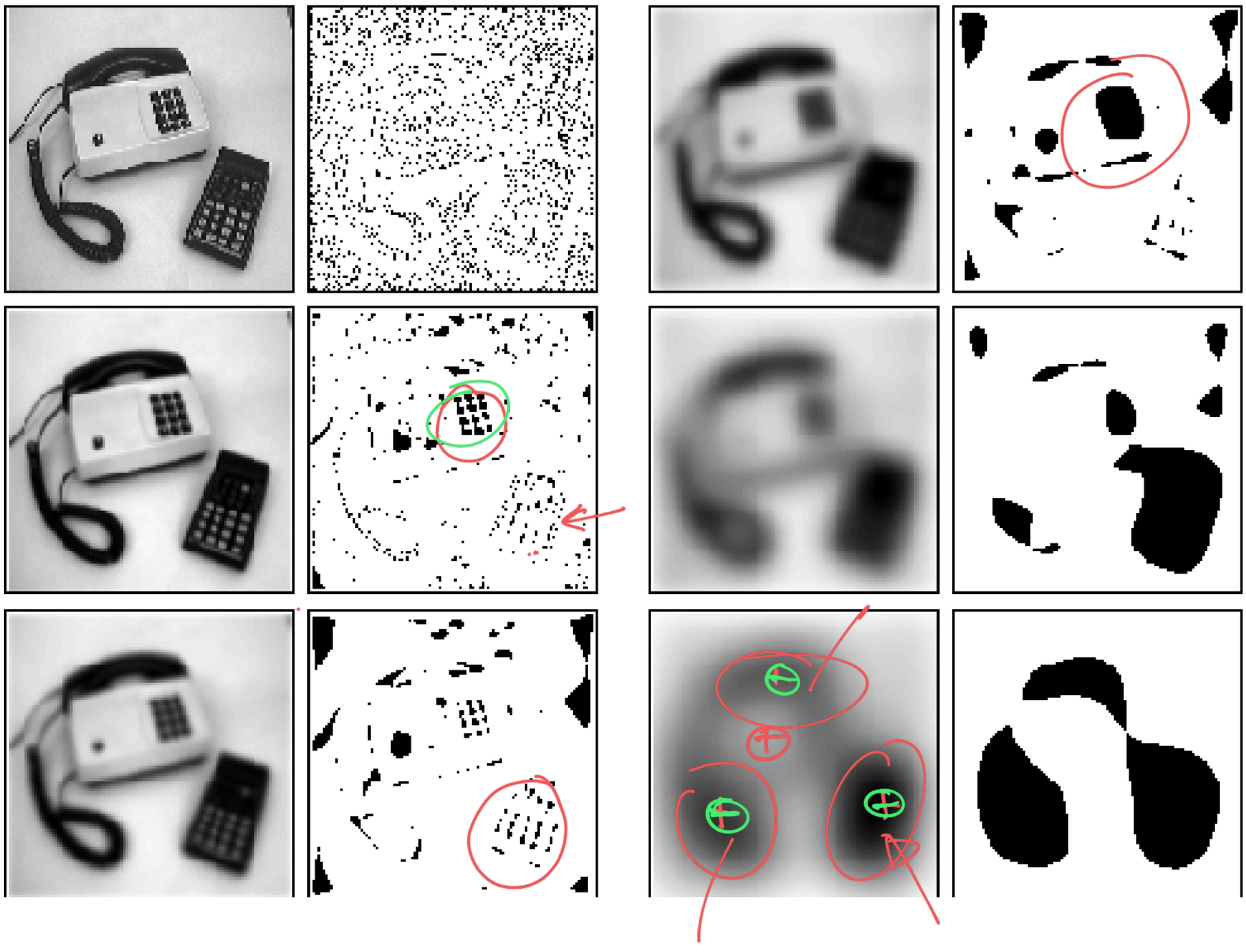
Scale Space Maxima





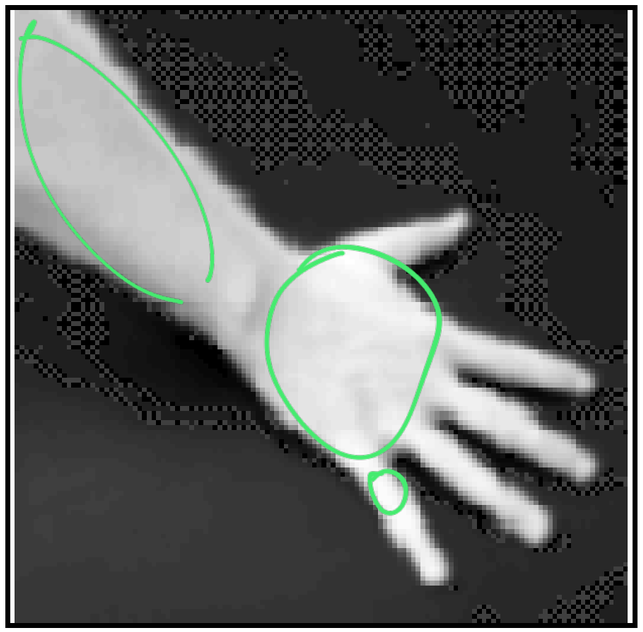
Maxima of the `scale-normalized Gaussian derivatives', at different scales (for a 1D example).

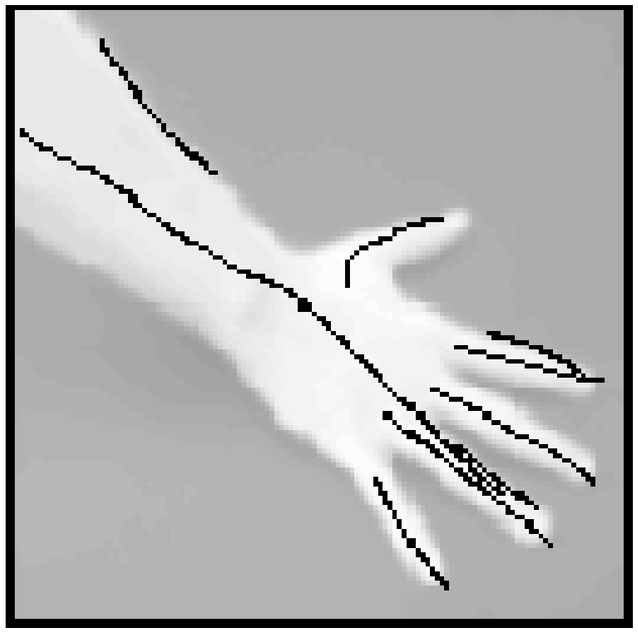
Scale space local minima (filled)

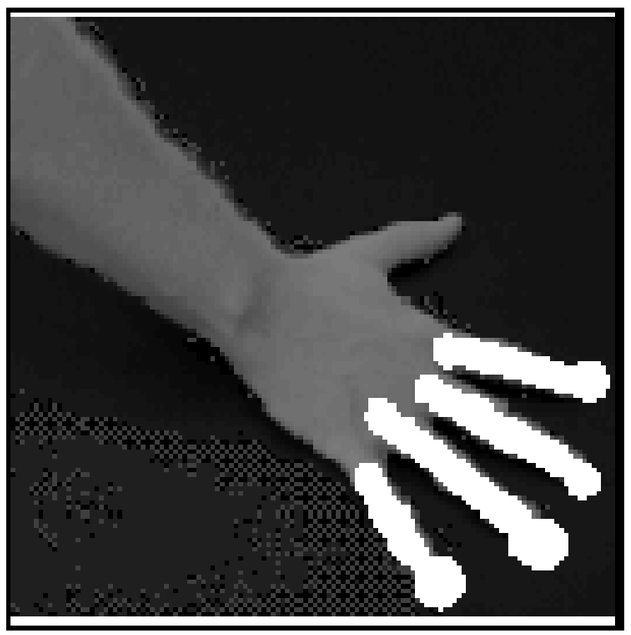


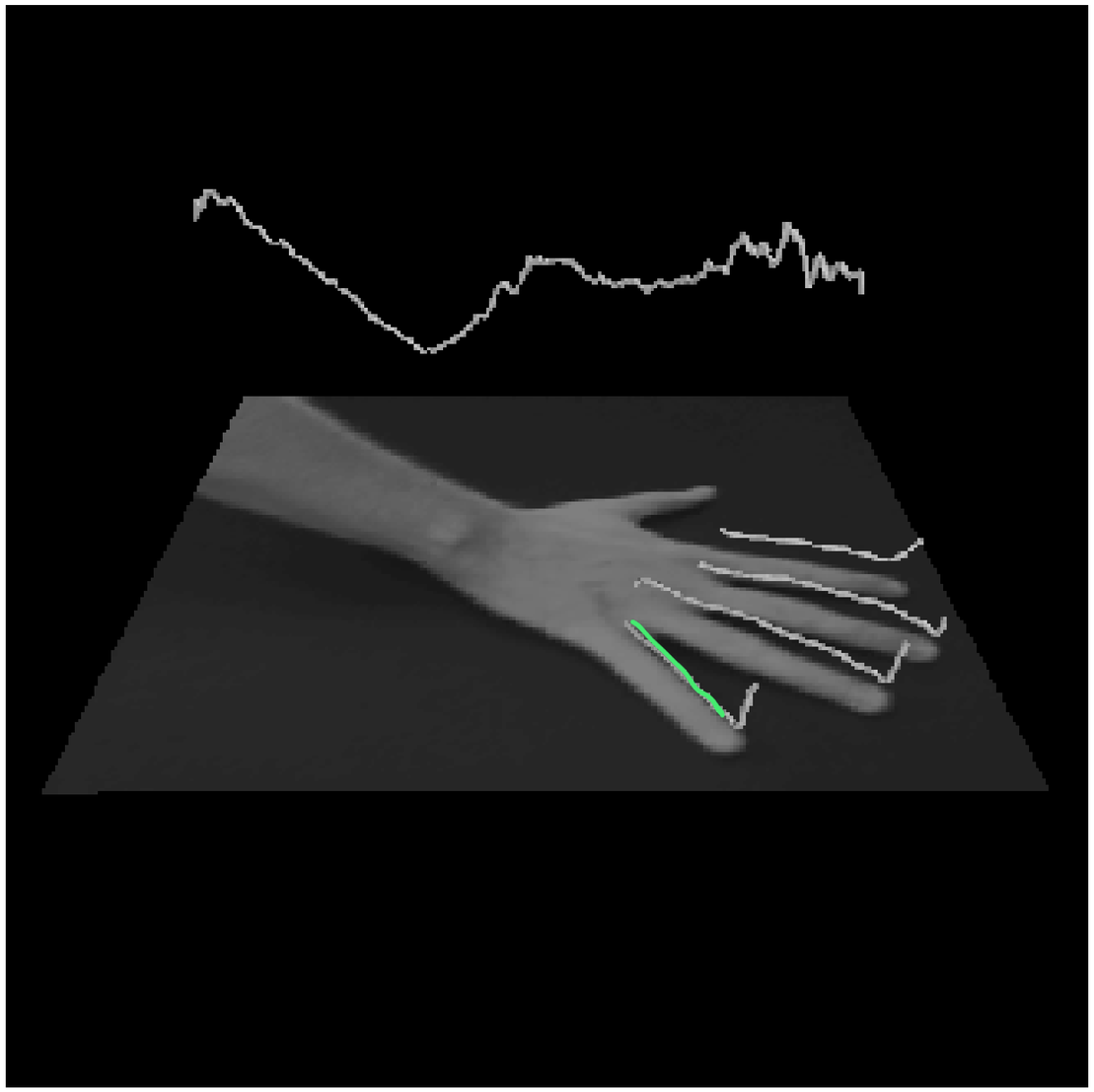
To emphasize the local variations in the grey-level landscape, local minima in the grey-level images at each scale have been indicated by dark blobs (grey-level blobs with spatial extent determined from a certain watershed analogy, which essentially describes how large a region associated with a local minimum can be filled with water, without water flooding over to regions associated with other local minima).

Ridge detection





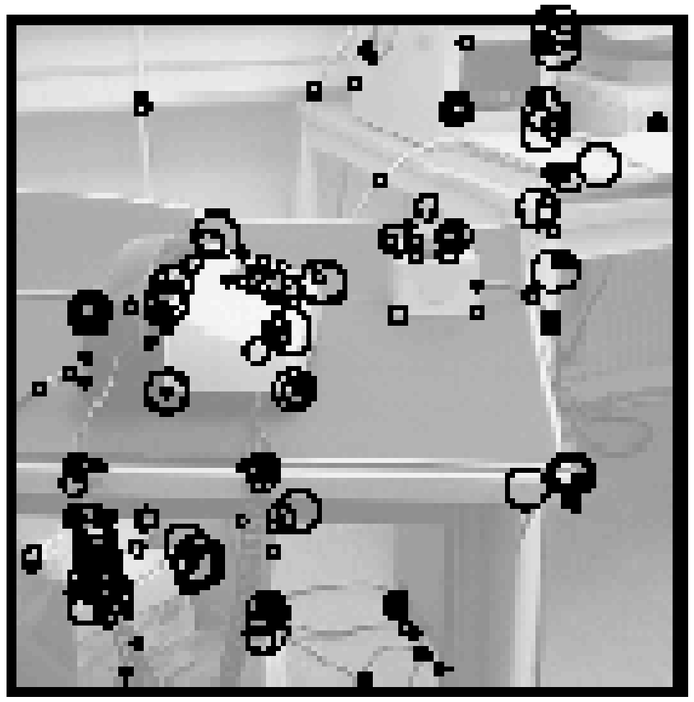


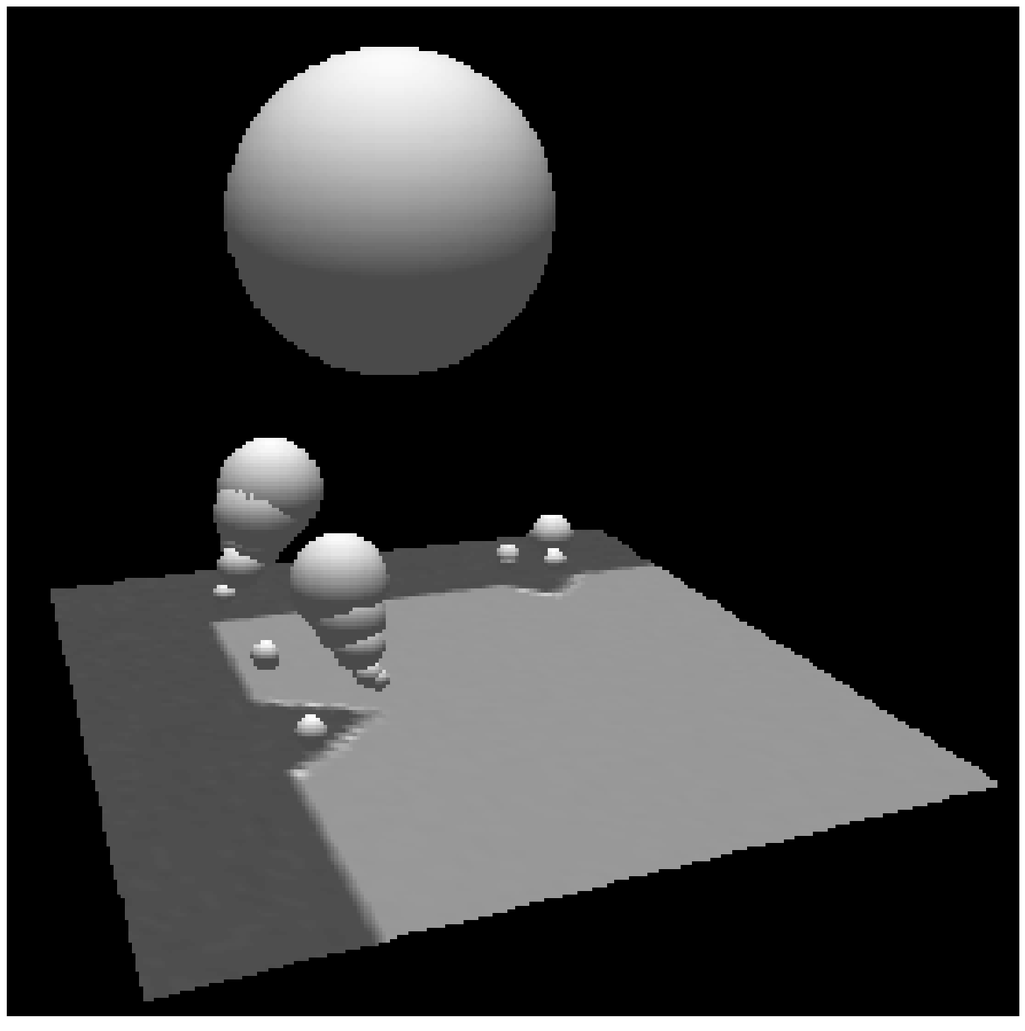


Scale space view

http://www.nada.kth.se/~tony/cern-review/cern-html

Corner detection

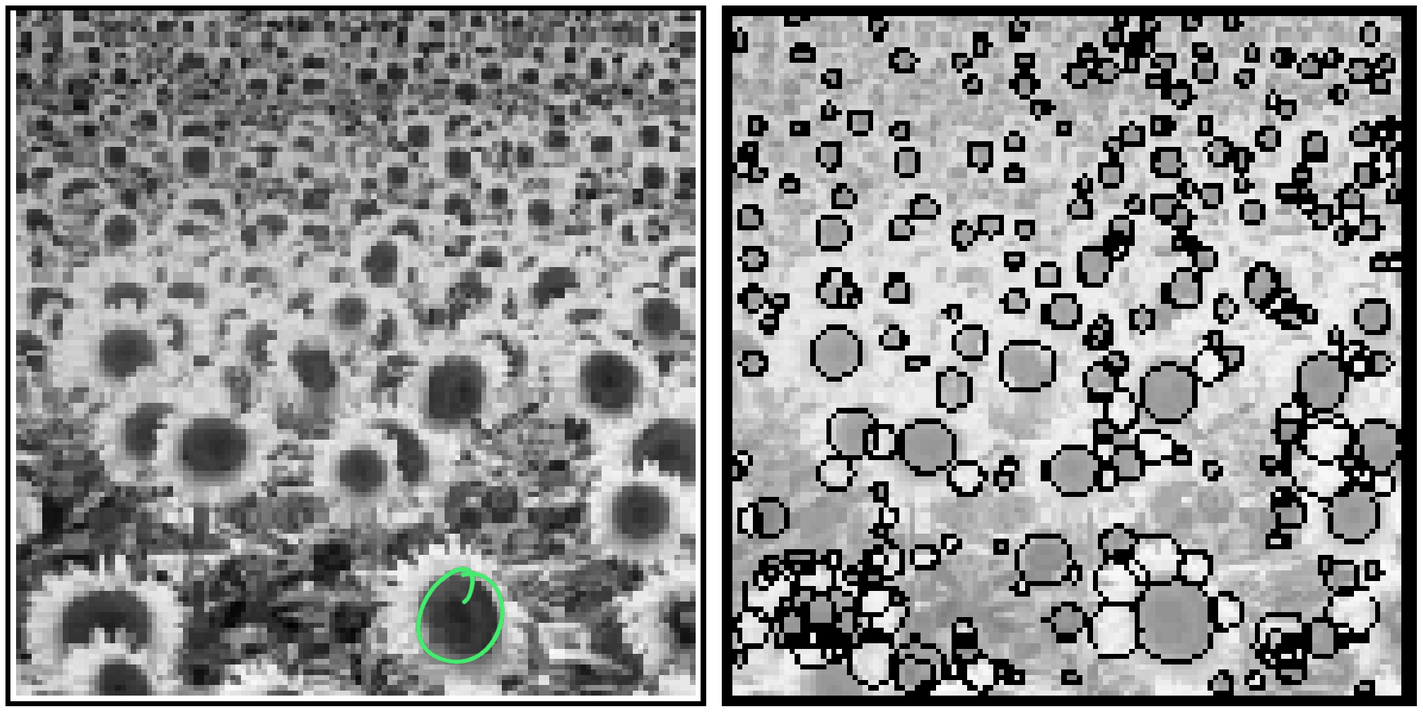




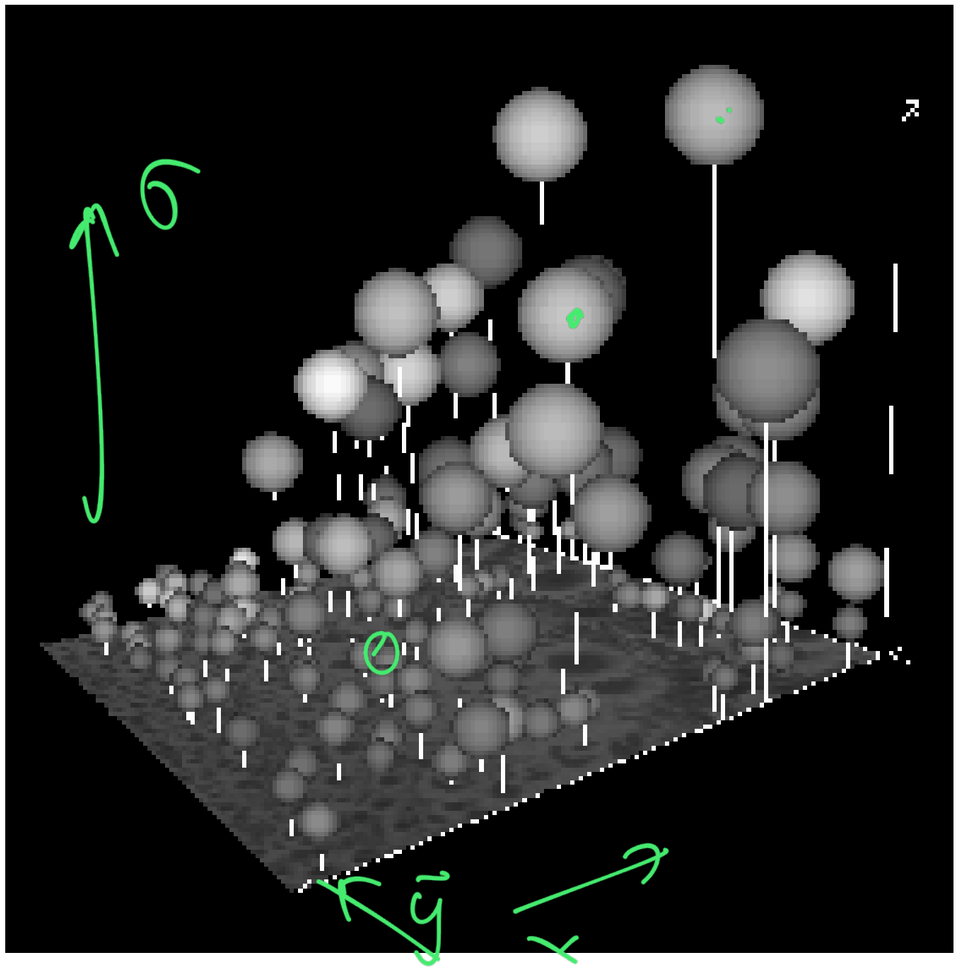
http://www.nada.kth.se/~tony/cern-review/cern-html

Blob Detection

Blobs at different scales



Scale space view



http://www.nada.kth.se/~tony/cern-review/cern-html

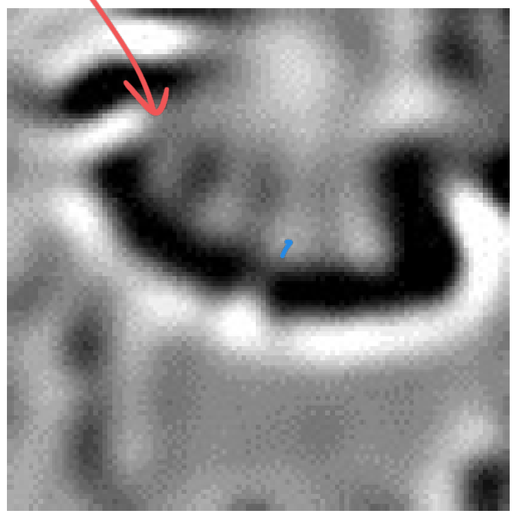
LoG: Laplacian of Gaussians

$(\nabla^2 G_\sigma) * \text{Image}$

$$\nabla^2 G_\sigma = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) G_\sigma$$

Image

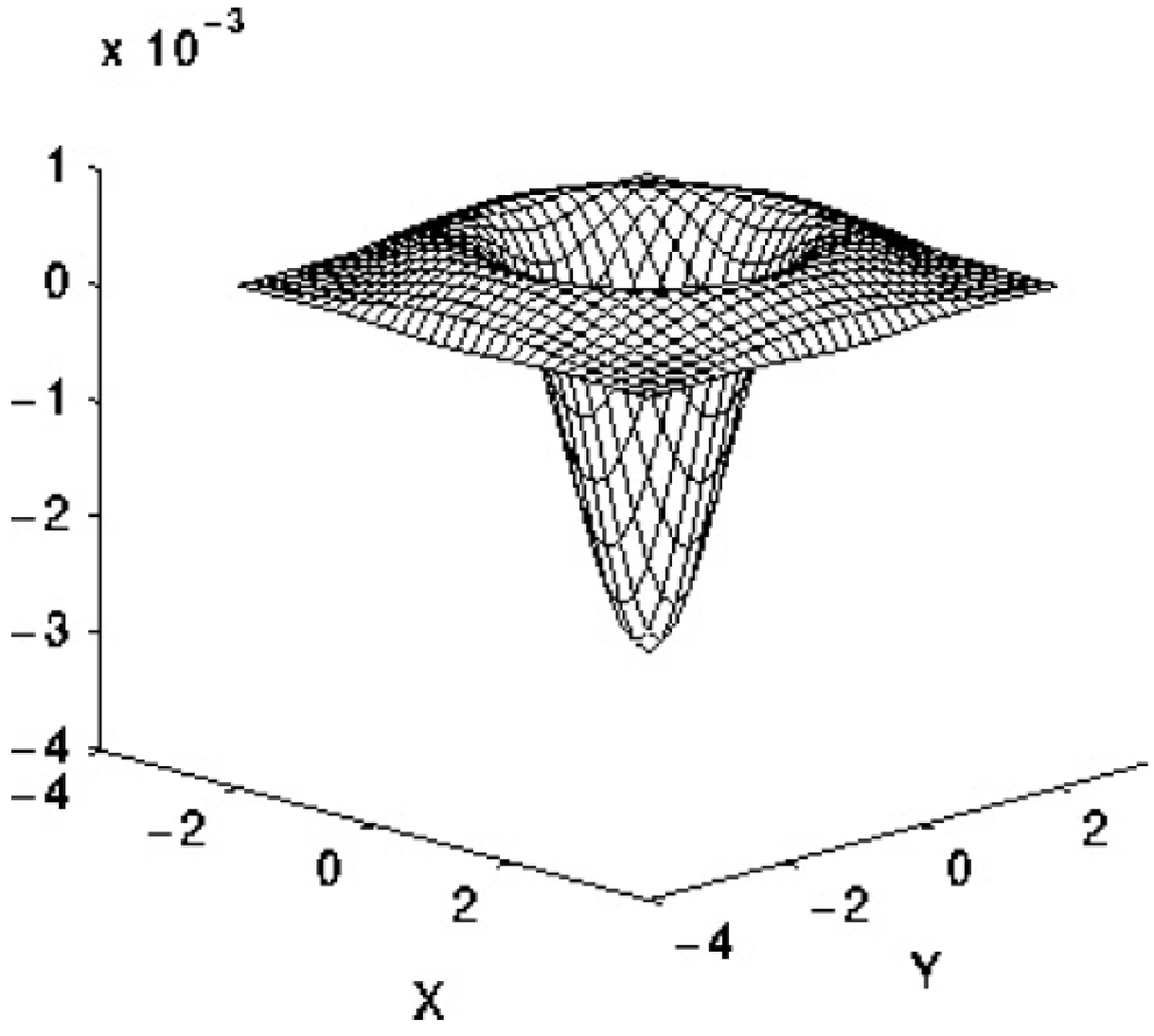
(a)



(b)

(c)

(d)

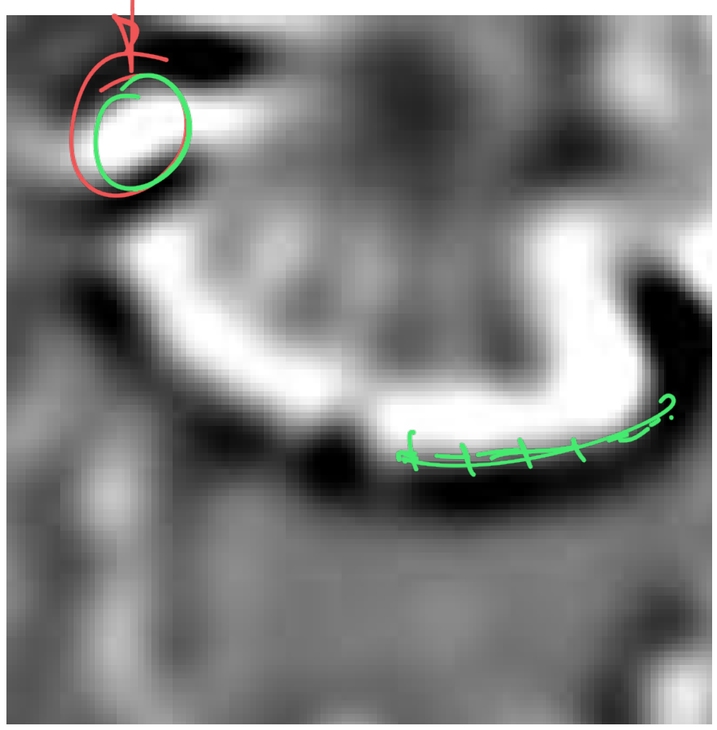


(e)

Laplacian of Gaussians

Scale 1

Scale 5



The trace of the Hessian measures the ‘total second derivative’.
It is a good blob detector (but picks up edges, too).
When multiplied by the square of the scale, this *`scale-normalized Laplacian’* peaks at the scale of the blob.

extrema of $\sigma^2 \nabla^2 G * I$

Trick: Computing LoG as a DoG

The relationship between D and $\sigma^2\nabla^2 G * I$ can be understood from the heat diffusion equation (parameterized in terms of σ rather than the more usual $t = \sigma^2$):

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G.$$

he might have taken his time here…

From this, we see that $\nabla^2 G$ can be computed from the finite difference approximation to $\partial G / \partial \sigma$, using the difference of nearby scales at $k\sigma$ and σ:

$$\sigma \nabla^2 G * I = \frac{\partial G}{\partial \sigma} * I \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} * I$$

and therefore,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$$

1. Lindeberg / Koenderink
LoG extrem $\sigma^2 \nabla^2 G * I$
2. diff val related to $\frac{\partial G}{\partial \sigma} * I$
3. truc: verschil DoG

This gives a `scale-normalized Laplacian', precisely what is needed.

Implementation

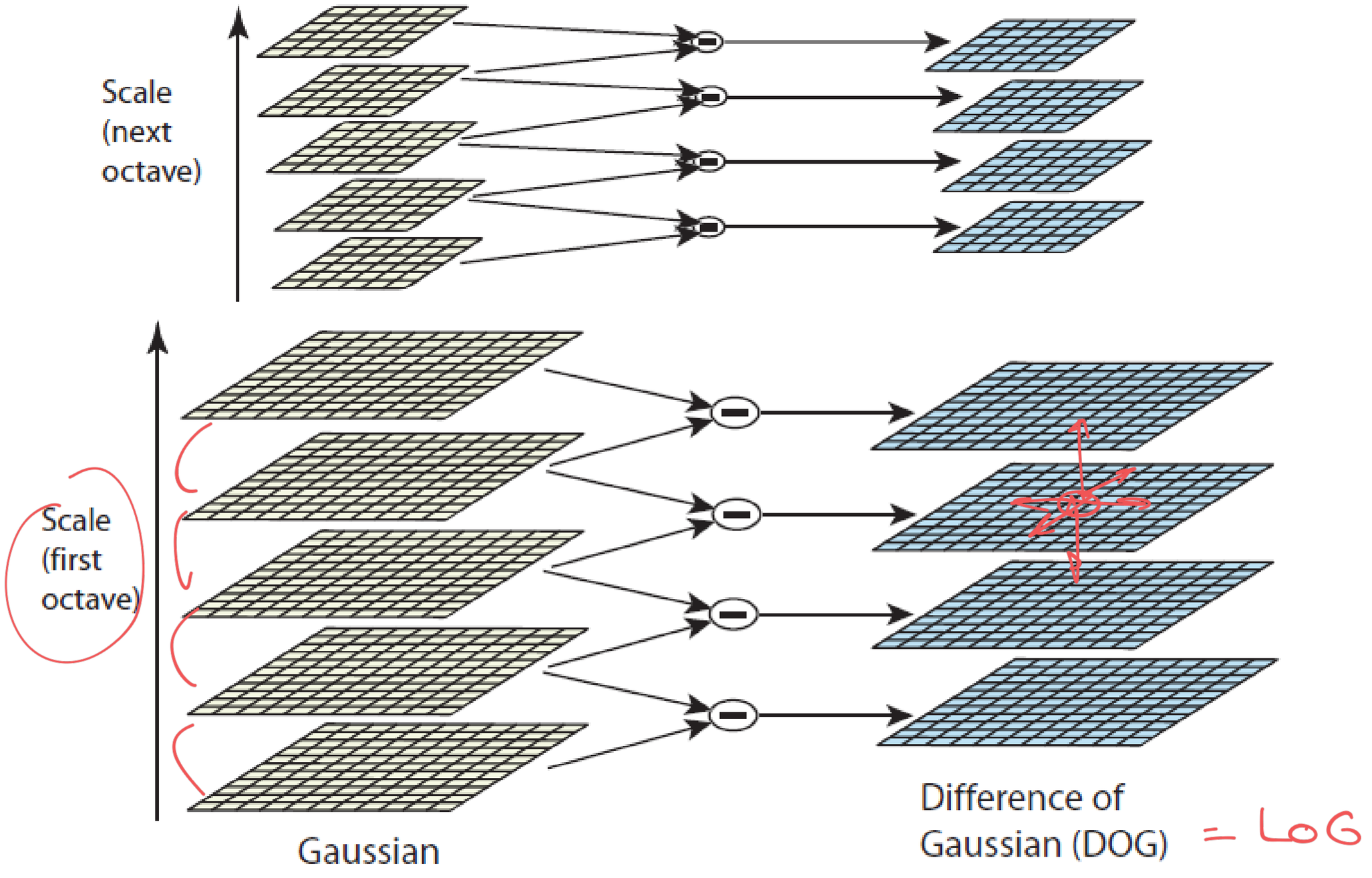


Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

From how_to_read_lowe.pdf

3. Detection of scale-space extrema

And we're off! Scale space is a bit new to you at this point, look it up or come to the lecture. But you know Gaussian convolution!

Your first pass of reading should tell you that the order of this section is messy. He want to use his own method (extrema of difference of Gaussians) because it is efficient. Oh, and actually it is a good approximation the Laplacian (look up what that is!). That is of course the really fundamental relationship to local structure. After having given (1), he derives it (he could/should have done that first), but does not quite return to 'his' D. A lab question asks you to close the gap.

What is an octave? Why does he 'need $s + 3$ images in the stack of blurred images for each octave'?

Pass 1, Section 3.1, 3.2, 3.3

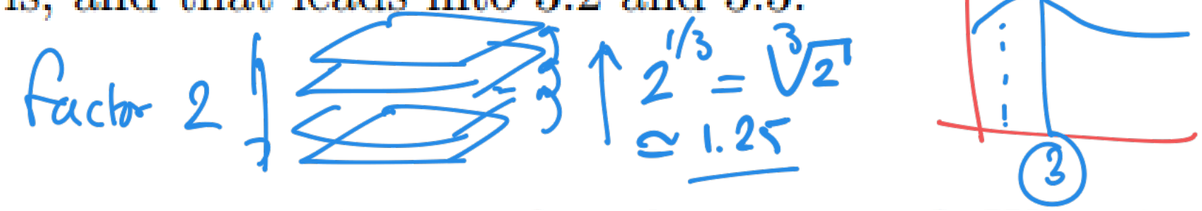
- It’s about implementational choices
- Read these in one go in 10 minutes
- Let’s discuss the bottom line

vaaf
9:25

From how_to_read_lowe.pdf

3.1 Detection of scale-space extrema

It is a pity that he does not quite give the intuition of why you want these extreme in 'all' directions. This must be standard scale space theory. Look it up. What does this mean for the kind of detection he does in the local structure, can he detect blobs, ridges? Remember, D is essentially the Laplacian.
But he is taking a very simple local way of determining an extremum. Which? Because this is so simple, he is concerned about how good it is, and that leads into 3.2 and 3.3.



3.2 Frequency of sampling in scale

We are diving into details here - he is doing some experiments, for what purpose? What is the outcome?

3.3 Frequency of sampling in the spatial domain

What is going on here? He is going to smooth before he detects extrema. But he has already smoothed to make the scale space images. Why is yet another smoothing needed? Could he not have combined that with the earlier smoothing?

(Leo does not understand this section.)

From the lab exercise (Pass 2)

3.4. **(3 points)** Scale space extrema are going to be detected by computing D according to (1). According to Lowe "the relationship between D and $\sigma^2\nabla^2 G$ can be understood ..." but he does not return to D. Close the loop in his explanation yourself.

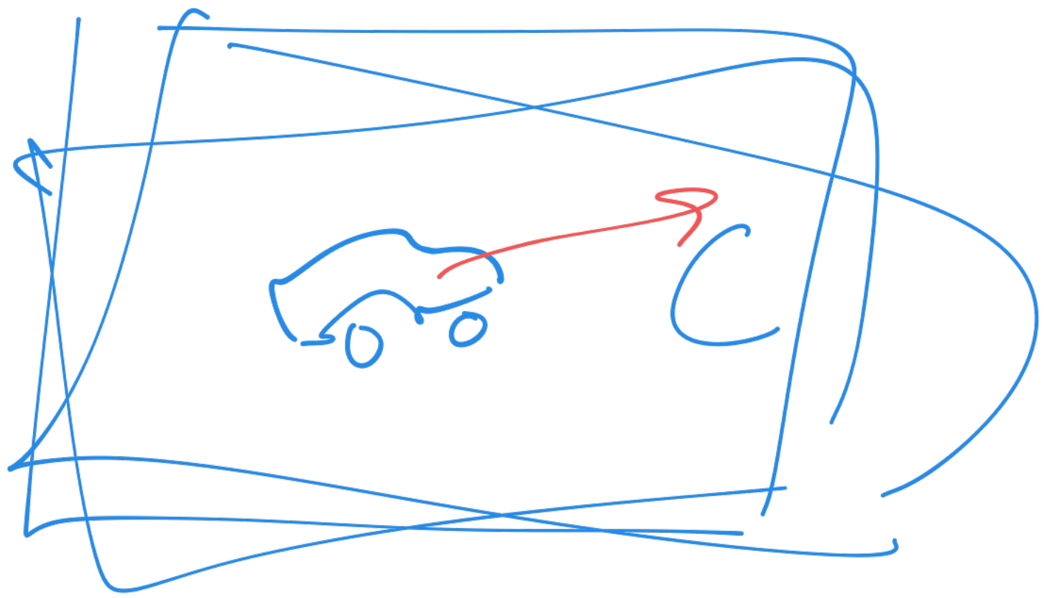
3.5. **(3 points)** If you did not look it up before, ∇^2 is the Laplacian, in 2D images it can be computed as the trace of the Hessian, i.e., $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Prove that the trace of the Hessian is equal to the sum of the eigenvalues (see page 12, where he uses this fact). Hint: use the eigenvalue decomposition on the Hessian. Is it always possible to diagonalize the Hessian?

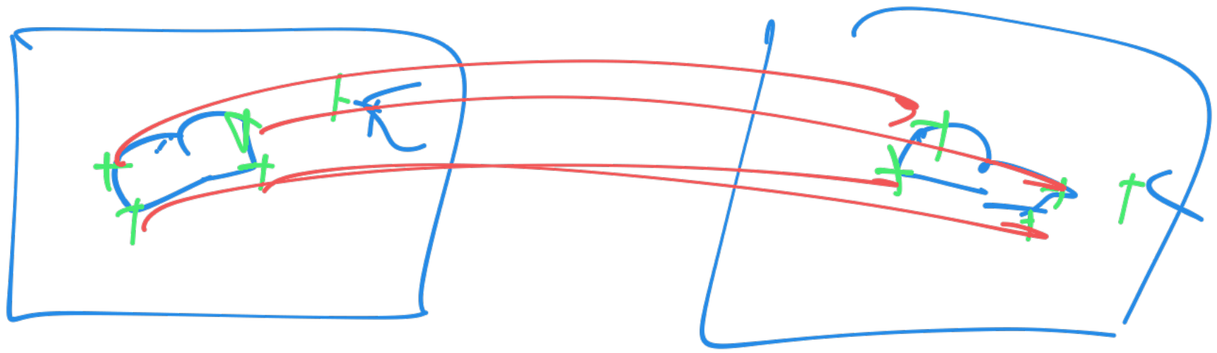
$$Hf = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}$$

$$tr(ABC) = tr(BCA)$$

$$\text{trace}(Hf) = \text{sum diagels}$$
$$= f_{xx} + f_{yy}$$
$$= (\partial_{xx} * G_\sigma) * f + (\partial_{yy} * G_\sigma) * f$$
$$= ((\partial_{xx} + \partial_{yy}) * G_\sigma) * f$$

LoG





extrema

$(\sigma^2 \nabla^2 G_\sigma * I)$

LoG

D * I

DoG

extrema

SIFT $D = G_{k\sigma} * I - G_\sigma * I$

$= (G_{k\sigma} - G_\sigma) * I$

$k = \sqrt[3]{2}$

section 3

(x) (1-a)

Pass 1: Section 4 Keypoint Localization

- Nice section that we can fully understand, even though it was apparently new for Lowe
- Read this in 5 minutes 9:10
- It will be an exercise to understand this at Pass 2 level – do you think you can do it?

4.0

4. Accurate keypoint localization

gradient van D

$$\mathbf{x} = (x, y, \sigma)^T$$

3D

$(\nabla D)^T \vec{x}$ 3 ords

Hessiaan ← 6 ords

$$D(\mathbf{x}) \approx D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \qquad (2)$$

$x^T H x$

vector

vector differentiation.pdf

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}. \quad = H^{-1} \nabla D \qquad (3)$$

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D}{\partial \vec{x}} \hat{x}$$

facet model

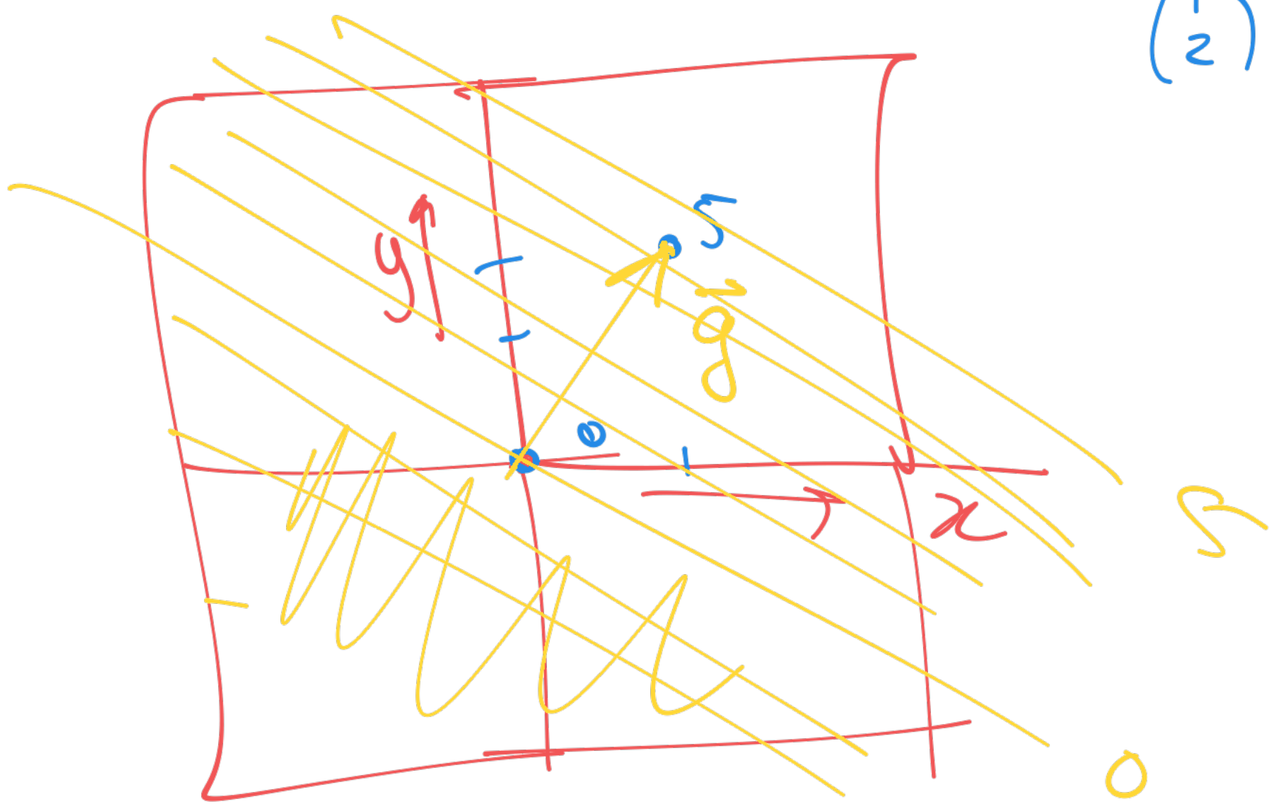
3x3x3 27

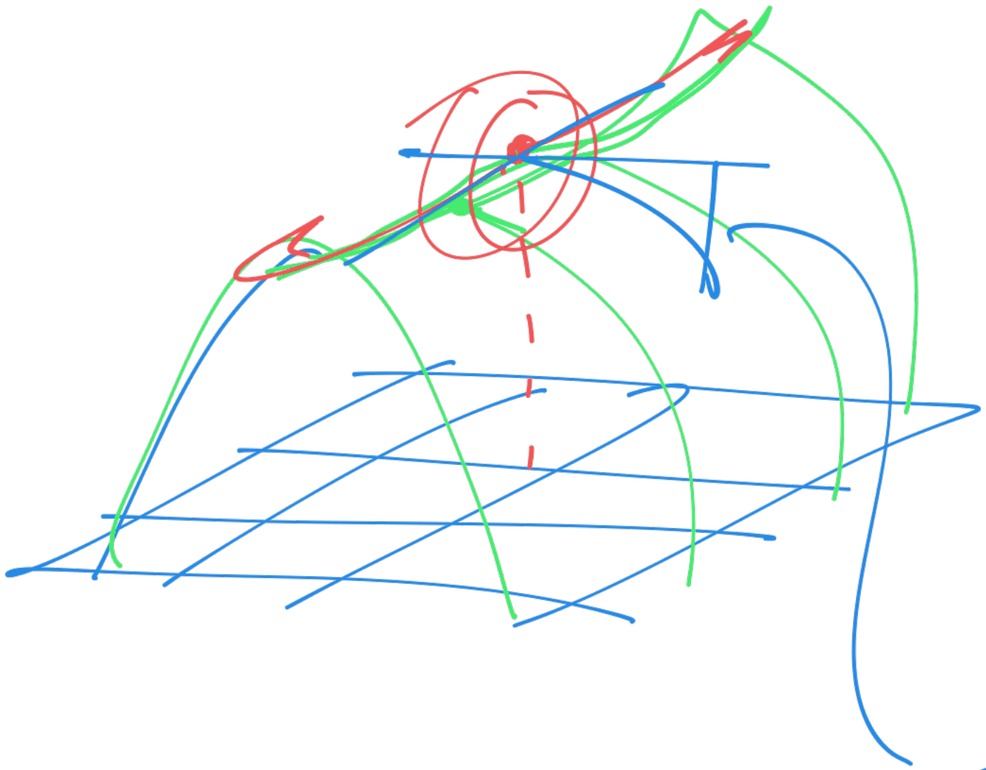
$$\frac{\partial}{\partial \vec{x}}(\vec{x} \cdot \vec{q}) = \vec{q}$$

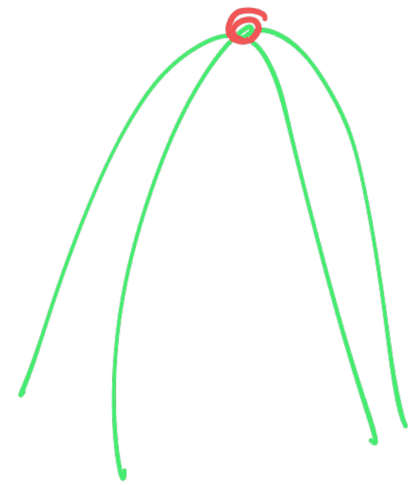
$$\vec{x} \cdot \vec{g}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} = x + 2y$$

$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$







P,q uit de diagonalisatie van Hessian

From how_to_read_lowe.pdf

4. Accurate Keypoint Localization

Finally, some math! Nothing could be clearer and unambiguous. Note how he used to do it naively but begins to appreciate that he is really in the realm of local structure, allowing sub-pixel accuracy.
Equation (2) is familiar, equation (3) you should be able to derive (in some years, it is even part of an exam!), and the consequence (4) you should derive for yourself.
He needs to do all this on the Laplacian or D image. And then, on page 11, he is doing the Hessian and derivative of D by local differences. Is that consistent? Do you realize that these are *third* and *fourth* derivatives of scale space image data?!

4. Accurate Keypoint Localization

From the lab exercise (Pass 2)

3.7. Section 4: fitting a local quadratic function (I think he means a 2nd order polynomial): consult the Facet Model in Chapter 2.7 of the Lecture Notes (or the 2011 eerste deeltentamen). It is a straightforward application of our linear algebra tools, and you see that this is not explained in detail: we are supposed to know.

3.8. **(3 points)** Make sure that you understand the equation on page 11: compared to (2) on page 10, the 1/2 looks like a typo. Is it?

SIFT keypoint selection

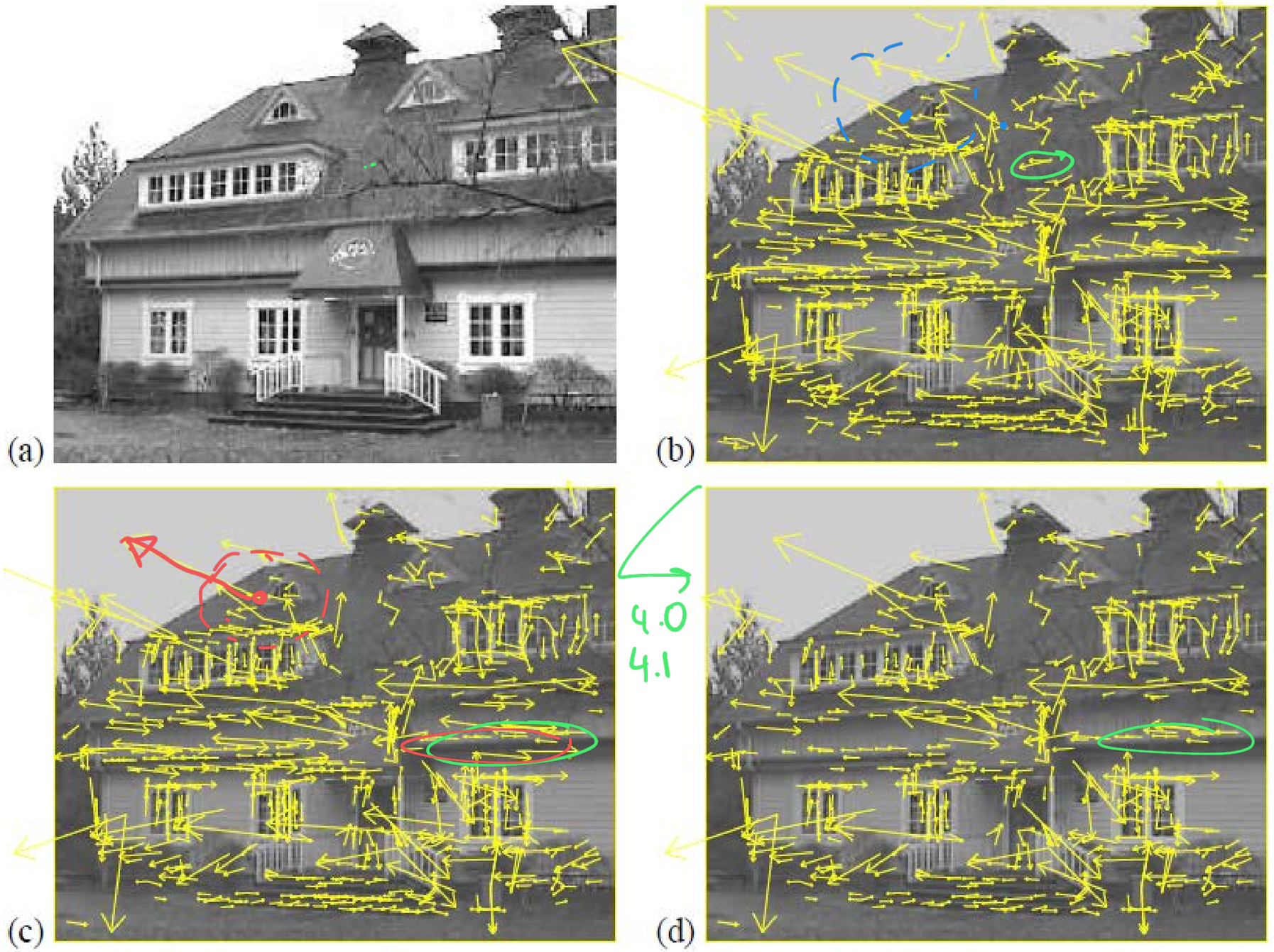


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

Pass 1: 4.1 Eliminating Edge Responses

- Hey, the Hessian – we know this stuff!
- Read in 5 minutes, then we'll talk

9:25

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$\vec{x}^T H \vec{x} \xrightarrow{H\ \text{sym}} \alpha p^2 + \beta q^2$

$\begin{bmatrix} p \\ q \end{bmatrix}^T \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix}$ diagonaliseren.

- Lindenberg tip: take reciprocal measure of

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}.$$

$\alpha + \beta = \mathrm{tr}(H)$

$\alpha\beta = \det(H)$

$\alpha = r\beta$

$H = U C U^T$

$\det(H) = \det(U)\det(C)\det(U^T)$

From how_to_read_lowe.pdf

4. Accurate Keypoint Localization

He is going to do more detailed analysis of spatial peaks of D at a given scale. This is actually a slightly different way of looking at the curvature gauge: we have treated the eigenvectors and eigenvalues; he uses trace and determinant. Why?
Why is it 'unlikely' that the determinant is negative (I would not know), but why does that not bother him?
Count the number of floating point operations - do you get about 20? If not, what does he also include in his calculation?

Pass 1: 4.1 Eliminating Edge Responses

- Hey, the Hessian – we know this stuff!
- Read in 5 minutes, then we'll talk

Hessian Magic

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{4}$$

The eigenvalues of $\mathbf{H}$ are proportional to the principal curvatures of D. Borrowing from the approach used by Harris and Stephens (1988), we can avoid explicitly computing the eigenvalues, as we are only concerned with their ratio. Let α be the eigenvalue with the largest magnitude and β be the smaller one. Then, we can compute the sum of the eigenvalues from the trace of $\mathbf{H}$ and their product from the determinant:

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

to check that the ratio of principal curvatures is below some threshold, r.

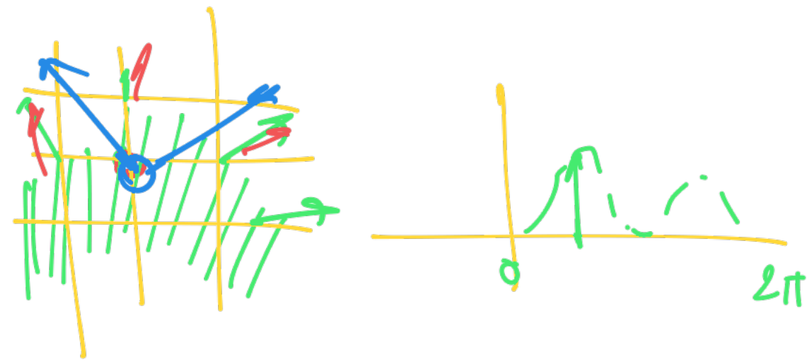
$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r},$$

(Actually, Lindenberg recommends using the reciprocal of this.)

5. Orientation assignment

- Read in 5 minutes
- Then we'll talk

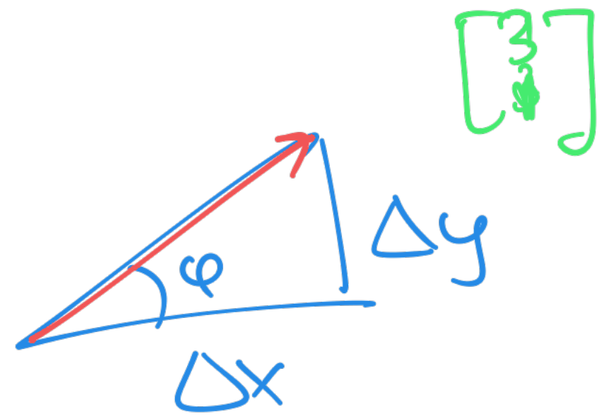
10:10



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

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

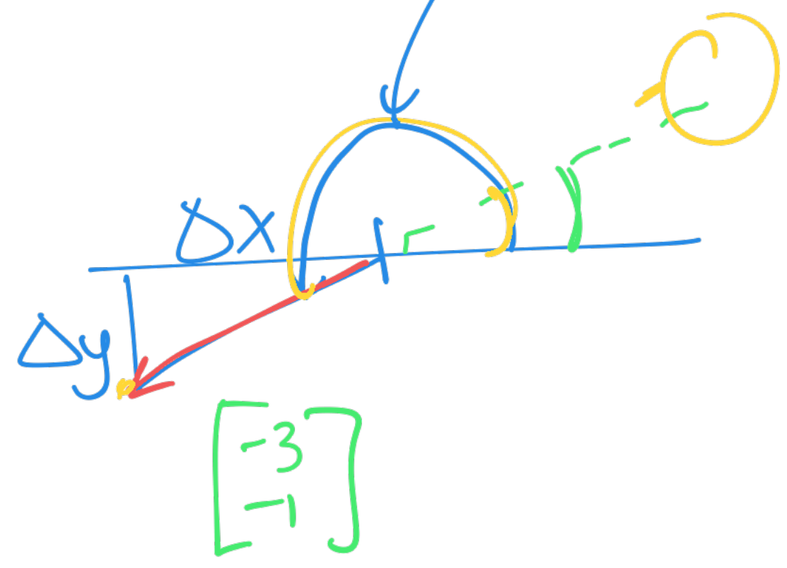
- Why such a simple point-difference gradient filter?
- Never use atan, use atan2() instead!



$$\tan\varphi = \frac{\Delta y}{\Delta x}$$

$$\operatorname{atan}\left(\frac{\Delta y}{\Delta x}\right) = \varphi$$

atan2



atan2(Δy, Δx)

From how_to_read_lowe.pdf

4. Accurate Keypoint Localization

Finally, some math! Nothing could be clearer and unambiguous. Note how he used to do it naively but begins to appreciate that he is really in the realm of local structure, allowing sub-pixel accuracy.
Equation (2) is familiar, equation (3) you should be able to derive (in some years, it is even part of an exam!), and the consequence (4) you should derive for yourself.
He needs to do all this on the Laplacian or D image. And then, on page 11, he is doing the Hessian and derivative of D by local differences. Is that consistent? Do you realize that these are *third* and *fourth* derivatives of scale space image data?!

4. Accurate Keypoint Localization

He is going to do more detailed analysis of spatial peaks of D at a given scale. This is actually a slightly different way of looking at the curvature gauge: we have treated the eigenvectors and eigenvalues; he uses trace and determinant. Why?
Why is it 'unlikely' that the determinant is negative (I would not know), but why does that not bother him?
Count the number of floating point operations - do you get about 20? If not, what does he also include in his calculation?

From how_to_read_lowe.pdf

5. Orientation assignment

The features are not going to orientation invariant, but the method will be. Resolve that paradox in your mind.
In the formula, he does not use the `atan2` function. I think he should (and probably does). Why?
He computes the gradients - where in his total set of computations is that done do you think - everywhere or only at the peaks? And is this still the gradient of D (as above) or of something else?
'Finally a parabola is fit', make sure you understand to what. This is in fact a 1D version of the localization of extrema we saw before.
Again he puts in some experiments for this step. We can feel him developing and testing his modules step by step. Educational it is.

6: The Local Image Descriptor

- This is the core of the paper, the true innovation
- Focus on 6.1 and Figure 7
- Read carefully, take ~~20-30~~ minutes

10 from 10:25

SIFT keypoint selection

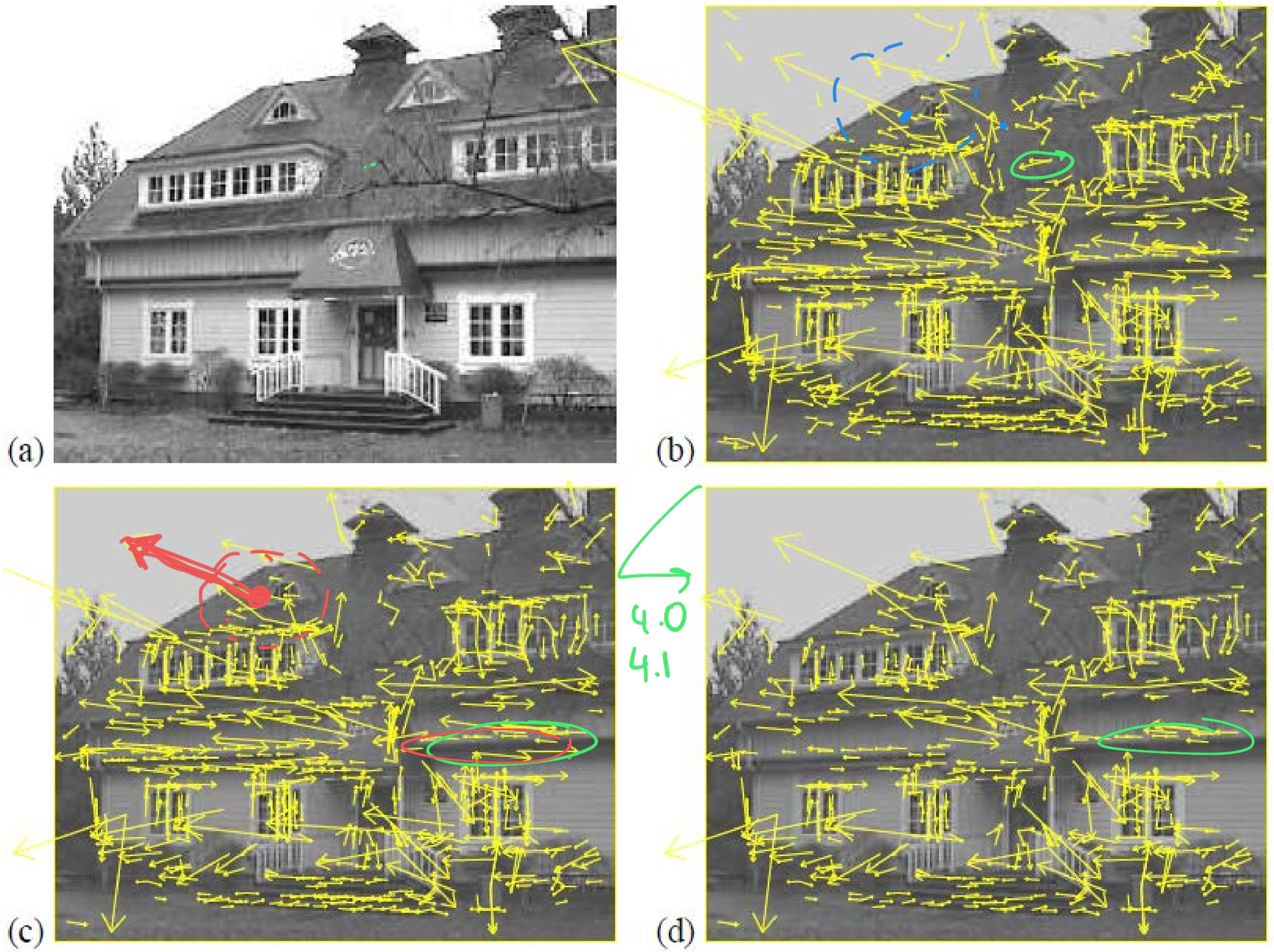


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

From how_to_read_lowe.pdf

6. The local image descriptor

We get a biological motivation. This seems a strange place for it, you could also imagine it in the introduction since we AI people would have experienced that as an extra reason to be interested in the method. It must have been an afterthought.

6.1 Descriptor representation

Here we get a descriptor, localized by a Gaussian of width σ. What is this σ, is it one of the earlier values given?
What is trilinear interpolation and why are we in 3D? What are the 3 dimensions?
What does he mean by 'affine changes in illumination'? Gather this from the surrounding text!
That bit about 0.2: yuck, how *ad hoc*!

6.2 Descriptor testing

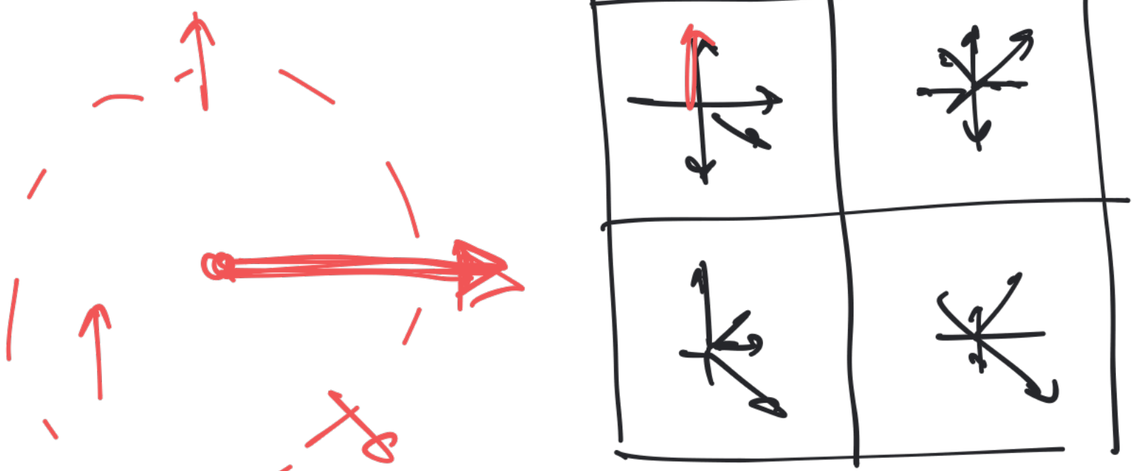
We get the motivation for some numbers he mentioned earlier. This and the next sections are of course important to convince people of the effectiveness of his method, but they do not make for exciting reading.

6.3 Sensitivity to affine change

Yawn. I mean, impressive enough.

6.4 Matching to large databases

For modern applications, these properties of SIFT (and its successors) are very relevant.



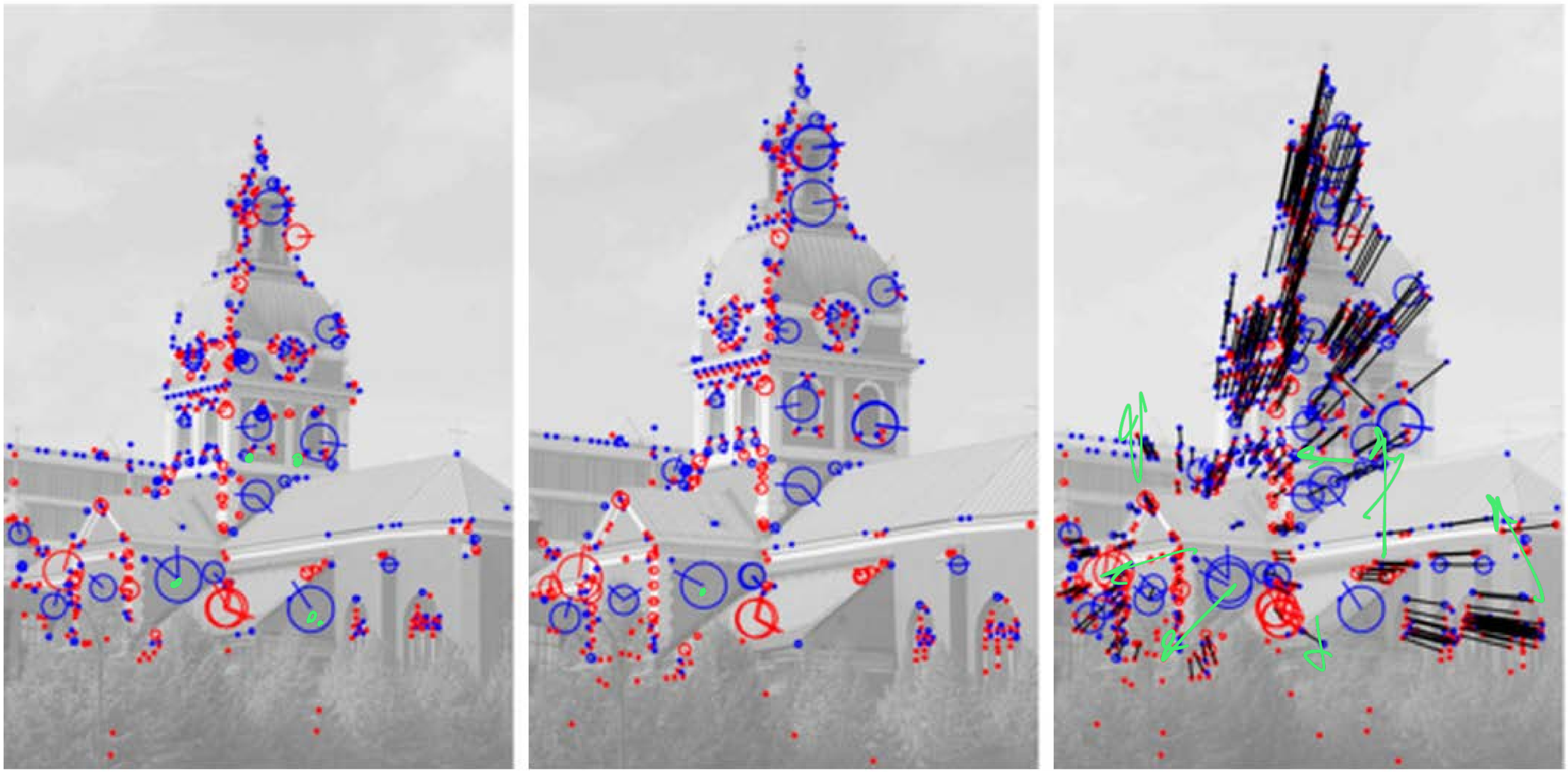
Keypoint descriptor

From the lab exercise (Pass 2)

3.8. **(3 points)** Make sure that you understand the equation on page 11: compared to (2) on page 10, the 1/2 looks like a typo. Is it?

3.9. **(2 points)** Page 15, line -4, mentions trilinear interpolation. Look it up in `wiki`. Why are we in 3D, what are the meanings of the dimensions of the space we are in? (Hint: it is not scale space!)

SIFT features used for scale-invariant matching



7: Application to Object Recognition

- Here we see the new feature method enabling an application.
- Take 10-15 minutes for a first pass of 7, 7.1-4
- We are going to do Mosaicing - how can that use Keypoint Matching?
- 7.3: Note how he uses Hough to vote for a transformation!
- But we are going to use RANSAC rather than Hough for this lab exercise (next lecture).
- 7.4: nothing new for us.

Figure 12

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

From how_to_read_lowe.pdf

7 Application to obect recognition

7.1 Keypoint matching

7.2 Efficient nearest neighbor matching

7.3 Clustering with the Hough transform

The Hough transform is also part of Beeldverwerken. He uses elementary techniques, but this papaer enabled things we could not do before, certainly not as robustly.

7.4 Solution for affine parameters

What is an orthographic projection?
This section has techniques that should look very familiar indeed!

8 Recognition examples

9 Conclusions

You read these in your first pass. Do you agree with his conclusions? Are there some that are missing?

8: Recognition & 9: Conclusions

- 5 minutes suffice