# Some useful hints
# (sorry, no "help" menu)

At any moment you are in one of two modes: the MANUAL (like now) or the DEMO. To get from:

- the DEMO to the MANUAL: Click in the left window half

- the MANUAL to the DEMO: Click in the window

Here are your options:

**MANUAL** Use the LEFT/RIGHT arrow keys to cycle through the pages presently available. Make sure that you always try whether there might be a next page. If there is only one page, nothing happens. If there are more pages then the unsuccessful try gets you to the first page again.

**DEMO** At any time you can hit the A, B or C and 0, 2, … (depends) to navigate to another topic. The texts at right tell you about your options.

Make sure you use the texts at right to your best advantage. They tell you where you currently are, what you can do, what the present parameter values (when they make sense) are, and so forth.

**Producing eidolons**

Although this DEMO is just what the name suggests (a DEMO), you can use it to produce eidolons, although your options are limited.

**Firstly**, at program startup you get the chance to load the image of your choice. But notice:
- the image is stripped of its color (the program selects the green channel)
- the image is mapped on a 512-square format

A better way than depending upon the program is to cut the image down to size yourself. It puts you in a limited position of control.

**Secondly**, you may at any time same the image currently displayed at left. Just hit the ENTER or RETURN key (depends upon your platform). This will bring up a file definition window. By naming the file, say "MyEidolon.png" you also specify the format (here "png"). If you don't specify an extension you get a "tif" file (on most platforms).

This allows you to produce eidolons from 512x512 monochrome images, which is better than nothing.

You will be notified of success or failure by red letters superimposed over the image at left. (Hitting the space bar removes these.)

**Computations & waiting time**

The program computes fresh instances at almost any state change or event. For instance, it will recompute when you hit the keyboard space bar, which is very useful to sample different instances of an eidolon.

It also recomputes when you return from the MANUAL and in some cases has to (at least partly) recompute when you change a parameter value.

Computation takes time. This may vary from a fraction of a second to several seconds. It makes the DEMO feel unresponsive sometimes. It is best to get used to the delays and expect these, then you are unlikely to try all kinds of things because you think the program is stuck. If it is computing there is nothing you can do about it but wait.

Waiting times are reasonably acceptable on my computer (a mid 2009 MacBook Pro), so it may well be faster on yours.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# Manual Pages

The manual pages are designed such as to get more out of this demo. Not only that, you will probably get frustrated if you don't use them!

This first page is only prepended to the present topic (A.0. you're in), you will not encounter it elsewhere.

This demo is part of the distribution of a paper by Jan Koenderink, Andrea van Doorn, Matteo Valsecchi and Karl Gegenfurtner, called EIDOLONS: NOVEL STIMULI FOR VISION RESEARCH, which appeared in 2016 in Journal of Vision XX(XX), pages xxx–xxx. The demo probably will make little sense to you if you haven't read the paper. This must be an important truth since its opposite is true too.

We hoped you would end up here after starting the application. We'll suppose that you cancelled the file selection and slavishly clicked both CLICKME's. This is the point where we explain the interface.

A moment ago you were in the actual DEMO, right now you're in the MANUAL pages. It is easy enough to toggle between the two: click in the window. When in the demo you should click in the image at left – best to click on the leftside throughout.

**What if nothing happens?** First of all PATIENCE! After a click the demo generates a fresh eidolon, this may take several seconds on your computer. If nothing happens after quite some time, then perhaps your click didn't register. Try again, preferable on the left side.

This manual can be paged using the LEFT/RIGHT arrow keys. If nothing seems to happen, then that is because the particular topic has only a single page. The arrows cycle you periodically through the pages.

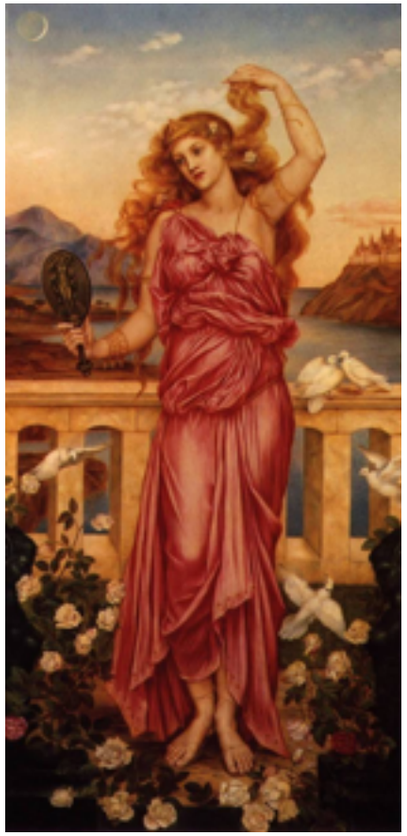
You get some basic help by hitting the "H" key in demo mode.

**Structure of the demo** The demo is partitioned into three main topics A, B and C, whereas each topic is subsectioned into numbered (1, 2, …) parts. Right now you are in A.0.

When in the demo you may hit the A, B or C keys at any time to change the major partition. You may also hit the 0, 1, 2, … keys at any time to get to another subsection. (Notice that A.1001 or Z.0. fail to exist, so nothing will happen if you try to get there.)

In all cases, an image will be on the left, here is your main interest! However, don't skip the right, because it holds important information for you. It tells you where you are, what can be done and lists the current parameter values.

**Why EIDOLONS?** You may remember the poem of that title by Walt Whitman (Leaves of grass). But eidolons were already common in the ancient world. Helen of Sparta is usually called Helen of Troy ("the face that launched a thousand ships"), but according to Euripides' play she never was at Troy: for ten years the Greeks fought over an eidolon (apparition, doppelganger, shapeshift, …)



Helen of Troy,
Evelyn
de Morgan
1898

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# A.0. Fiducial image

This manual page has several uses: it shows the fiducial image and it introduces you the the major eidolon parameters, the "reach" and the "grain".

After starting up the image at left will be an eidolon so close to the fiducial image that it can easily pass for it.

Do can do something about that by holding the UP arrow key for a spell (don't worry, you'll SEE what happens!). You will notice the image deform, a lot even.

Now try the LEFT arrow key. The changes should be obvious enough! Play with both the UP./DOWN and the LEFT/RIGHT arrow keys so as to grow familiar with the phenomena. What you are doing here is to control the reach (UP/DOWN arrow keys) and the grain (LEFT/RIGHT arrow keys.

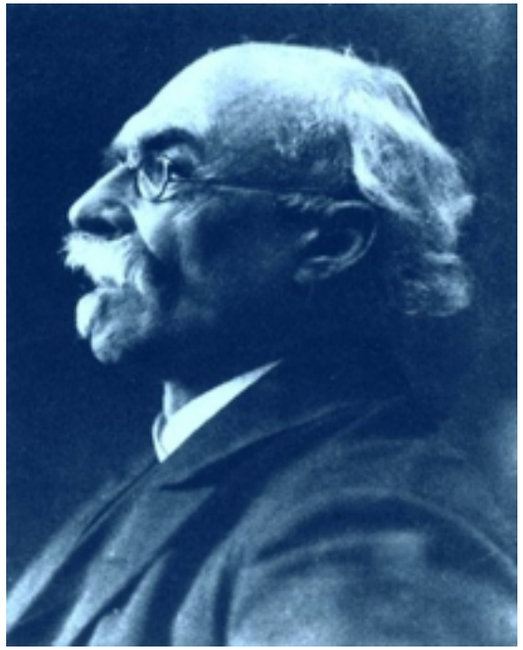
So what do they do? You should know, since you got the experience! But here we go:

The **grain** is a spatial parameter that determines how fast a deformation can change as you move away from any given location. Thus a large grain implies a global deformation whereas a very small grain yields a grainy, diffuse look.

The **reach** is an amplitude-like parameter. A very small reach has hardly any effect, you get an eidolon very close to the original image. A large reach yields a either a major deformation (if the grain is large), or an unrecognizable grainy mess (if the gain is small). Of course, the less extreme cases are the visually most effective.

Both reach and grain parameterize local disarray, which is the major tool we use to produce eidolons. The disarray is based on Hermann Lotze's notion of "local sign" (he said "Localzeichen" in his native German).

Hermann Lotze
1817–1881

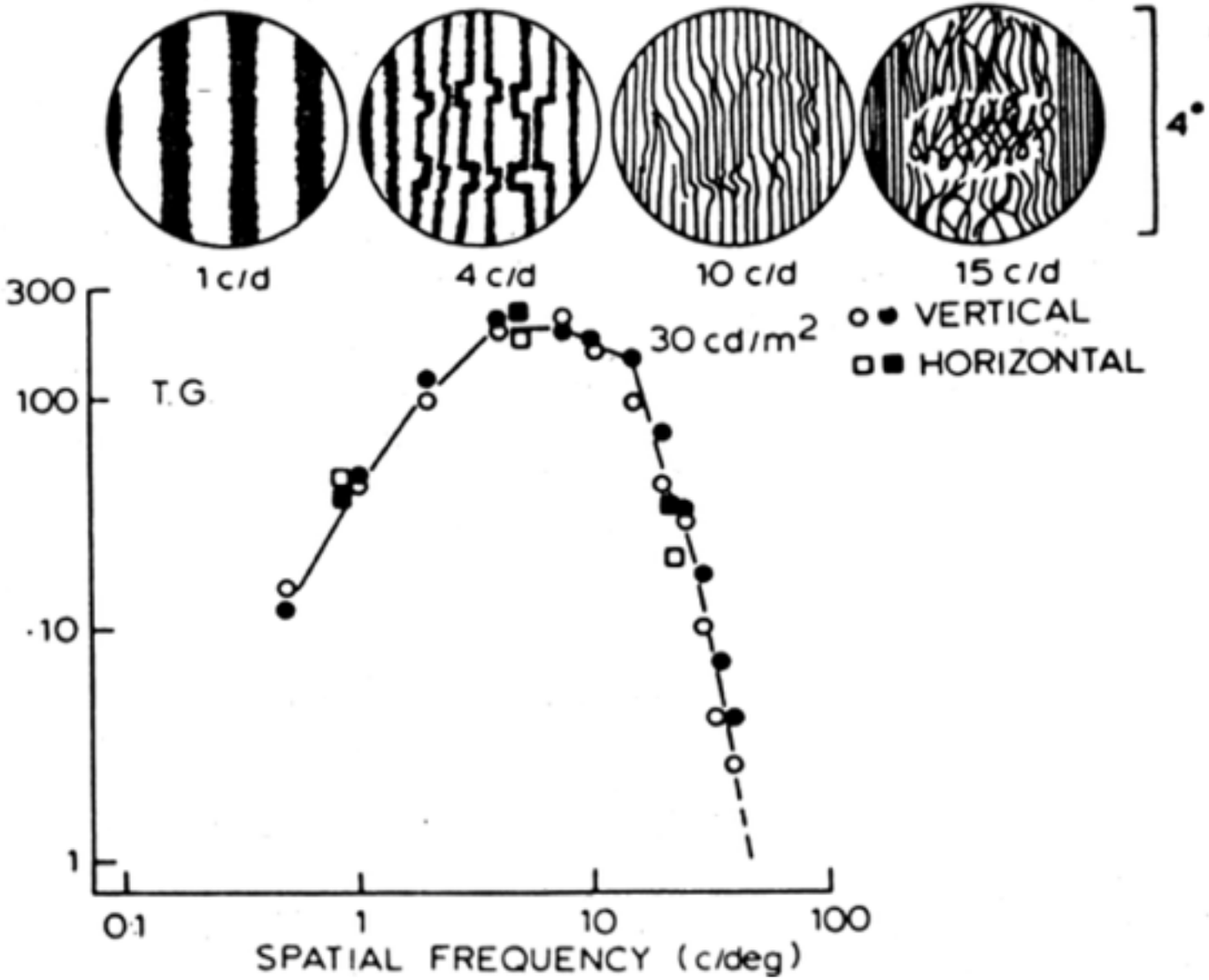


Hermann Lotze was a philosopher, medical researcher, biologist, psychologist and so forth. His local sign idea was given in by the conviction that visual awareness needs some labels in order to get spatial location from neural activity. After all, space is in the mind, not in the brain. People like Helmholtz (physicist, medical man, physiologist, psychologist and so forth) took this very seriously – not such a bad reason for us to do to the same.

There is some evidence that Lotze was right, think of the amblyopia termed "tarachopia" (scrambled vision) by Hess. The tarachopic amblyope has perfect acuity and contrast sensitivity, yet cannot read the headlines of the newspaper. Why? Because scrambled vision has all the pieces, but apparently "cannot do the puzzle".

We use "spatial disarray" to produce eidolons, this is just a "disturbed local sign". In this topic (A.0.) we apply it to the pixels of the fiducial image.

More generally we apply the disarray to "parts" of an image, where the parts could be various things. For instance, we perturb boundary presentations at all scales, and so forth.




This is a figure from Hess' paper. The amblyopic eye has exactly the same MTF for sine-wave gratings as the good eye. This acuity and contrast sensitivity are not impaired. Apparently the optics, an the anatomy/physiology one immediately thinks of are okay.

So what is wrong? Hess had the patient draw records of visual awareness. Apparently the amblyopic visual field appears "scrambled". The patient has a local sign problem! Since no causes can be identified, it has to be classified a "soul blindness" (Seelenblindheit, agnosia).

We use this type of "scrambled vision" in our eidolon factory. We think it likely that such scrambling is also associated with various properties of vision in the peripheral visual field.

But what is scrambled? Most likely, not anything like "pixels". But the presentation in visual awareness are not "pixellated" anyway: think of a uniform blue sky, how many pixels has it? Likely "parts" are suggested by the accounts of visual artists. This points to "parts" such as boundaries (or edges) of various types and of various scales. Such ideas fit very well with accounts of formal geometry (differential geometry in a scale-space setting) and the familiar "representations" in the primary visual cortex.

(Notice that cortical "representations" are categorically different from the "presentations" of visual awareness!)

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# A.1. Lotze disarray

**What is the mechanism of local sign?** Wrong question! There can be no such "mechanism" because local sign related mental entities - location in visual awareness - .to physical or physiological entities - the retinal location say. One needs some kind of psychophysical linking hypothesis here.

Lotze's proposal was that local sign is learned through experience with the effect of eye movements on visual awareness. The eye movements are self produced (a "Willensanstrengung", literally a exercise of the will, a voluntary action), thus the agent can "experiment" with the effects. The mind has access to both the motor intentions and the resulting awareness, so there is no need to refer to any spookiness such as homunculi that look at the brain.

Notice that Lotze's proposal does not mention such things as the structure of the primary visual cortex and so forth. Thus any disarray due to sloppy local sign will apply to all image detail, irrespective its degree of detail. We expect large structures and small structures to be randomly displaced by the same stochastic process.

The result has to be that fine structure is destroyed, whereas coarse structures - with respect to the degree of disarray - will largely survive.

**How is local disarray implemented?** ,The method is simply to displace image structures randomly. The structures may be "pixels", but also more intricate parts. In the eidolon factory implemented in this demo we develop the image into local "edgelets" of all possible orientations and resolutions. Here we took our cue from the understanding derived from the visual arts, especially painting. Here we find the kind of phenomenological understanding of visual awareness that we need. Perhaps not surprisingly, it is not hard to relate this to formal models of the primary visual cortex.

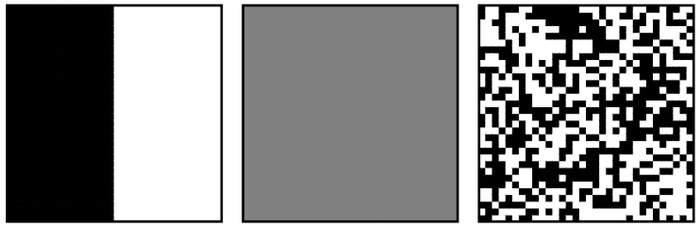
The edgelet development is a development in a linear, overcomplete basis, where the basis vectors are local kernels, looking much like receptive fields. A suitable sum over all edgelets simply yields the image again. (Notice that we don't say the image is "composed" of edgelets, but merely that it can be **decomposed** into them.)

Why is this important if the edgelets simply represent the original image? Well, it yields a handle on where the local sign disarray might do its work. We might displace the edgelets randomly before pooling them to synthesize the image.

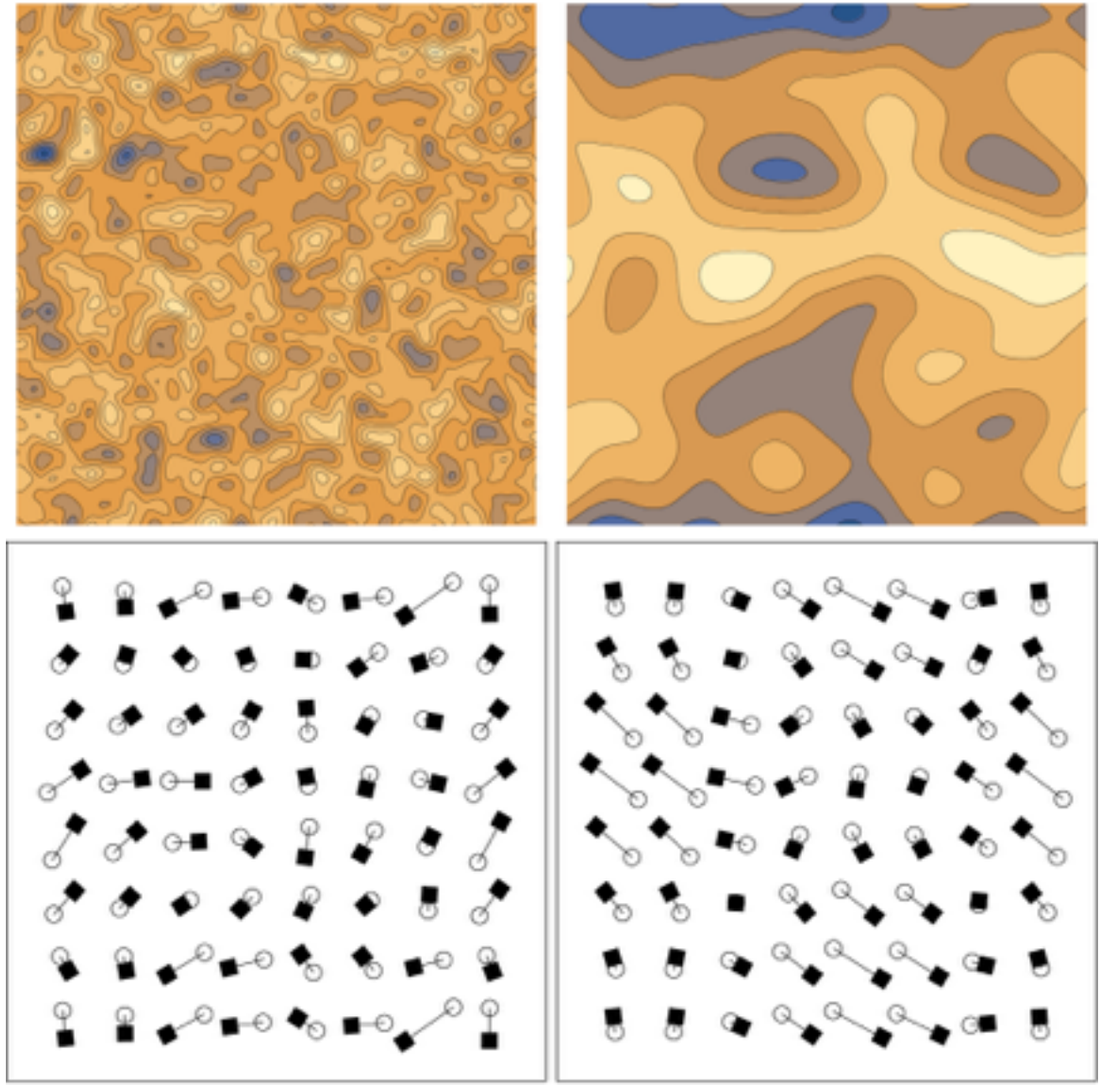
This yields methods of disarray that can be aimed at particular scales or orientations

**Implementation of Lotze disarray** In order to implement Lotze disarray we simply displace all edgelets randomly irrespective of scale or orientation. This implies that fine detail will be much earlier destroyed than coarse detail as you increase the reach.

The visual impression is that the fine detail "diffuses" away and soon becomes unrecognizable. However, there is an important difference with blurring. Both blurring and Lotze disarray decrease the resolution, but their effect on the histogram is very different. A strongly blurred image becomes uniform, a strongly disarrayed one becomes "texture".



Image, blurred image and disarrayed image

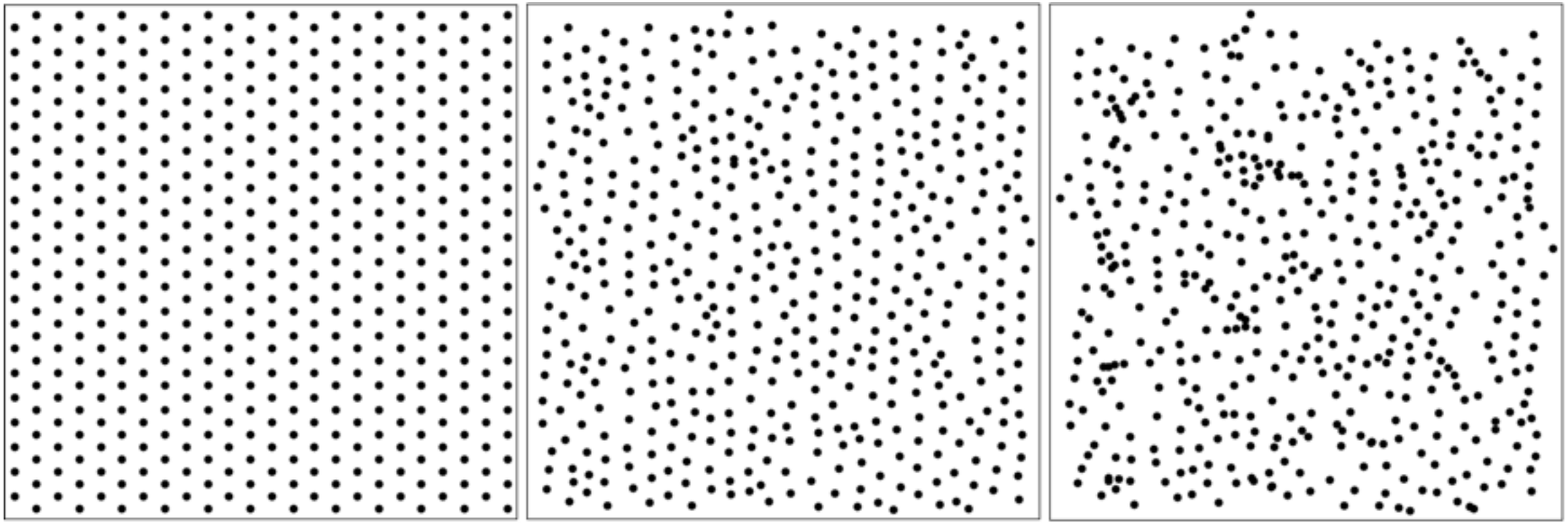


**Effect of the grain.** The disarray fields are Gaussian random fields of two dimensional displacement vectors. One constructs them by simply generating normal deviates for the Cartesian components. The normal distribution is very special in that this yields an isotropic two dimensional density.

Such fields, when generated pixel by pixel, are very fine-grained. Blurring them increases the grain-size, so Gaussian blur is used to set the grain.

In these pictures one has a contour plot of one Cartesian component at top and a (sampled for clarity) vector field at bottom. The field at right is coarse-grained, that at left is fine-grained.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!



**Effect of the reach.** The reach is an amplitude. Here we see a hexagonal grid, disarrayed by the same random vector field, except for the amplitude scaling. At left the reach is so small that you practically see the original image, at center it is seriously disarrayed, although the spatial order of the points has been largely retained, at right the reach is so large that the spatial order has become lost.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# A.2. Helmholtz disarray

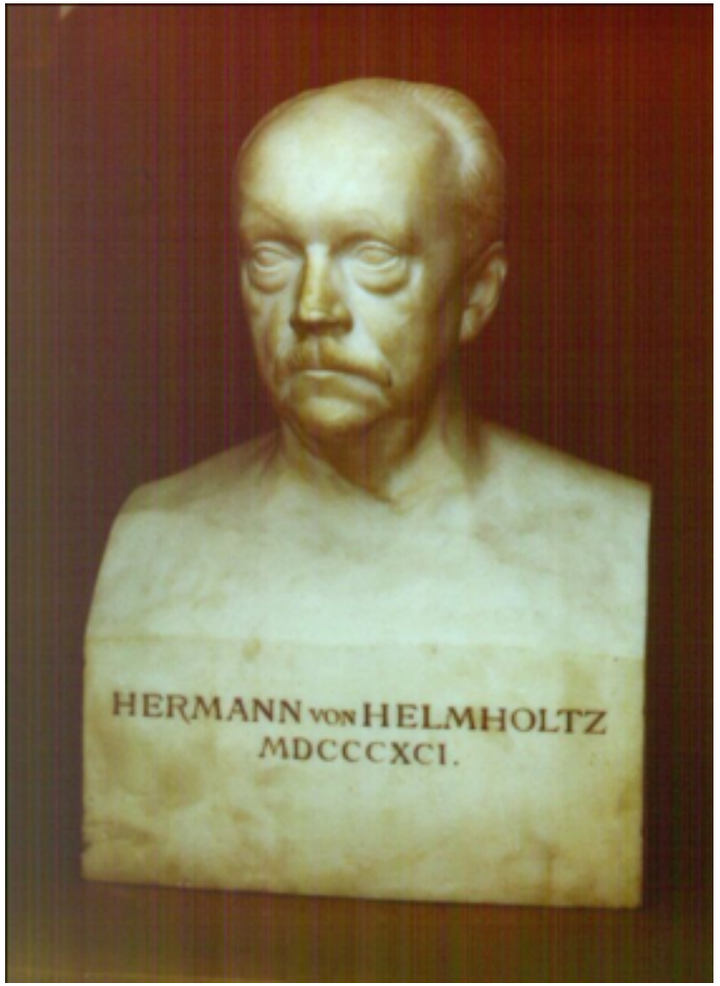
Helmholtz started his career as an army doctor. He noticed that in cases of acute toothache the patient is often unable to indicate the precise location, it being ambiguous whether the upper or lower jaw is involved. Pushing the teeth resolves the ambiguity soon enough. Helmholtz reasoned that the brain must regard the neurons at opposite positions as spatially coincident because their signals are necessarily correlated when chewing food. He proposes this as a possible physiological cue as to local sign.

This is a purely intrinsic mechanism, it does not rely on seeing anything, nor on motor intentions. It relies on the detection of correlations between neural signals and interprets these as geometrical overlap of receptive fields.

It is formally possible to generate the topology of the visual field on this basis, Helmholtz proposition is really extremely powerful.

In the Helmholtz perspective the overlap relations between receptive fields are crucial. The local "slop" is thus expected to scale with the average receptive field diameter. Thus we expect the disarray to be scaled with the scale.

In our implementation we scale both the reach and the grain with the scale, albeit with different factors. This yields a type of disarray that is very different from the Lotze type (A.1.).




Hermann von Helmholtz by Adolf von Hildebrand

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# A.3. Coherent disarray

There is still something missing in the Helmholtz disarray implementation, it is the lack of synchronicity between the disarray at different scale levels. Indeed, one expects relations between scales to figure in the Helmholtz picture, because the inclusion relations (which small ones are included in which large ones) are certain to play an important role.

There are at least two alternative proposals for local sign that we known of. One is Platt's idea of the mid twentieth century (though it can be traced decades, or perhaps centuries back) that eye movements shift straight lines within themselves. This would allow the agent to cluster receptive fields into mutually collinear sets.

Another proposal is more recent (about the turn of the millennium) by Ahissar. He suggests that the basis for local sign is not spatial, but temporal. Of course, eye movements play an important role again. Ahissar's mechanism is configured after the whisker systems common in many mammals (such as cats, one speculates that early primates may have had such systems too). The whisker hairs are tactile sensors used to scan the (very local) environment. Such systems work perfectly in the dark, thus they complement the visual system.

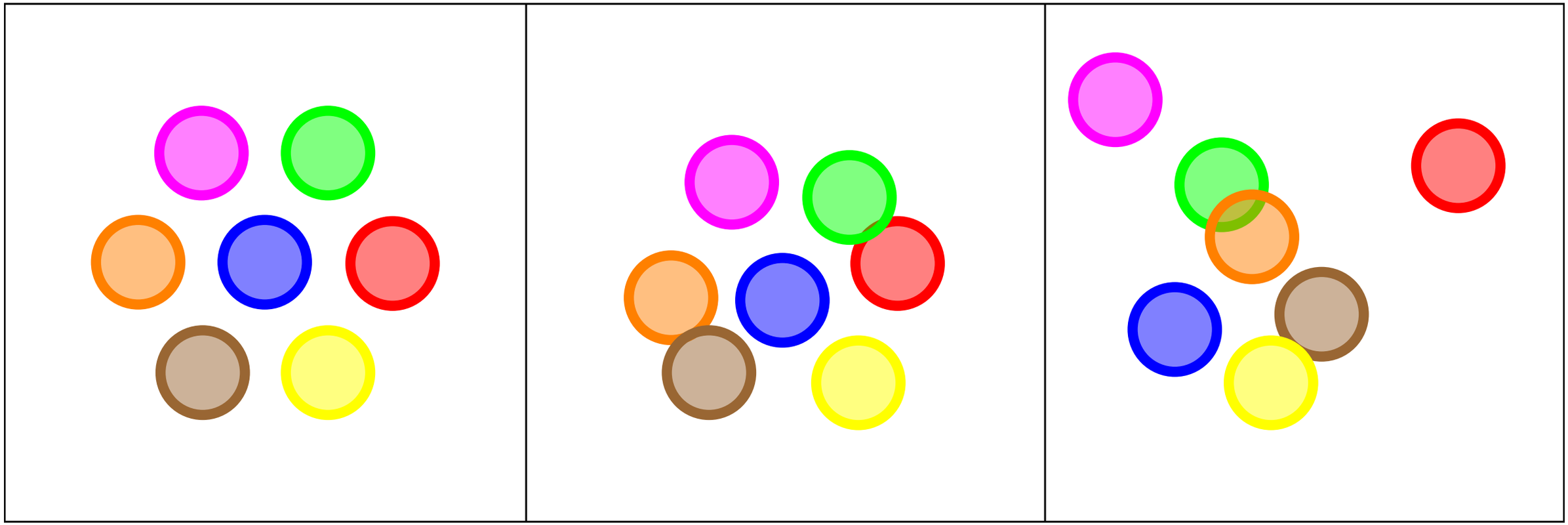
Such considerations led us to the "coherent disarray" in which coarse image structures drag finer image structures along with them.

The implementation is fairly straightforward in a scale-space setting. When a large structure is displaced, one adds its dislocation vector to those of the finer structures included in it. Of course, these finer structures are independently dislocated themselves, but in the Helmholtz way, that is to say, their dislocation vectors are scaled with their sizes.

This implementation makes that the total disarray field as applied to the finest details are of a fractal nature.

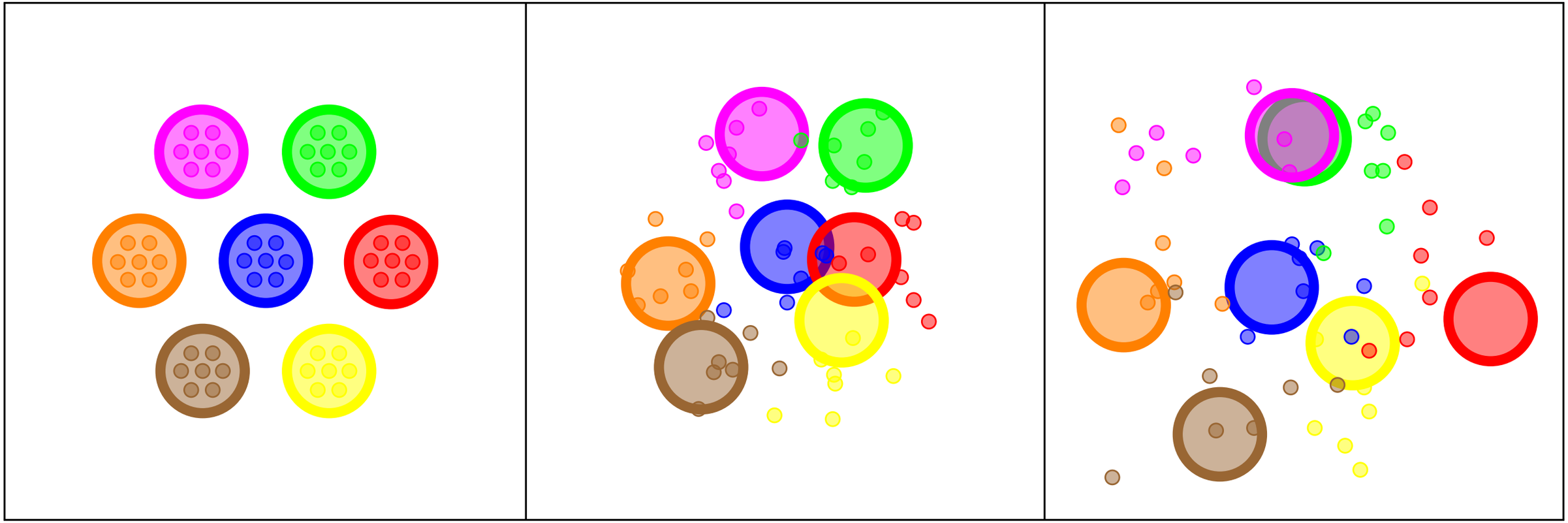
small waves riding larger waves riding even larger waves … and so forth is an apt model for the structure of coherent disarray

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

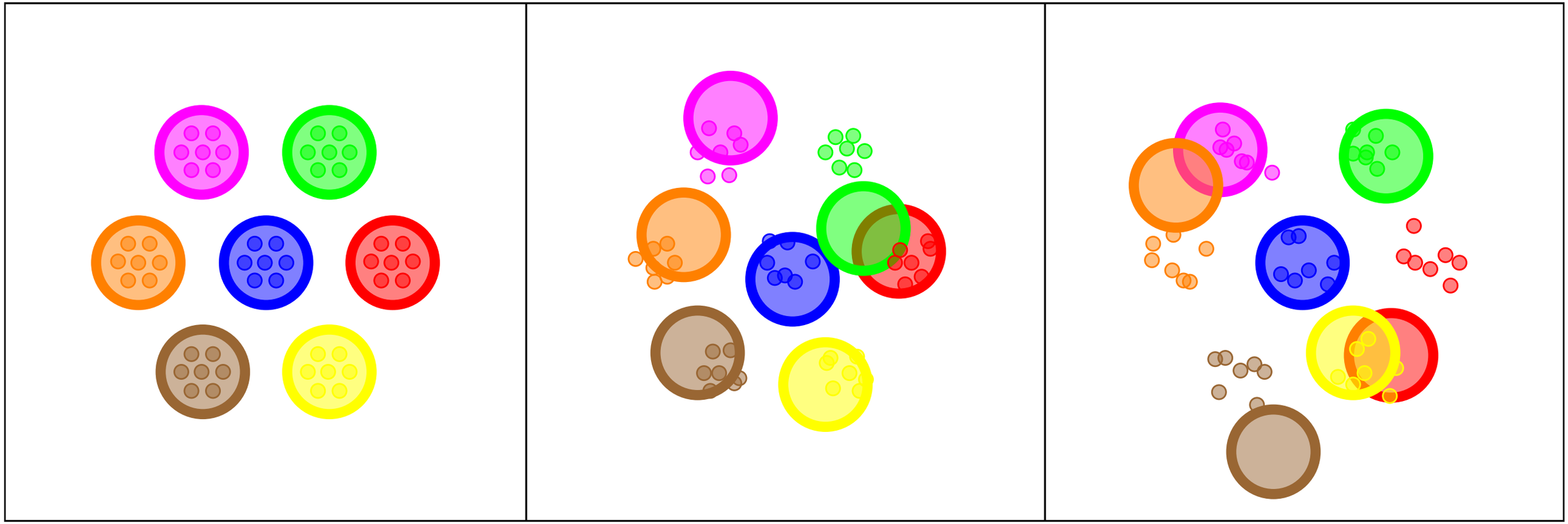


This illustrates the general idea of “disarray”. At left a simple, regular pattern. At center it has been disarrayed a bit. Notice that the patterns has become seriously deformed, although it topology is still intact. At right the reach is larger still. Here the topology is broken, for instance, the triangle formed by the blue, orange and brown points has collapsed and changed orientation.

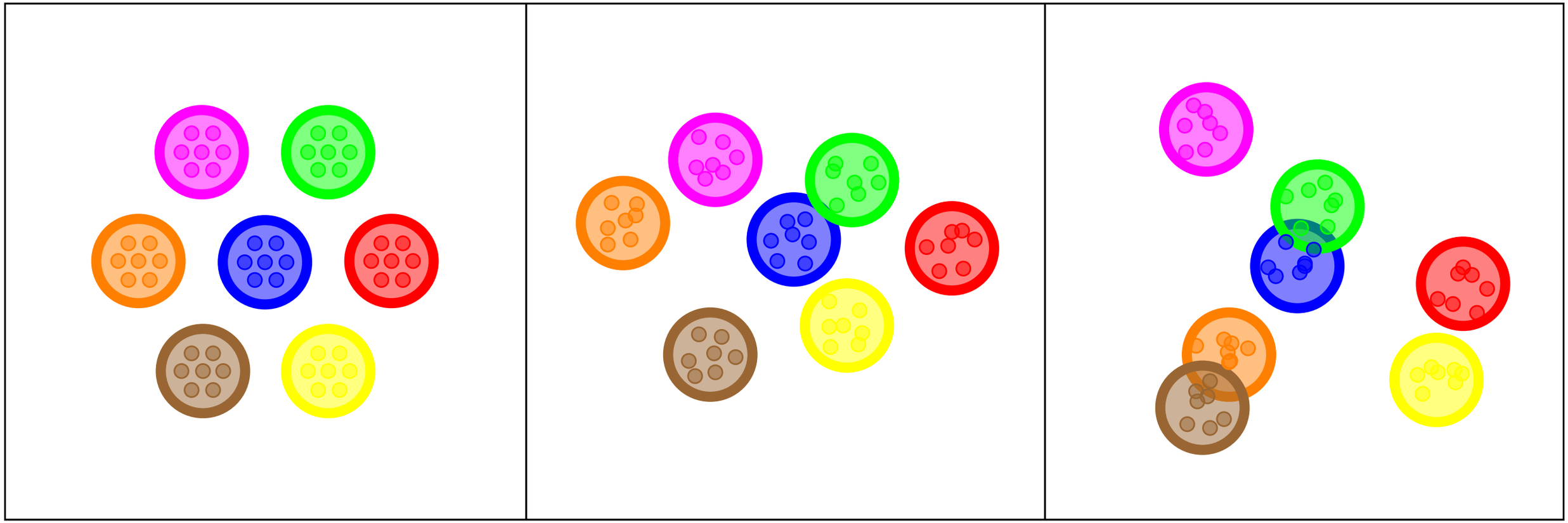
CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!



This is Lotze disarray. Notice that the relations between the small dots are completely lost, even when the large dots are still in a more or less okay configuration.



This is Helmholtz disarray. Here the small dots stay together. However, they float away (as a group) from the large dots that originally contained them. There is no coherence over scales.



This is coherent disarray. The coherence is over scales, so the small dots are always contained in their large containers. However, only as a group: within the group they are still shuffled. We have a fractal type of local disarray here.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

## A.4. Partially coherent disarray

Both the Helmholtz and the coherent disarrays make lots of sense. But, how could it be otherwise, the truth lies most likely in the middle!

No problem. We can simply introduce a "degree of coherence" that smoothly interpolates between the pure Helmholtz disarray (totally incoherent, degree of coherence 0) and coherent disarray (totally coherent, degree of coherence 1)!

It is of much interest to set a not too large, but noticeable reach and vary the degree of coherence over the full range. This will allow you to gain a good intuitive notion of the importance of synchronicity between scales.

# B.0. Lotze disarray

The simplest type of disarray field is the Lotze disarray. Its only structural parameter is the grain (the reach is just an overall amplitude that we ignore here).

The disarray is based on the simplest possible basis, an image with a value draw from a normal distribution of zero mean and unit variance at every pixel. We follow up with a Gaussian blurring to define the grain. An independent pair of such Gaussian random fields yields a disarray field: we simply regard the fields as representing the Cartesian coordinates of disarray vectors.

Of course, there are technical points, such as which random generators to use, how to do fast (but precise) Gaussian blurring, and so forth. We ignore such technicalities here, but they are of importance in an actual eidolon factory.

In the demo you can view the Lotze disarray fields as a function of grain (controlled by the UP/DOWN arrow keys). The direction is coded by hue, the magnitude by saturation.

## B.1. Helmholtz stack

In the case of the Lotze disarray the same grain and reach (but fresh, independent instances at each scale level) are used throughout.

In the case of the Helmholtz disarray the grain and reach are proportional to the scale. Thus we generate a noise "stack" with a different disarray field at any of the (discrete) scale levels. The noise stack is thus constructed as parallel to the scale-space.

The UP/DOWN arrow keys let you travel through the various layers. Notice that the images have been scaled for clarity. They clearly display the grain. For the reach you have to look at the numerical values reported in the green letters at center right.

## B.2. Coherent stack

In the case of coherent disarray large image details drag finer details contained in them along. This is very simply implemented by adding the noise from the low resolution layers to that of the finer resolution layers.

Thus the "coherent stack" is simply the progressively cumulated Helmholtz stack.

Although simply enough in principle, building a fresh coherent stack involves serious number crunching. It is the major reason why generating eidolons is a comparatively slow process, taking several seconds for our JAVA implementation running on an Intel Core 2 Duo 3.06Ghz in 8GHz RAM. Much time can be saved by trading computation for memory. However, there is a limit to that because one requires constantly fresh, independent noise samples.

It is most interesting to look at the finest grained layer of this stack, because it has received additive contributions from all other layers. The noise field looks rather different from the single scale noise fields you see in the Lotze case. In the coherent stack you deal with noise fields having fractal structure.

# C.0. Image scale-space

In the "C" department you find "datas stacks" containing layers of the same image at different resolution. In the most obvious case the image is simply the original, fiducial image. We refer to such data stacks as (image) "scale-space".

The scale-space notion is crucial for the eidolon factory since it is allows us to perturb image structure in a finely controlled manner.

The UP/DOWN keys let you move through scale-space. We have discretized the scale domain in factors of square root of two, stating from the pixel scale and ending at (roughly) image size. Notice how enormous this range is, the image becomes hardly recognizable at the low resolution levels. On each level there is (obviously) spatial structure. By toggling between two layers you become aware of the structural changes involved in a scale change. This "deep" image structure depends upon the spatial structure. Of course, because low resolution layers can be computed from high resolution ones.

Some intuitive understanding of scale-space theory is very helpful if you want to understand the algorithms used to produce image eidolons.

**Images and scale.** All vision researchers are familiar with Fourier analysis. In Fourier analysis one decomposes an image into "sine-wave gratings". In Fourier synthesis one sums all components obtained by the analysis so as to obtain the original image again. This has led people to say things like "images are composed of sine-waves". This is WRONG, although they can be decomposed into sine-waves.

Consider: is a sausage composed of slices? No, but you can decompose it into slices. The analysis produces the parts. Or consider this: are the numbers A=1+1+1+1, B=1+3, C=2+2 and D=4 different? If you consider them to "have different pairs" your answers should be yes. But no, the number 4 has no parts. You produce them in saying "four is two plus two" (wrong!) although "two plus two is four" (correct!). The case of Fourier analysis is not different in this respect.

The sine-waves are objects of different "grain", given by their wavelength. Thus Fourier analysis lets you address the deep structure of the image. You can see this when you omit the very fine grained sine-waves from the synthesis: the image will look "blurred" and "has lost resolution".

This example shows why it might be useful to analyze and then synthesize an image: after all, you only end up what you started with! The point is that the analysis gives you "hooks" that can be used to perturb image structure in a focussed manner. This is also true for Fourier analysis, although it is only good for global operations like blurring the whole image. That is because the parts - that are the sine-waves - are global. They have no significance to the phenomenology. You will rarely have sine-waves in your visual awareness, unless you are a vision researcher.

In most cases there is little relation between mutually distant region in an image, then global methods are useless. That is why "Gabor functions" have become popular in vision research. Gabor functions are useful if you need to concentrate stuff both in space and in the spectral domain. They were useful in Gabor's original application (acoustics), but it is hard (we find it impossible) to think of applications in image processing. Vision researchers use them for arbitrary (historical) reasons, yet believe them to be of fundamental importance. Even receptive fields in the brain are believed to implement Gabors. Against all odds, one might say, for the wriggling behavior of Gabors (all the way to infinity) make them unlikely enough candidates for the job.

an image

the same image except that the phases of the Fourier components where randomly scrambled by random angles of the same variance (S.D. 46°) for all sine-wave gratings. This is an alternative class of eidolons. Indeed, it has been used in vision research. Do we like it? No.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

**Local image structure.** The ecological fact that one part of an image is not a great predictor for the structure of other (distant) parts, suggests that if images have "natural parts" at all, these parts should be local.

What is meant by "local"? In any case, local has to be related to the resolution or "grain-size" of the image. For instance, an "eight-by-eight grains icon" might be considered a local natural image part. The image could be described in terms of a quilt of such icons. Operations could be done at the icon level. And so forth.

This implies that "local" has different meanings according to the scale-space levels to which it is applied. It reminds on of Euclid's definition of a "point" as "that which has no parts". Applied to scale-space a point would be represented by a single "grain", which has a different size at each scale level. This perhaps reminds one of the painter Kandinsky: "a point may grow and fill the whole image". He was experiencing a vision of scale-space.

# C.1. Macchie scale-space

A "macchia" (Italian for "blob" or "blotch") is a technical term in painting. Paintings may be conceived as constructed from "macchie" (plural of macchia).

The image scale-space essentially gives you the macchie structure. This is best appreciated when you "posterize" the image in a (very) limited number of levels.

Try the UP/DOWN keys to view the blob structure at many scales. At an intermediate range of scales you get representations that might easily be turned into a macchie painting.

Macchiaioli ("blotch painters") at the Cafe Michelangiolo about 1856

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

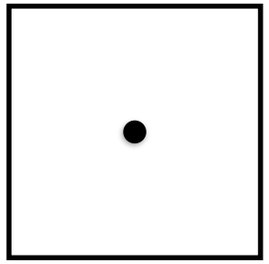
"Macchia based painting: capturing Italian campagna from train"

(http://jerryrosspittore.blogspot.nl/2012/01/theory-behind-macchia.html)

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# C.2. Pointlet stack (DOGs)

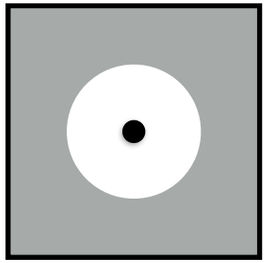
Is a point the most local object imaginable? Perhaps most people would agree, except that some might find the notion of a point as "infinitely small" not imaginable. To the latter, we would agree. However, the point we are making here is different. Consider an "image of a point", perhaps like this:



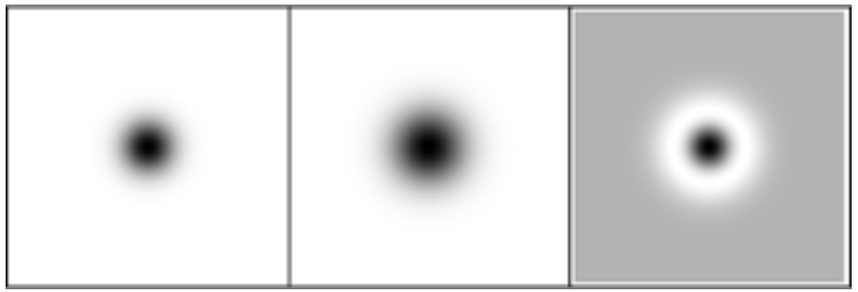
Now how "localized" is this point? You only see the point because of the contrasting background. Here we changed the background:

Is the point still there? Perhaps, but not in visual awareness.

A point in visual awareness necessarily has a contrasting background. In that sense, the "point" shown in the first image fills the whole frame! Here is an attempt to localize the point, by decreasing the size of its background. This succeeds, though only partly, because we have created a structure that seems to have lots of non-point-like detail!



A better away to obtain a localized point is to let the point create its own background by blurring it. And, since we're at it, we also present the point itself as a blurry macchia in order to get rid of its sharp boundary, which introduces structure at the very different (much finer) scale.



This works well, we have constructed a "pointlet", which captures the notion of a "localized point".

Such a pointlet has the profile known as "DOG (Difference Of Gaussians) receptive field profile".

When the sizes of the point and its surround are the same - this is a formal limit, for even if of the same size, the surround is still "larger" than the point! - you obtain the "Laplacian operator". The Laplacian can be defined in terms of the average of second order directional derivatives of all orientations. However, a more intuitive understanding is as an operator that extracts "local excess". This is the Laplacian as the derivative of an image with respect to scale.

The structure of scale-space is due to this Janus-face nature of the Laplacian. (Formally, this is the "diffusion equation", which says that the two understandings of the Laplacian are equivalent.)

# C.3. Edginess stack (edge finders)

"Edge finders" or "edge detectors" are commonly applied in image analysis. The idea is that image objects are bounded by edges, which are thus one-dimensional local objects. They are simply there in the image, your task is to "find" or "detect" them.

We suggest that this puts you on the wrong foot. Edges are in the mind, not in the image. Indeed, if you run an edge detector on the photograph of some natural scene, you will find that it reports edges at any pixel, except perhaps a few. This is well known, so people will tell you that you should use a threshold so as to ignore most of the edge detector's output. Setting the threshold such that one percent of the pictures are "edges" will often be considered "too low"! The "correct" threshold is the one that yields the edges as you see them when you look(!).

Although common practice, this simply won't do. Instead of imposing arbitrary thresholds, we will say that running a so-called "edge detector" results in another image that represents "edginess". Even better, we notice that such edginess is simply the magnitude of the gradient of the image. If we use this formal language we don't need the concept of "edge" in the analysis any more. It may be reintroduced in the subsequent synthesis, which has the character of an interpretation, or "presentation".

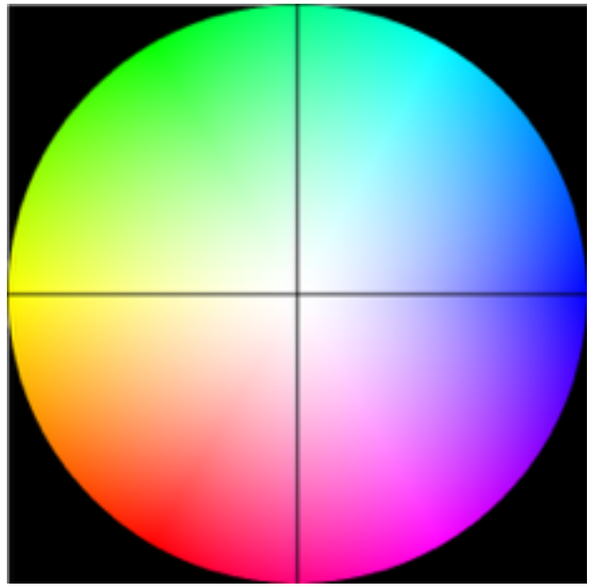
Edge detectors are bilocal objects. You take two points measure the image intensity at each. Then you report the difference of these divided by the mutual distance of the points as an estimate of the image gradient in the direction from the first to the second point. A true edge detector is obtained in the limit of zero distance. Such an object encapsulates the points. It has become "atomic" in that you have no access to the points. Only the direction and the scale of the points (assumed equal) are retained as characteristic properties of the object.

Formally, such atomic objects are "tangent vectors" or "directional derivatives" (same thing). You implement them by receptive fields that are the derivatives of a Gaussian, a point. Indeed, they are "derivatives of a point". These objects behave like vectors in the plane. For instance, the usual vector addition works. Thus you can construct edge detectors for any direction by suitably weighted addition of just two of them. (This has led to the notion of "tunable filters". but really all this is just basic geometry.)

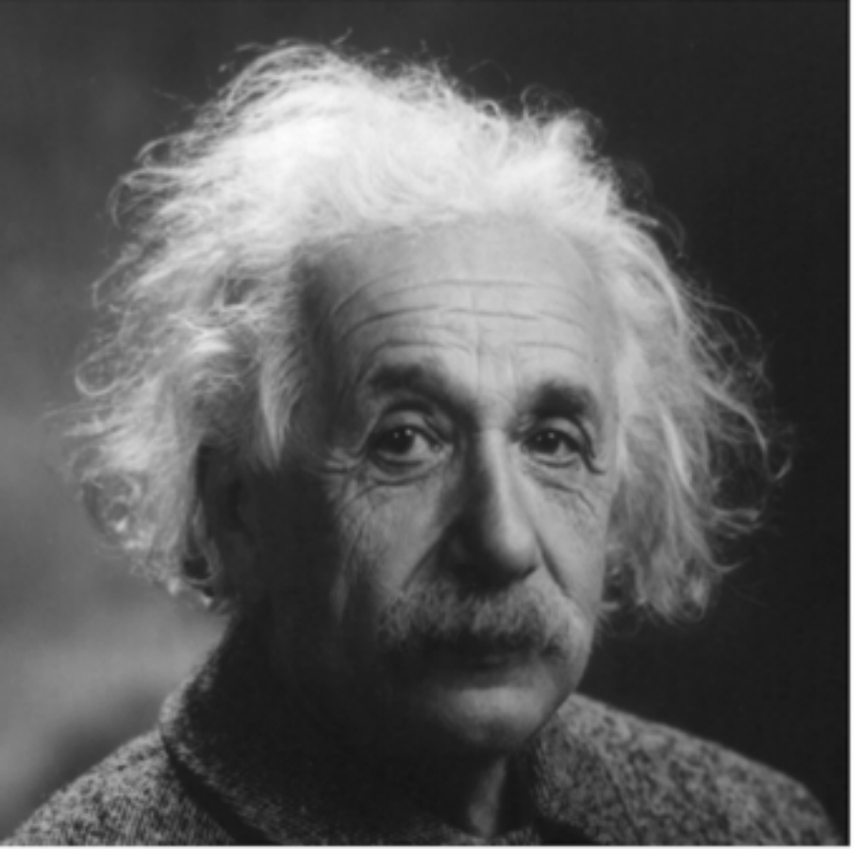
Thus the edginess stack is constructed from two scale spaces, one for the horizontal and one for the vertical image dimensions. This automatically takes care of all directions.

In the demo you use the UP/DOWN arrow keys to navigate through the scales. The image shows the edginess indicated by hue (the direction) and saturation (the magnitude).

Notice that there is "edginess" everywhere, but that there is a range of scale levels in which you tend to see that "edges were detected".

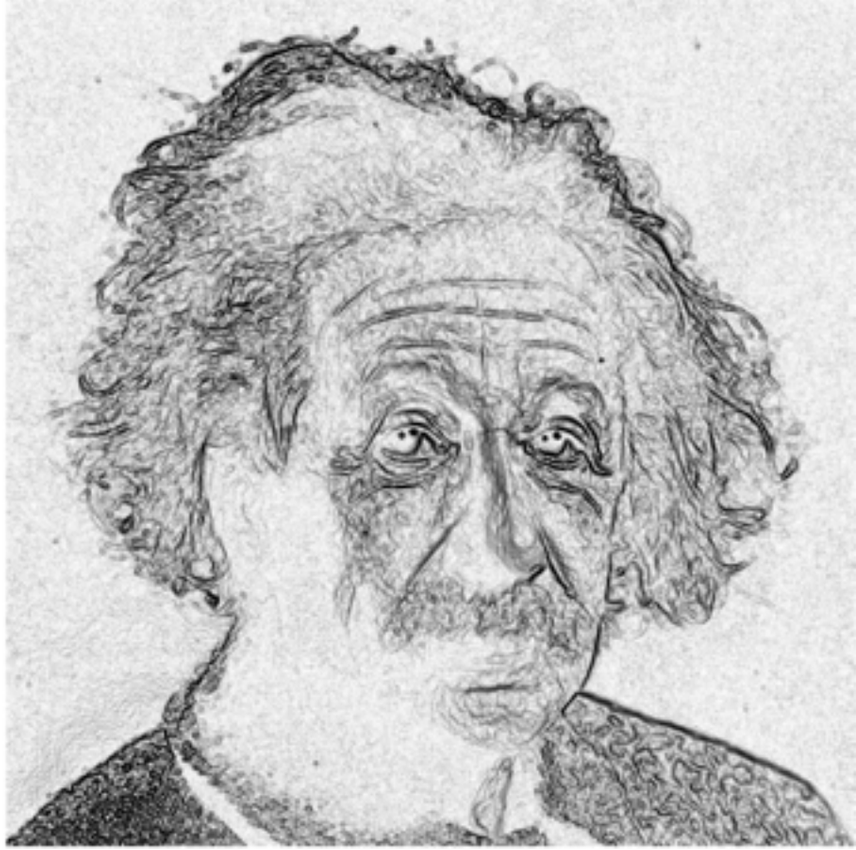


CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!



A fiducial image

Its "edges" as detected by Photoshop©. Since there are no edges everywhere, apparently a threshold was applied. Yet, there are probably more edges than many would appreciate.



This is "edginess". It has a value at every pixel, thus it cannot be held to "detect edges". Yet most people are visually aware of localized edges. These are in the mind, rather than in the image.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# C.4. Edgelets stack

Although "edge detectors" are supposed to get you "edges," (no one could define "what they are"!) even in the best of worlds, their output fails to evoke the notion of "edge" to your mind's eye (your immediate visual awareness).

In order to actually EXPERIENCE an "edge" you need to present the appropriate image structure. Here (current) vision science has nothing to say, the "experts" (REALLY, though the majority of scientists would certainly oppose the idea) are the visual artists. Their message is primarily communicated through their works (and rightly so!) although a minority attempts to put their reflective thoughts - different from their visual awareness! - into words. Society owns them a great deal for that, but one should perhaps correct for the fact that in the transition between immediate visual awareness and reflective thought, everything suffers a "sea change".

How to define "edge"? Perhaps as a region of transition between two macchie. Two abutting macchie suggest an "edge", whereas a suitably executed "edgelet" (see below!) may suggest "two macchie", even though these were never painted! This is the familiar "watercolor effect" described by Baingio Pinna.

This is obvious in the antique map of the Netherlands shown at right. The provinces are clearly delineated and even seem "colored" although only "edgelets" have been indicated. Pure "edginess" is fully incapable of achieving that.

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

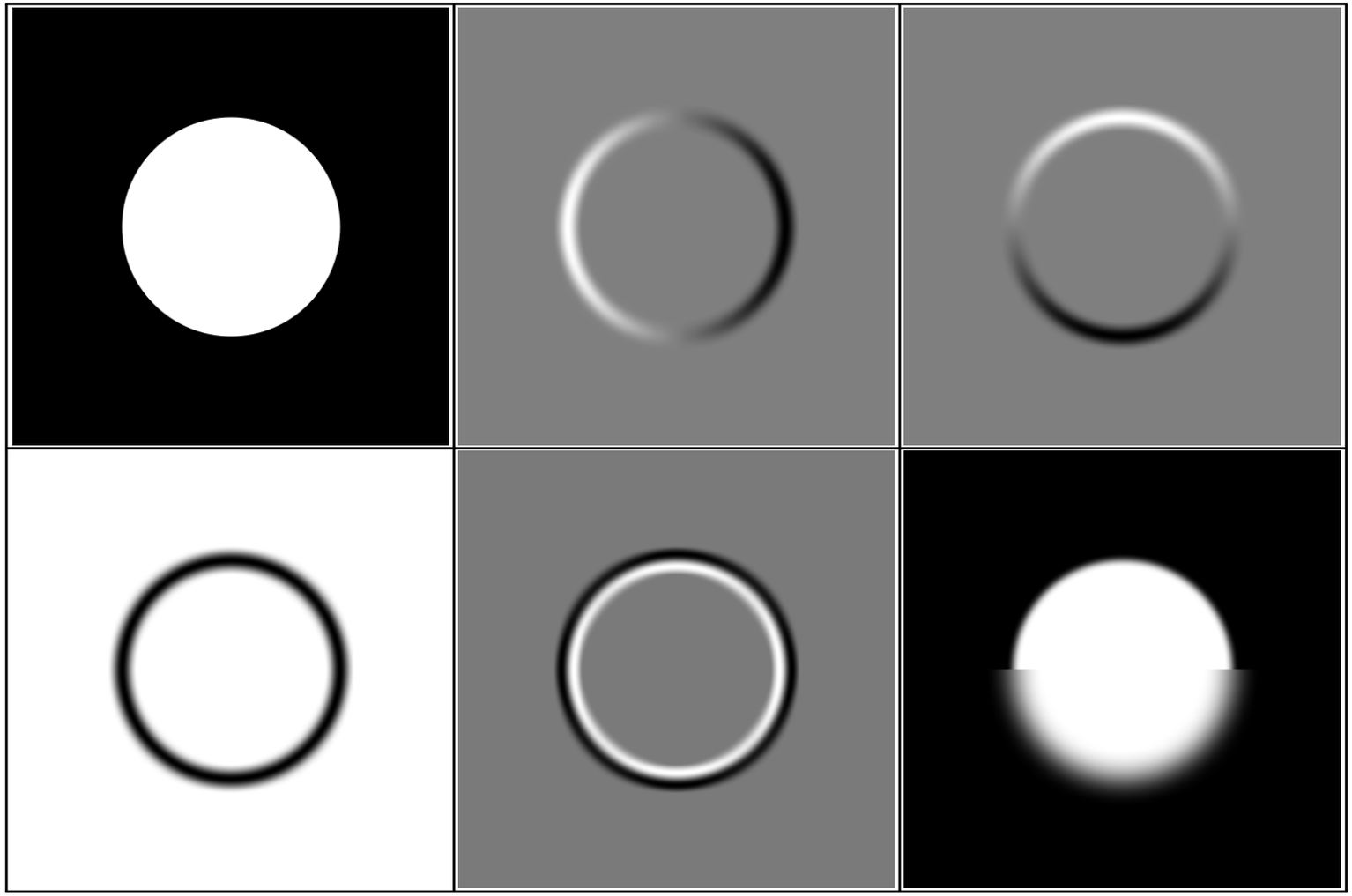
One defines "edgelets" in the same way as we defined "pointlets", you simply compare an image to a slightly lower resolution copy: in the difference image only the pointlets (if there are such) and edgelets (very likely) are retained, whereas the macchie are lost - they have been subtracted away.

This is a very important insight: whereas the edginess reveals "where the edges are", the Laplacian PRESENTS them as "edgelets".

The eidolon factory is based upon this insight. The eidolons obtained by synthesizing the RANDOMLY DISARRAYED EDGELETS obtained by an analysis into a Laplacian scale-space stack closely fit the psychogenesis of visual awareness.

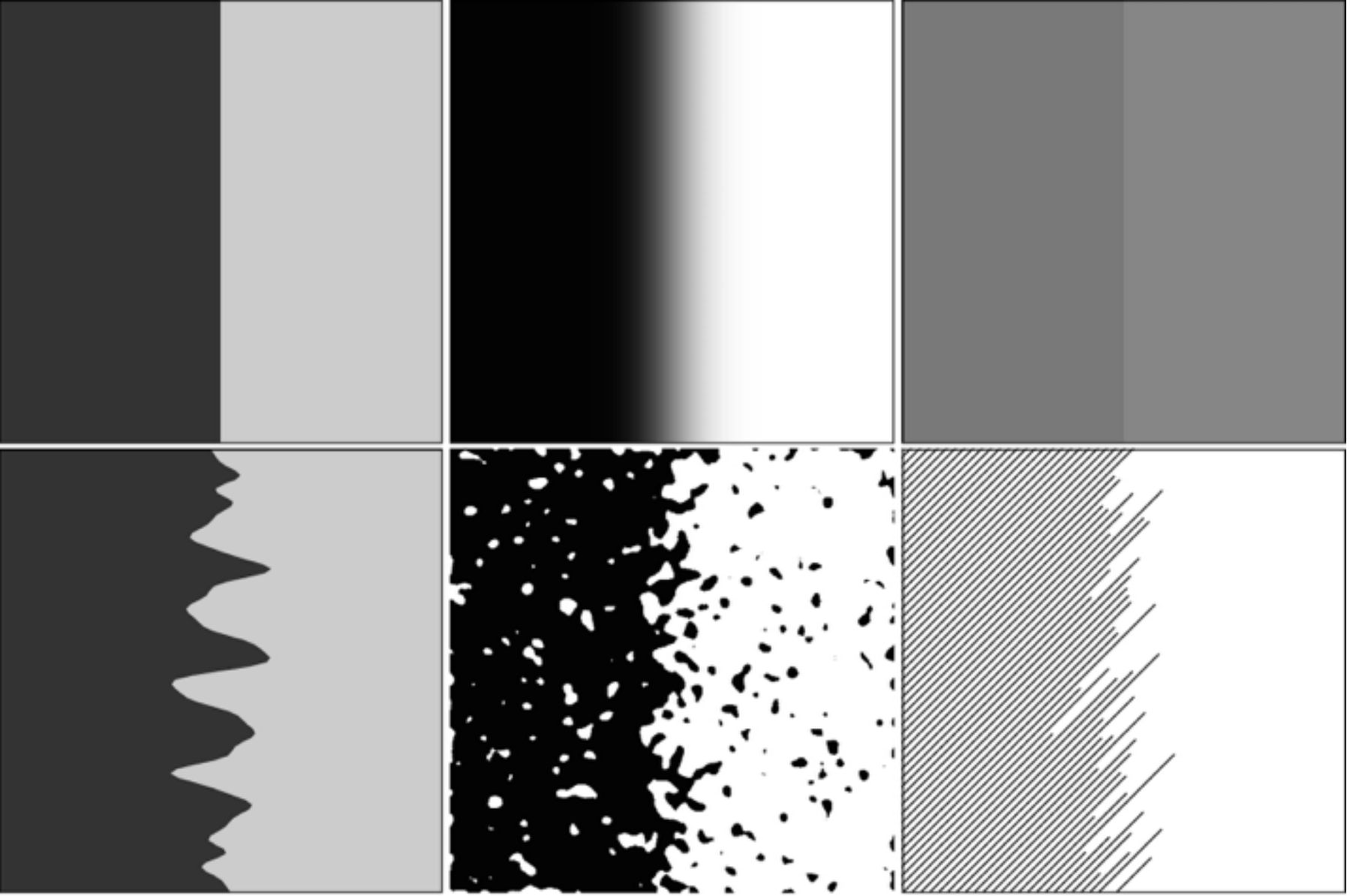
No doubt, this is why such eidolons looks "natural", sometimes "painterly" or esthetically pleasing. This in contradistinction to - for instance - eidolons obtained by phase scrambling the Fourier spectrum on an image.

This closely resembles the insights that may be gleaned from a variety of texts on "how to paint". The "edge treatment" is every bit as important as the treatment of the macchie. The two are complementary. You may paint edgelets through macchie and you may paint macchie through edgelets. But the edgelets are NOT edginess (what edge detectors yield), nor are they "sharp". They come in a huge variety. "Getting the edge treatment right" is an important achievement in painting.



Right to left, top to bottom: An image; horizontal derivative; vertical derivative; "edginess" (square root of the sum of squares of the derivatives); the edgelet representation (here the Laplacian); a higher (top part) and a lower (bottom part) version of the image - their difference is (to a good approximation) the Laplacian. This illustrated "what edgelets are", intuitively.

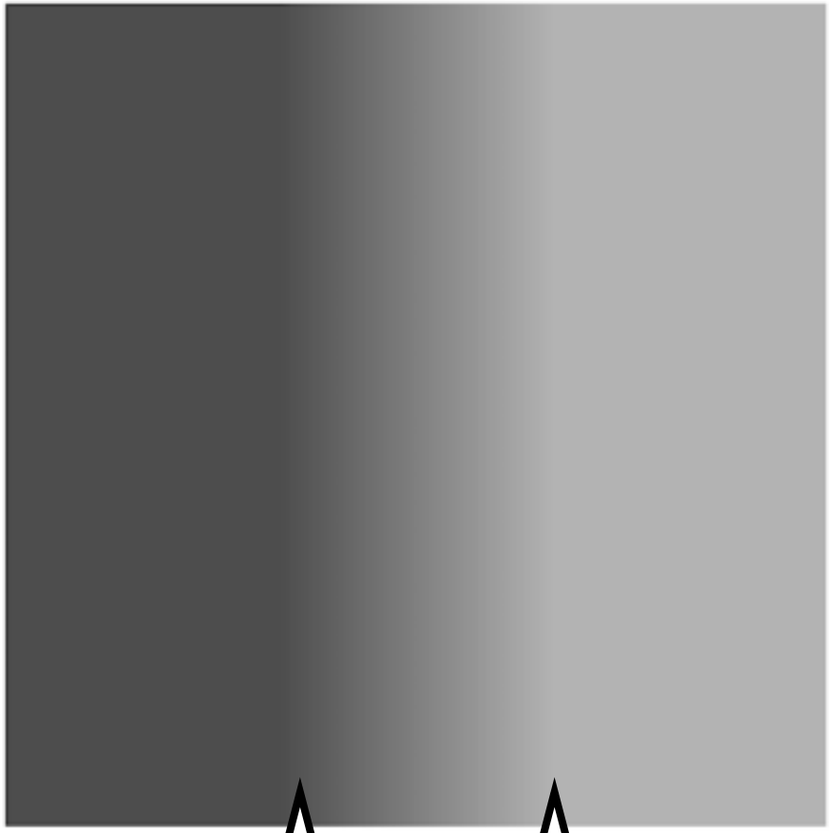
CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!



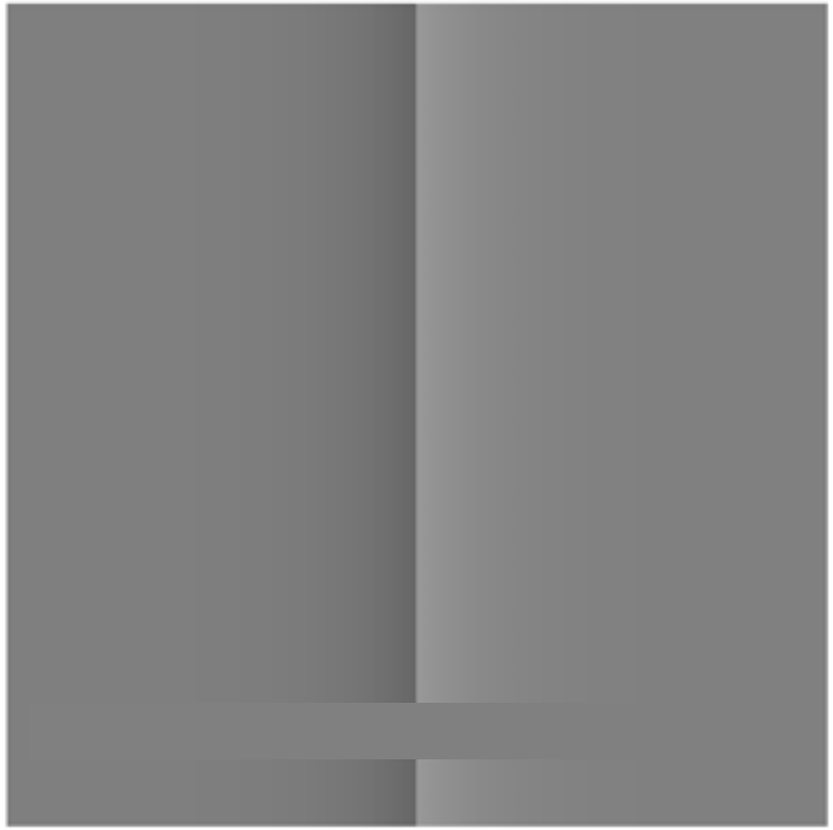
In the visual arts these count as bona fide vertical edges. As you see, edges come in infinite variety.

Just show us your favorite "edge finder" and it will be easy enough for us to draw an image on which it would fail!

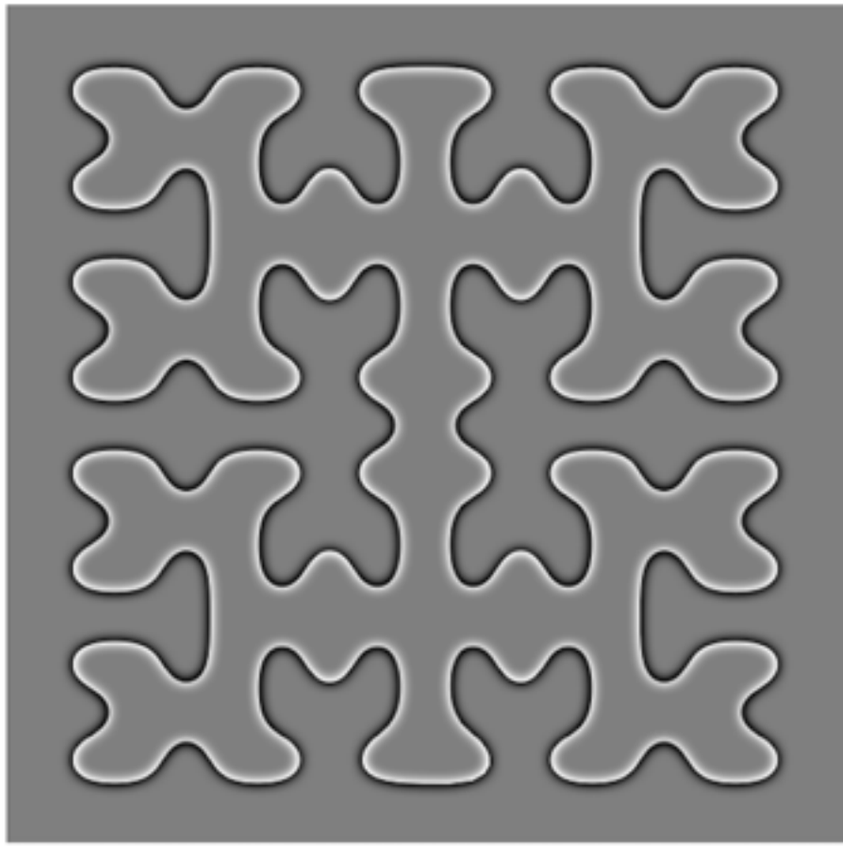
CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!



A linear ramp stimulus. Do you see the two "Mach bands"?



The Cornsweet-Crane illusion: an edgelet produces two macchie (the lower horizontal bar is a uniform gray!)



Another instance of a (very convoluted) edgelet producing two macchie. One macchia appears as "background" for the other one. This is Baingio Pinna's "watercolor illusion".

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

# C.5. Ridginess stack (line finders)

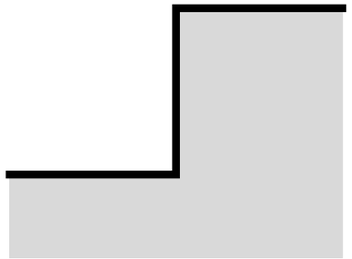
A "line finder" is a second order directional (or rather orientational) derivative. It can be understood as a trilocal operator composed of three mutually collinear points, the center points bisecting the segment defined by the outer two. The operator takes three point samples and reports the difference of the value of the center point with the average of the values at the outer points, divided by the square of the separation. Thus the object reports "excess" in a certain orientation. In the formal limit of the points getting arbitrarily close together, one obtains an atomic object characterized by a certain orientation. The kernel profile looks like the receptive field of the type of single cell denoted "like detector" by Hubel and Wiesel. Technically it is a second order directional derivative. It is a tensorial, not a vectorial object. A basis can be defines as a triple of such objects, all others (ANY orientation) can be obtained by suitable linear combination.

What is of interest is that a pooling of all line finders of all orientations yields the Laplacian. Thus the line finders can be regarded asa further analysis of the Laplacian with respect to orientation.

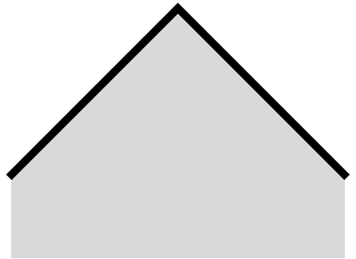
Pooling all Laplacians yields the original image, this is indeed intuitively obvious, because the Laplacian is the difference between adjacent scale levels.

As an immediate consequence, the image can also be synthesized from the ridglets stacks. This yields an even finer handle on image structure. Instead of merely disarraying edgelets, one may choose to disarray the various orientations selectively. (We have not implemented this in the present demo.)

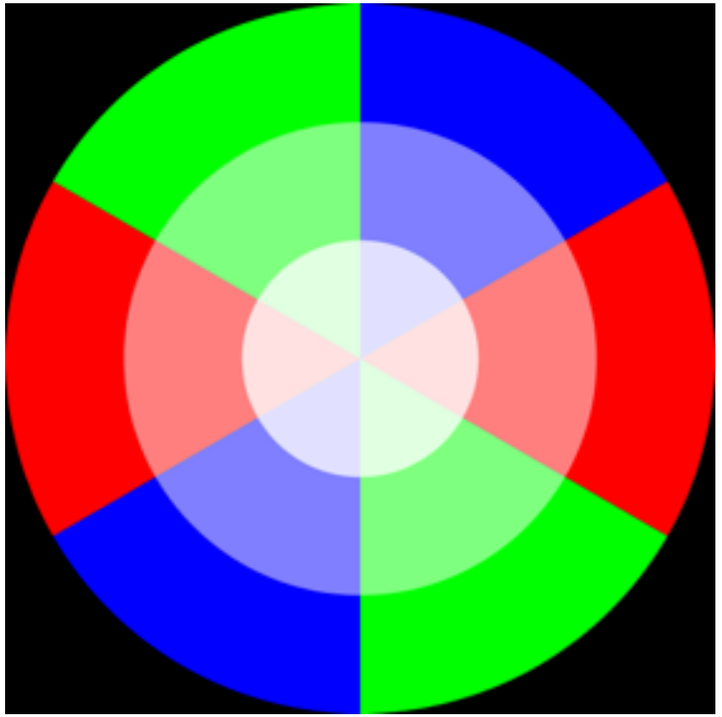
Why "ridginess"? Well, we use ridginess in the same way we use edginess. The local singularities in the second order orientational derivatives are ridges and ruts. But, of course, the ridges exist only in the mind. Ridginess exists ant every pixel. The "Mach bands" effect suggests that the psychogenesis of visual awareness takes extremes in ridginess into account.



an "edge": zeroth order discontinuity detected with first order derivative

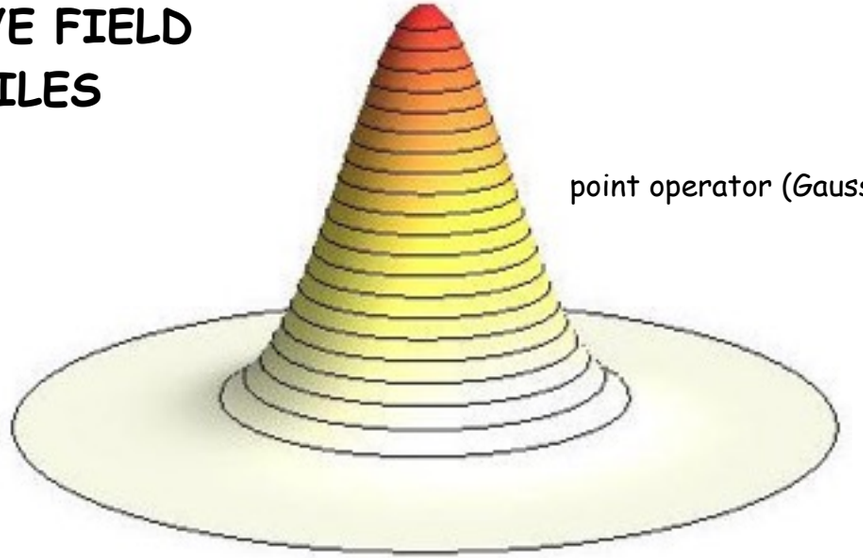


a "ridge": first order discontinuity detected with second order derivative

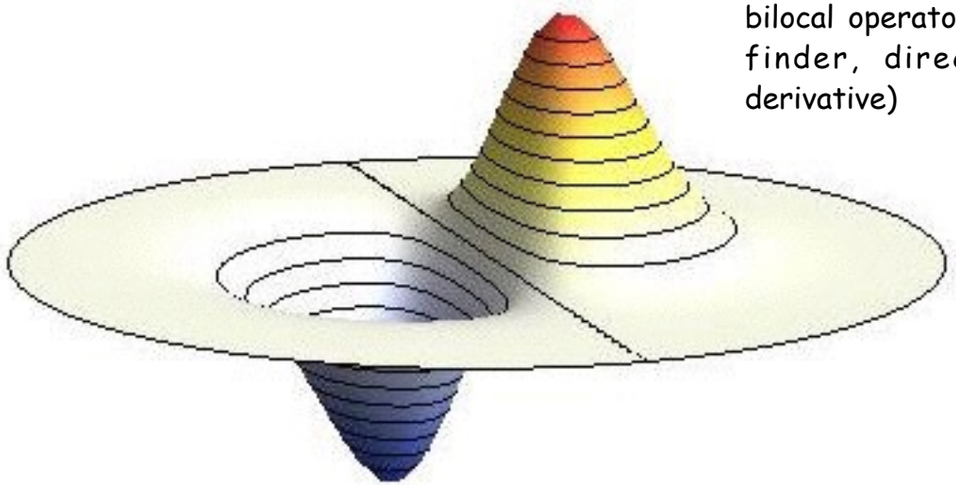


The orientation scale repeats by 180° so it is different from the direction scale which repeats by 360°. Here hue encodes orientation whereas saturation encodes magnitude.

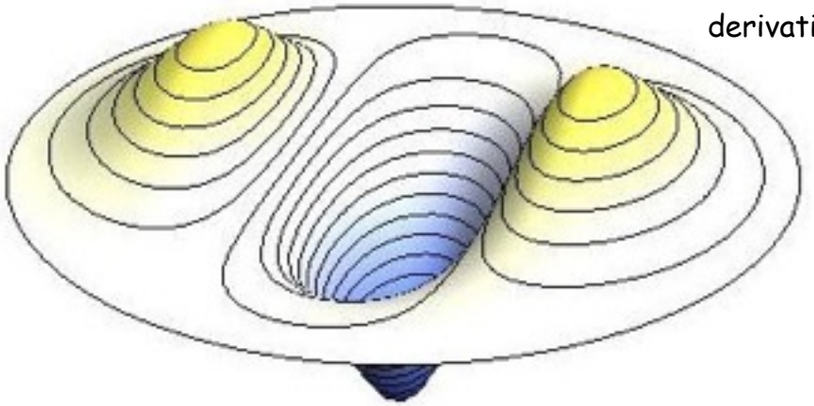
# RECEPTIVE FIELD PROFILES



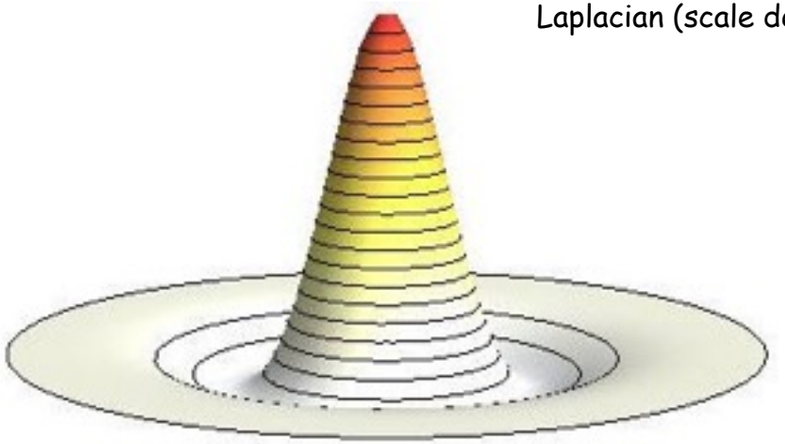
point operator (Gaussian)



bilocal operator (edge finder, directional derivative)

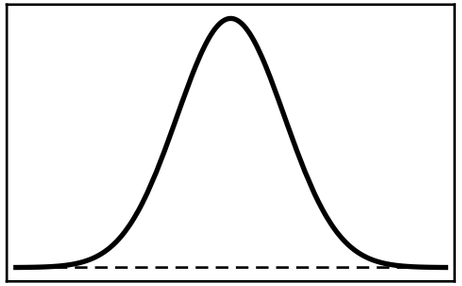


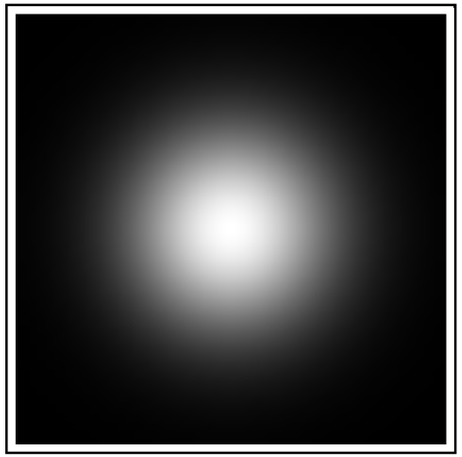
trilocal operator (line finder, 2nd order orientational derivative)

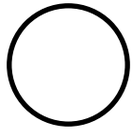


Laplacian (scale derivative)

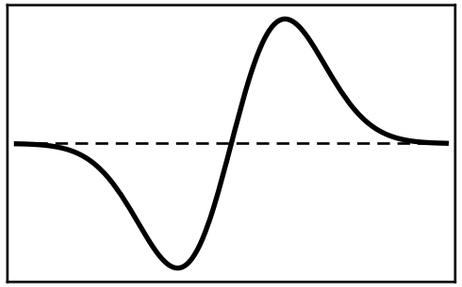
CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

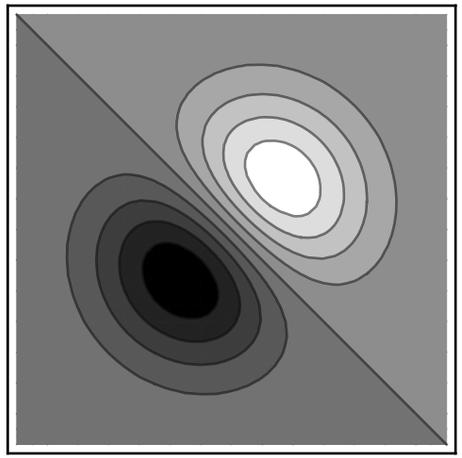






The "point operator: A RF cross section, RF density map and a symbolic icon

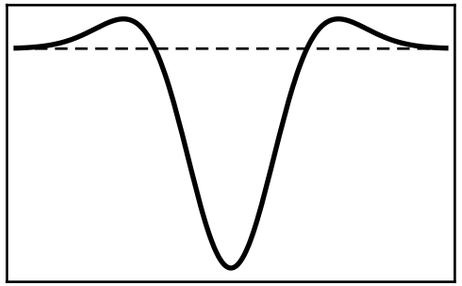


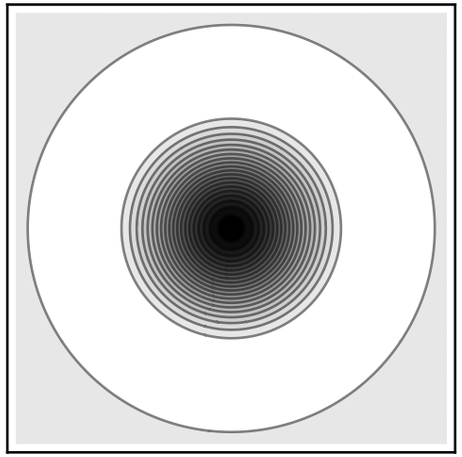


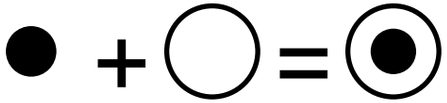
The "bilocal operator: A RF cross section, RF density map and a symbolic icon

The final icon suggests that this is an ATOMIC entity: You cannot look inside to the the two points

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!

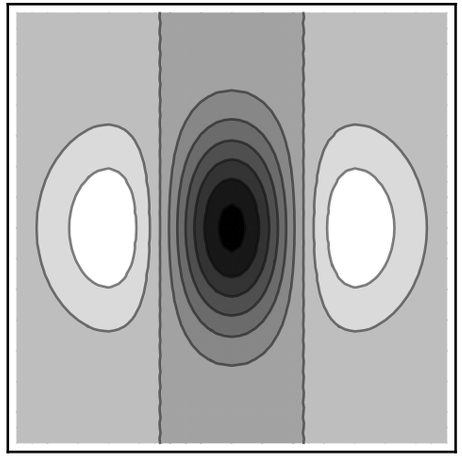


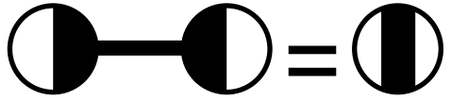




The "Laplacian operator: A RF cross section, RF density map and a symbolic icon

It is also a bilocal operator, but the two points coincide in space, they have different "sizes"

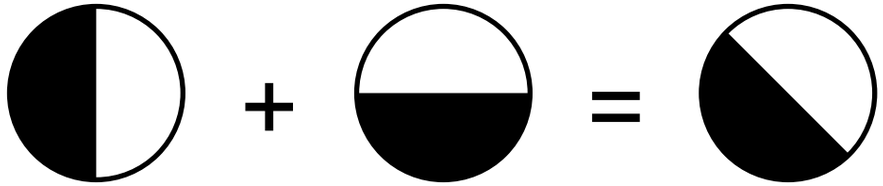




The "trilocal operator: A RF cross section, RF density map and a symbolic icon

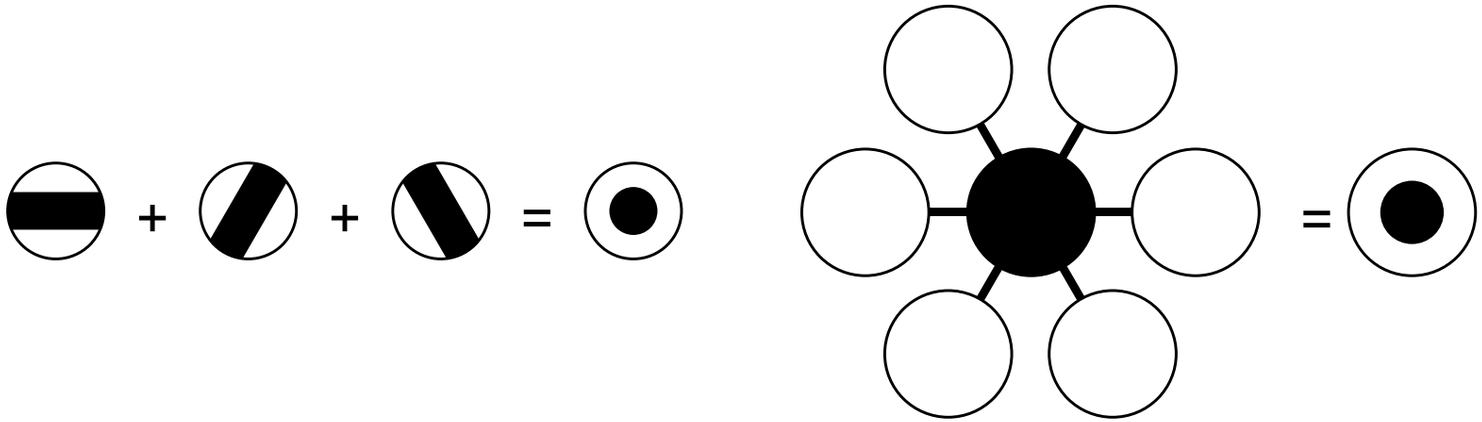
It is an ATOMIC entity. You cannot "see" whether it arose from three points or from two bilocal objects. It makes no difference, much like 4 is 1+1+1+1 just as much as it is 2+2. It makes no difference to "4".

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!



The sum of two edge finders is an edge finder given by their vector sum

Thus you need only two edge finders in order to be able to make all others



Two way to construct the Laplacian

Both constructions yield the same result.

The rightmost construction reveals the Laplacian as an excess operator

The leftmost construction shows that local excess is composed of local ORIENTED excess

CLICK IN WINDOW TO RETURN – LEFT/RIGHT ARROWS FOR PAGING – PATIENCE: COMPUTATION TAKES TIME!