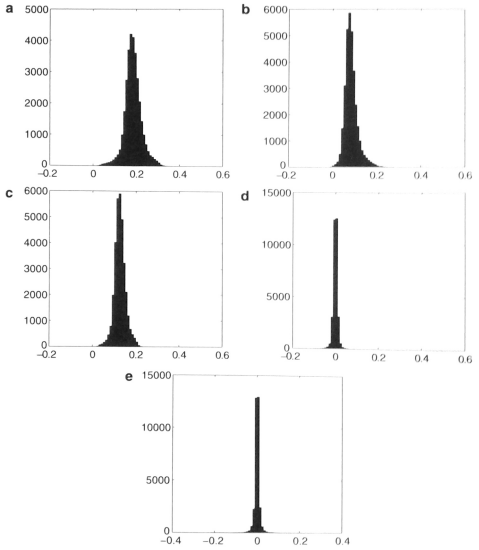
Visual feature learning

Winter 2013

Roland Memisevic

Lecture 12, March 12, 2013

Dependence of ICA features

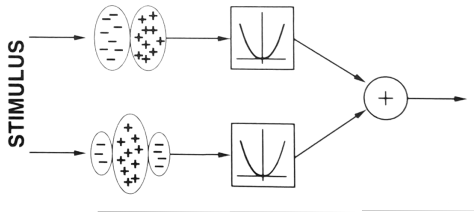


- a: $f(s) = |s|$
- b: $f(s) = s^2$
- c: $f(s) = |s| > 1$
- d: $f(s) = \text{sign}(s)$
- e: $f(s) = s^3$

Dependence of squares

- How can we fix this?
- Idea: Learn another feature learning layer on top of transformed (eg. squared) features.
- This leads to complex cells, so it seems to make sense biologically $\rightarrow$ one benefit of using squaring non-linearities.
- But we may also try to understand *why* these dependencies exist.

Benefits of squares (II): Local Fourier energy



- Recall that the sum over the squared responses of two quadrature Gabor features (90 deg phase-difference) can detect an edge independently of phase and polarity.
- To this end, we need to sum over squares across directions:

Derivative filters

- In computer vision it is common to detect edges using horizontal and vertical derivatives.
- To this end, we can convolve the image with two filters like

$$h_x = (-1\ 0\ 1)$$

(to compute horizontal derivatives) and

$$h_y = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

(to compute vertical derivatives)

- Each of the two resulting output images will contain the derivative in one direction.

Image gradients

- The gradient $\nabla I(m, n)$ of the image I at pixel (m, n) can then be written

$$\nabla I(m, n) = \begin{pmatrix} (I * h_x)(m, n) \\ (I * h_y)(m, n) \end{pmatrix}$$

- Other filtermasks can be used, like

$$h_x = (-1\ 1), h_y = (-1\ 1)^{\mathrm{T}}$$

- Or diagonal variants, like the "Robert's cross" operator:

$$h_+ = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, h_- = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$I * h_x$

$I * h_y$

$I * h_+$

$I * h_-$

Edge strength and direction

- After computing $2-D$ directional derivatives, one can compute edge strength and direction from the 2-D gradient as follows:
  - Strength = Norm of the gradient
  - Direction = $arctan(h_y / h_x)$
- To compute the norm, we have to *sum over squared filter responses*.
- Note that, to compute direction, we can alternatively use a panel of filters that cover all directions.

Image contours $|s(m, n)|$

Benefits of squares (III): motion energy model

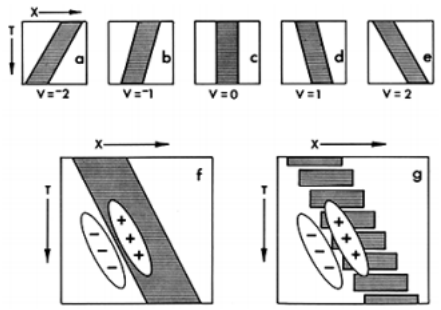
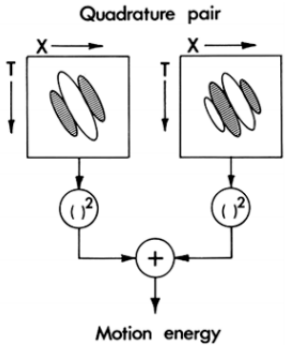


Fig 5. a-e (x, t) plots of bars moving to the left or to the right at various speeds. f, Motion is like orientation in (x, t), and a spatiotemporally oriented receptive field can be used to detect it. g, The same oriented receptive field can respond to sampled motion just as it responds to continuous motion.

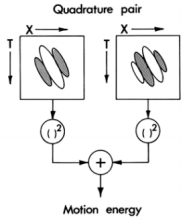
- Adelson & Bergen, 1983
- To to separate *motion* from *what is moving* you may compute a (local) amplitude response.

The motion energy model



- Adelson & Bergen, 1983
- For $2-D$ images, use the $3-D$ Fourier transform.
- This will compute a spatio-temporal, "oriented" energy.

Single-frame complex cells and "motion" in a single image



- If we replace the time, T, in the motion energy model with another image coordinate, we get a complex cell applied to a single patch.
- Thus, we may think of an image patch as a slowly transforming (1d) scan-line.
- This leads to another explanation for why we always get Fourier and Gabor features: Nearby scan-lines are related through *translation*.

Benefits of squares (IV): stereo vision

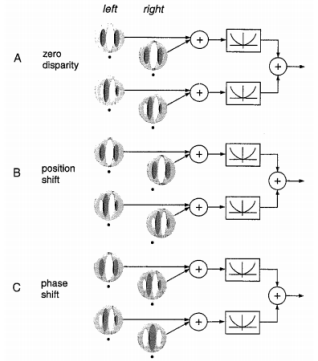
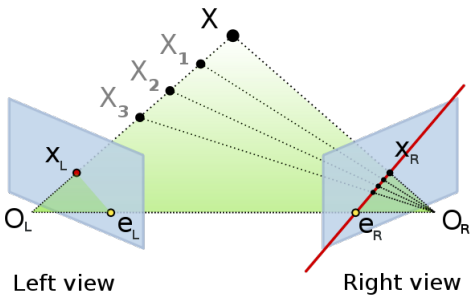


FIGURE 3. Disparity preferences of binocular energy neurons. (A) Zero disparity preference. (B) Non-zero disparity preference is introduced by shifting the positions of both right-eye receptive fields by the same amount (relative to the reference points). (C) Non-zero disparity preference is introduced by shifting the phases of both right-eye weighting functions by 90 deg.

- Fleet et al., 1996
- see also: "ODF" model
- Why does this work?

Epipolar geometry



- (wikipedia)
- The distance of 3-D point X is a function of the spatial relationship between X_L and X_R.

Multiview geometry

- Figuring out, where a world-point that is seen in the left image ended up in the right image can be difficult and messy. (The area around the point will not look exactly alike in the two images.)
- It gets slightly easier with images that are pre-processed, so that cameras may be thought of as being in parallel, because then we can search along horizontal lines.
- This is known as “rectification”.
- One can generalize this to more than 2 views.

Multiview geometry

- The core operation is *matching* of views across cameras, to find points that coincide.
- The common approach is to crop patches, compute feature vectors, search across the epipolar lines, and declare two patches as matches, if they are close in feature space.
- How well it works depends of course on the features, the similarity measure, etc.
- Still, matching is hard and incredibly noisy, but there are various hacks to help clean up (eg., “RANSAC”).

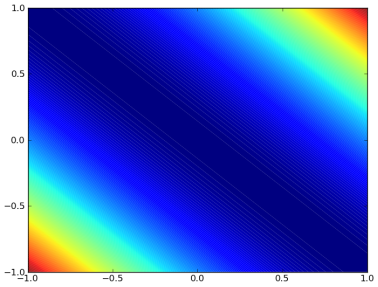
And in the brain?

- If the camera geometry is fixed, matching amounts to always searching along the same lines.
- Two points match if they are (approximately) translated copies of each other.
- So the sum of two phase-shifted squared Gabor features can detect matches, too, because it detects translation by a certain amount.
- A panel of energy responses with different phase shifts will be a *population code* for the translation.

Squaring vs. products

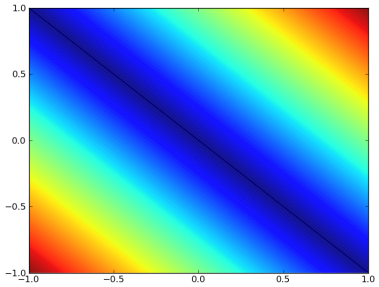
- Since
$$(a+b)^2 = 2ab + \text{const}$$
we can think of the complex cell response as a way to compute products.
- The product-version of the Adelson & Bergen model has been proposed around the same time.
- It is commonly referred to as *cross-correlation model*.
- It leads to hidden units that are tuned to inputs that are identical:

Square tuning



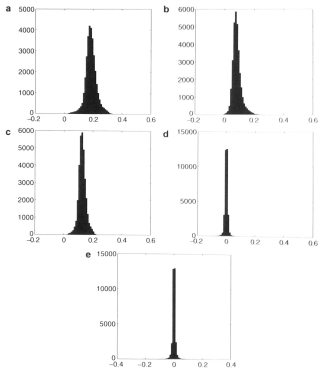
- square-of-sum
- Other even-symmetric functions will show the same behaviour:

Other even-symmetric functions



- Example: abs(sum)

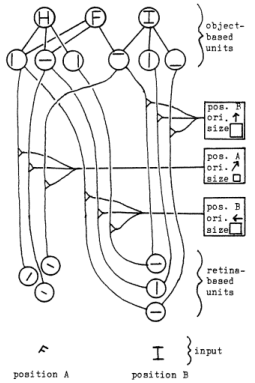
Suspicious coincidences



- a: $f(s) = |s|$
- b: $f(s) = s^2$
- c: $f(s) = |s| > 1$
- d: $f(s) = \text{sign}(s)$
- e: $f(s) = s^3$

- Is the unexplained variability due to *coincidences* (caused by motion, depth, contours, etc.)?

Mapping units



- The usefulness of products for motion, invariance, etc. has been suggested a long time ago. Eg.:
- Hinton, 1981: “mapping units”
- von der Malsburg, 1981: “dynamic mappings”

Active dendrites

- Hidden units that can gate connections between other hidden units behave more like a transistor:
- Weights are not passive “cables”, but they modulate their behavior depending on context.
- Eg. Archie & Mel., 2000

Benefits of squares (IV): AND and OR

- A hidden variable that can detect the product of incoming variables can also be thought of as computing something similar to a logical AND.
- This is in contrast to simple cells and the standard neural network hidden units, which compute something more like an OR.
- Zetzsche & Nuding, 2005