Vision in Human and Machine - Tutorial 4
Perceptual grouping with the CLM

Heiko Wersing

In the folder for Tutorial 4 you find the Matlab code for the routine `clm.m` which simulates a Competitive Layer Model (CLM) architecture. The target of this tutorial is to investigate different feature interactions that allow the modelling of Gestalt-based perceptual grouping on some test images.

1 Generating a set of features

As the first step, a call to the routine `get_image_features` is used to obtain local image feature vectors derived from an image. This routine is used to generate local intensity and orientation features. It works in the following way:

- Read in an image, and generate the convolution output of two orthogonal Gabor filters on the image. The two orthogonal filters are used to represent the local orientation gradient with a normalized vector. Assemble a feature vector `m` for each pixel, where the image intensity is larger than a threshold `img(x,y)>[0.1...0.3]`, where for the feature r
  - `m(1,r)` is the X position of the pixel
  - `m(2,r)` is the Y position of the pixel
  - `m(3,r)` is the gradient in horizontal direction (Gabor output 1)
  - `m(4,r)` is the gradient in vertical direction (Gabor output 2)
  - `m(5,r)` is the local image intensity
  - The vector `m(3:4,r)` is normalized (direction vector).

  The pixel intensity threshold is used to reduce the simulation to a smaller number of features and neglect the dark background.

2 Generating the feature interaction matrix

For the set of features now an interaction matrix `f(r,s)` has to be computed that carries the interaction between all pairs of features r and s . The name of the function is `get_interaction_matrix`. Look at the definition of the function and write the missing code. The function additionally has to return a vector `fpos(r)` which is the sum over all positive interactions of the feature `r`. This is required for setting the right CLM inhibition/competition parameter `J`.

For each pair of features `r,s` an interaction `f(r,s)` has to be computed based on the feature vectors `m(:,r)` and `m(:,s)`. If you define

```
pos=[1 2];
dir=[3 4];
intens=5;
```

Then `m(pos,r)` is the position vector and `m(dir)` is the local direction vector of feature r. Additionally the local pixel intensity is available. The following principles can be followed for the interaction design:

- Features that should be grouped together should interact positively.
- Features that should not be grouped together should interact negatively.

To obtain reasonable results for real Gestalt stimuli, however, the two components have to be balanced appropriately.

The first example stimulus is `images/proximity1.pgm`. Compute a suggestion for the interactions and visualize the results with the function `show_interaction`.

3 Gestalt grouping

Now use the CLM implementation to obtain a grouping result on the chosen image and feature interaction. Optimize the feature interaction until you can robustly perform an adequate grouping on the image. Sometimes it is necessary to slow down the speed of annealing by choosing values of η closer to one.

Design Gestalt interactions for the examples given in the `figure directory` and try to obtain succesful perceptual groupings with the CLM:

- `proximity1.pgm`
- `similarity1.pgm`
- `proximity2.pgm`
- `continuity1.pgm`
- `connected1.pgm`
- `continuity2.pgm`