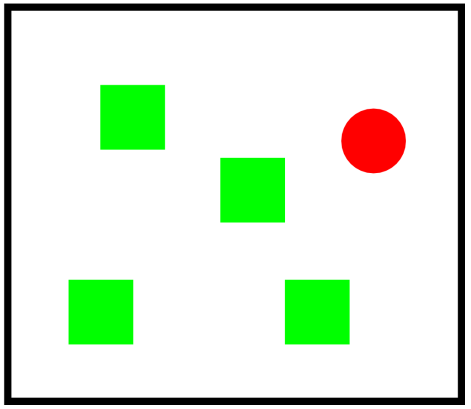
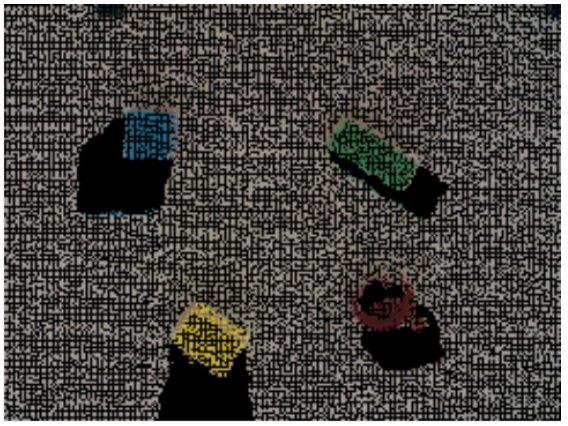
Object recognition in Dynamic Field Theory

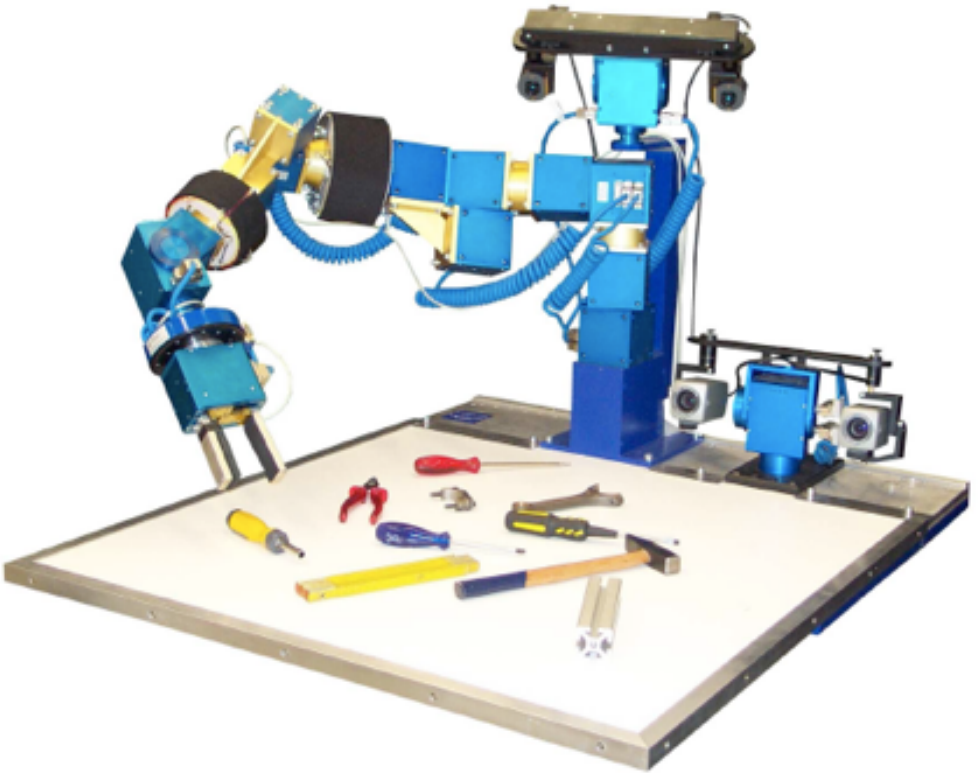
Oliver Lomp – Institut für Neuroinformatik – 16. 7. 2015

Why we need object recognition



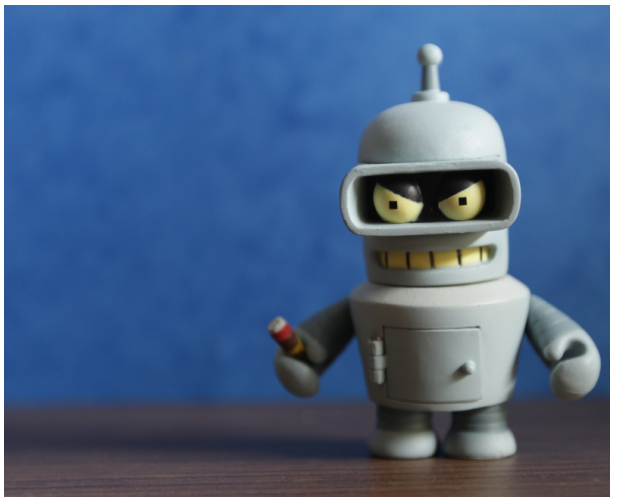


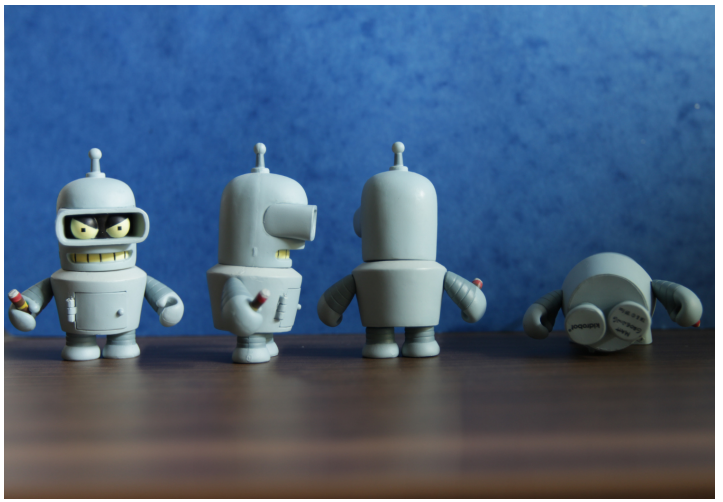
vs.



What is difficult about object recognition?

What is difficult about object recognition?





What is difficult about object recognition?

- 2D image of a 3D object
- Infinitely many projections

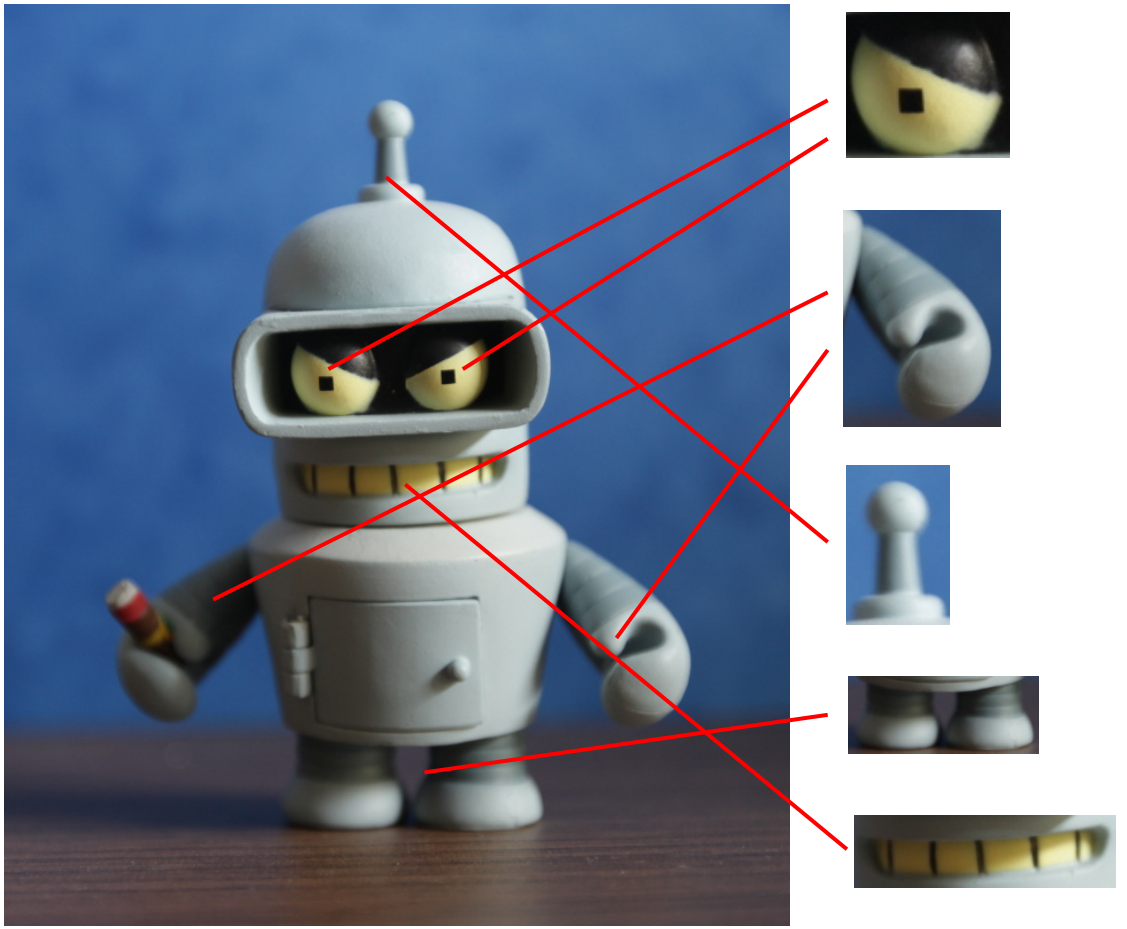
  → same object never looks the same

Common solution: invariance

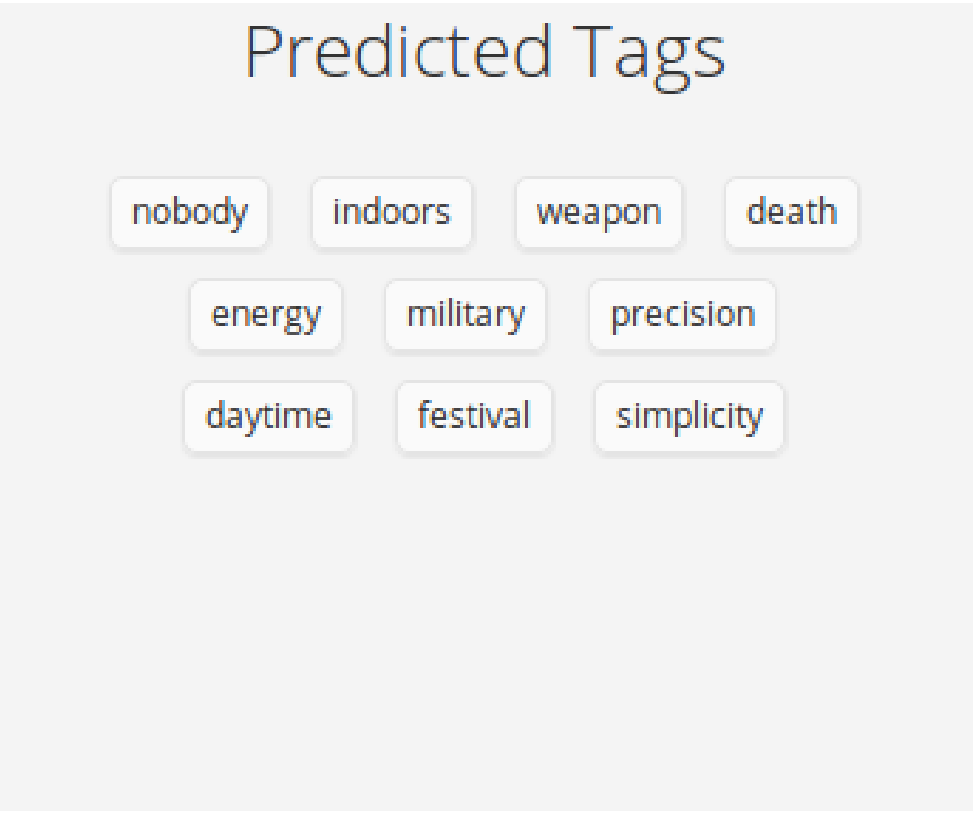
- Trade-off: invariance vs. discriminance
- Invariance reduces information

  → Know what, but not necessarily where

  → Category vs. instance



Invariance: example



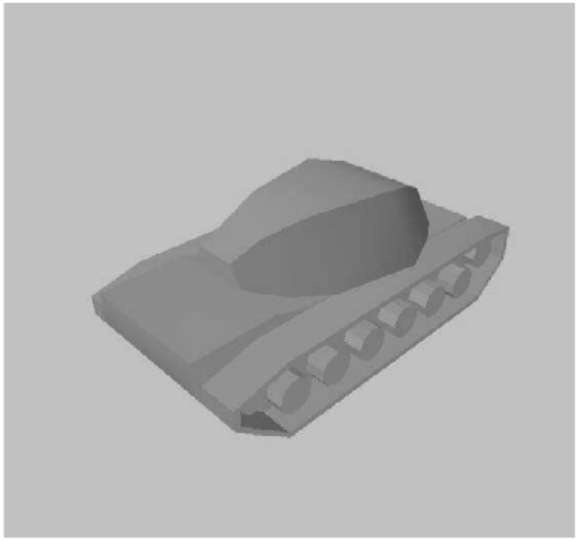
(tags by clarifai.com)

Object pose

- Instead of pose-invariant features: estimate pose, use it

Arathorn's map-seeking circuits

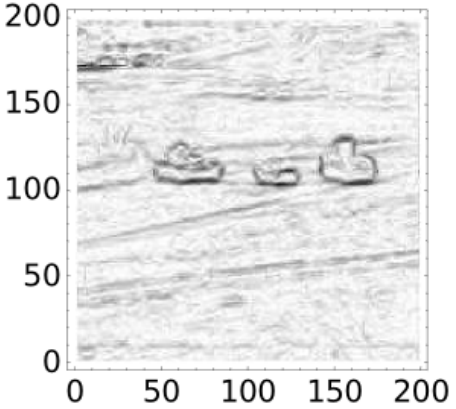
from Arathorn (2004)



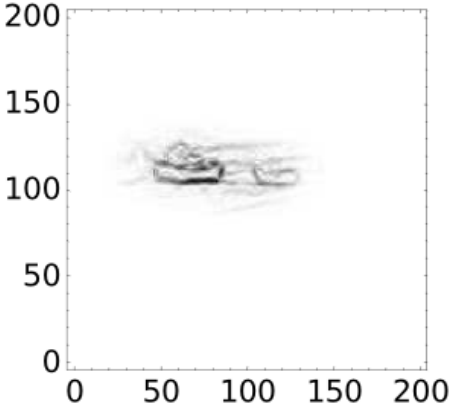
(a) 3D model of M-60 tank

(b) source image

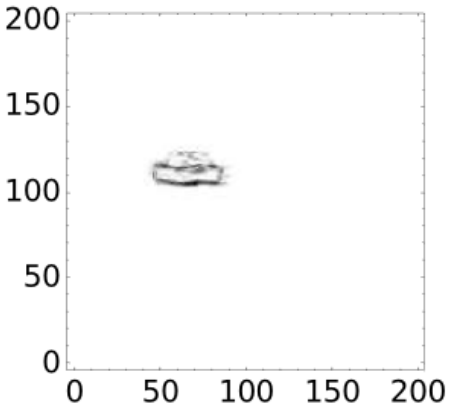
(c) input image - blurred



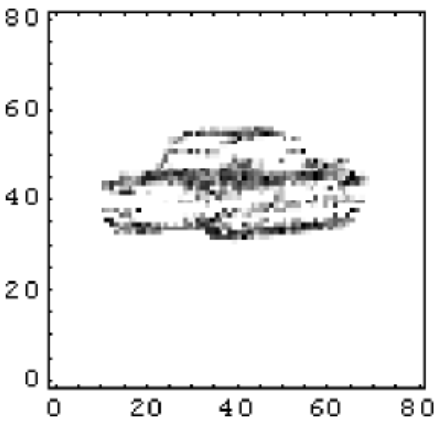
(d) iter 1



(d) iter 3



(e) iter 20



(g) final model pose

Problem

views in memory
(and associated label)

find best-matching view and pose

input

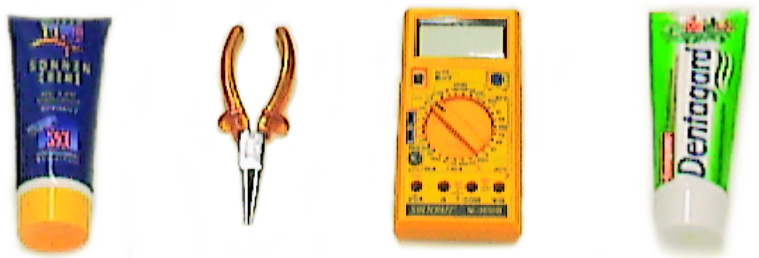
Subproblem I

view in memory

find transformation parameters

input

Subproblem II



views in memory

find best-matching view

input image

Subproblem III

views in memory

find best-matching view and pose
at the same time

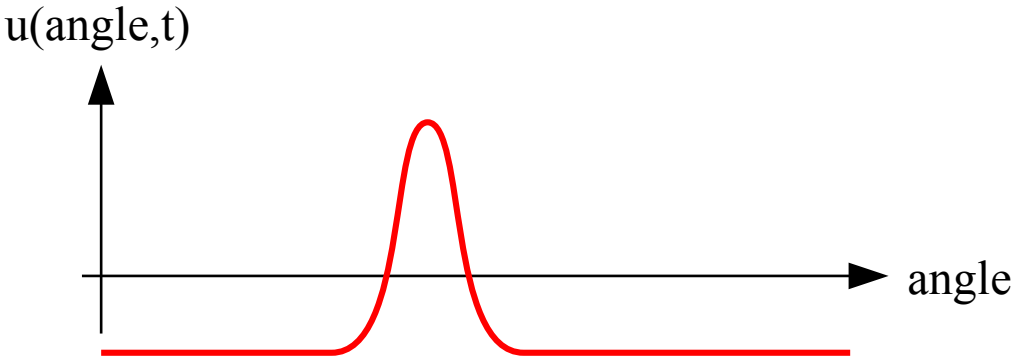
input

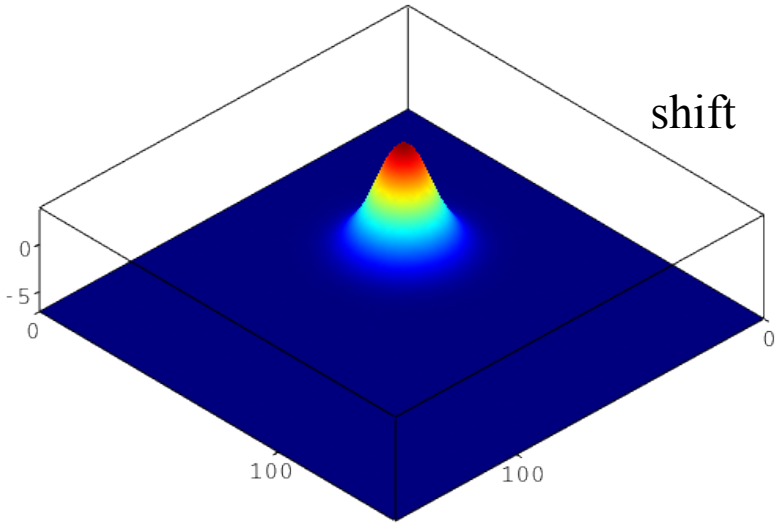
Pose parameter encoding

- How can pose be represented?

  → Neural field

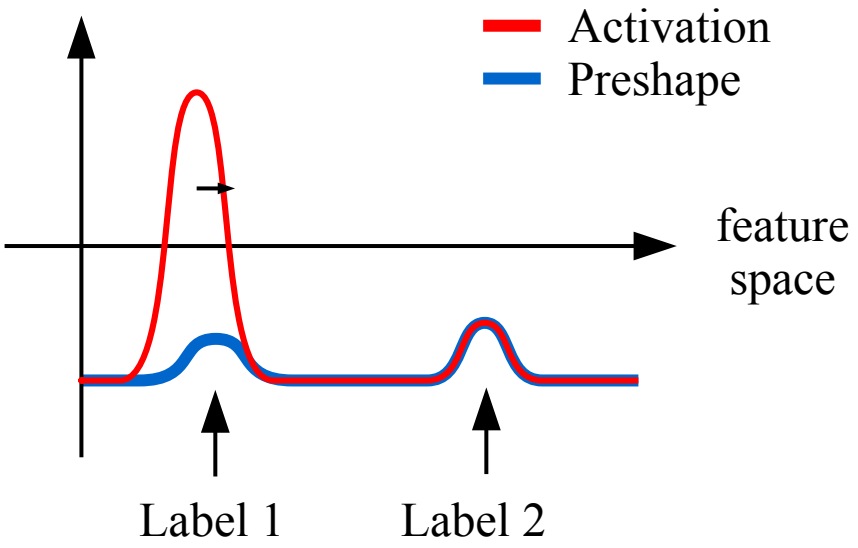
- Example: position as 2d peak, rotation as 1d peak



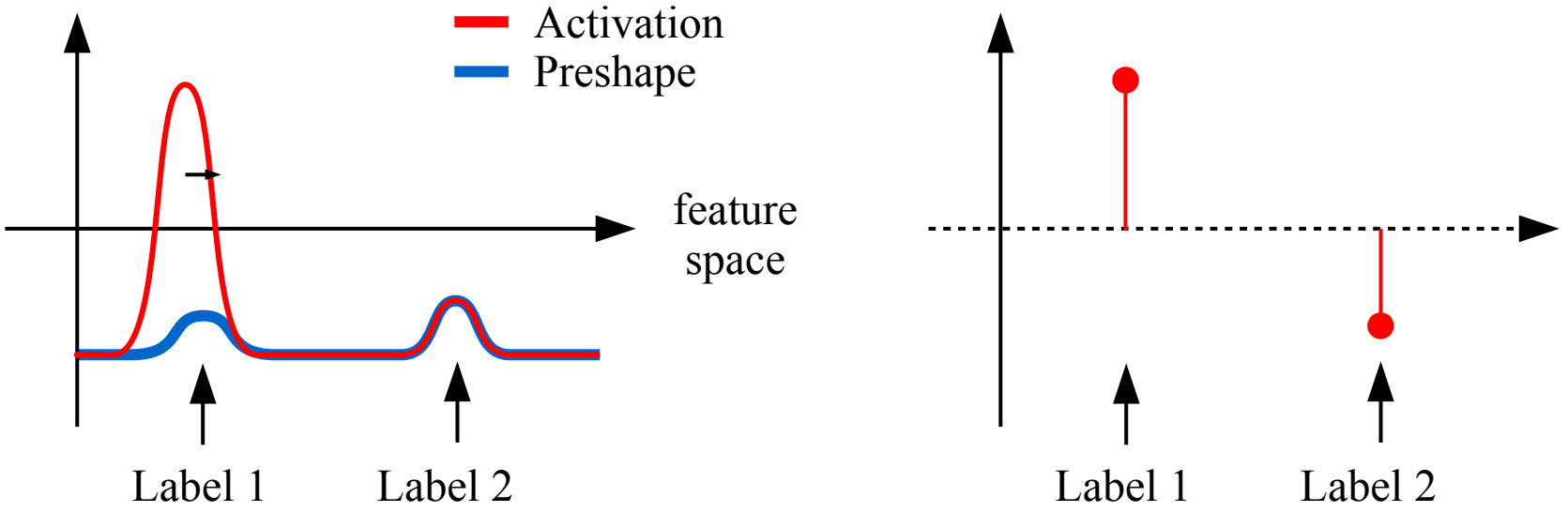


Label encoding

- What is the output of recognition?
  - Categorization decision (binary or graded response)
  - Conflicts with continuous nature of fields
- Categorization in dynamic neural fields (cf. categorical states, DFT core lecture 2)



Label fields



- Discrete nodes
- Only global interaction / no metric
- Supra-threshold activation = detection of label

The principle: 1D shift

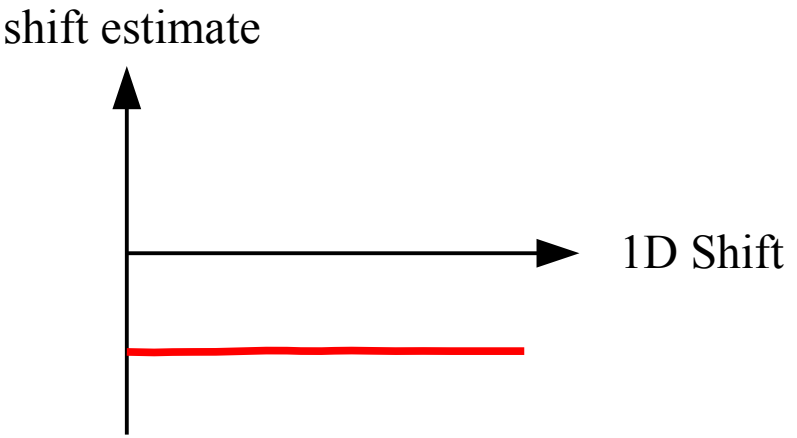
Estimating (horizontal) shift

view

input

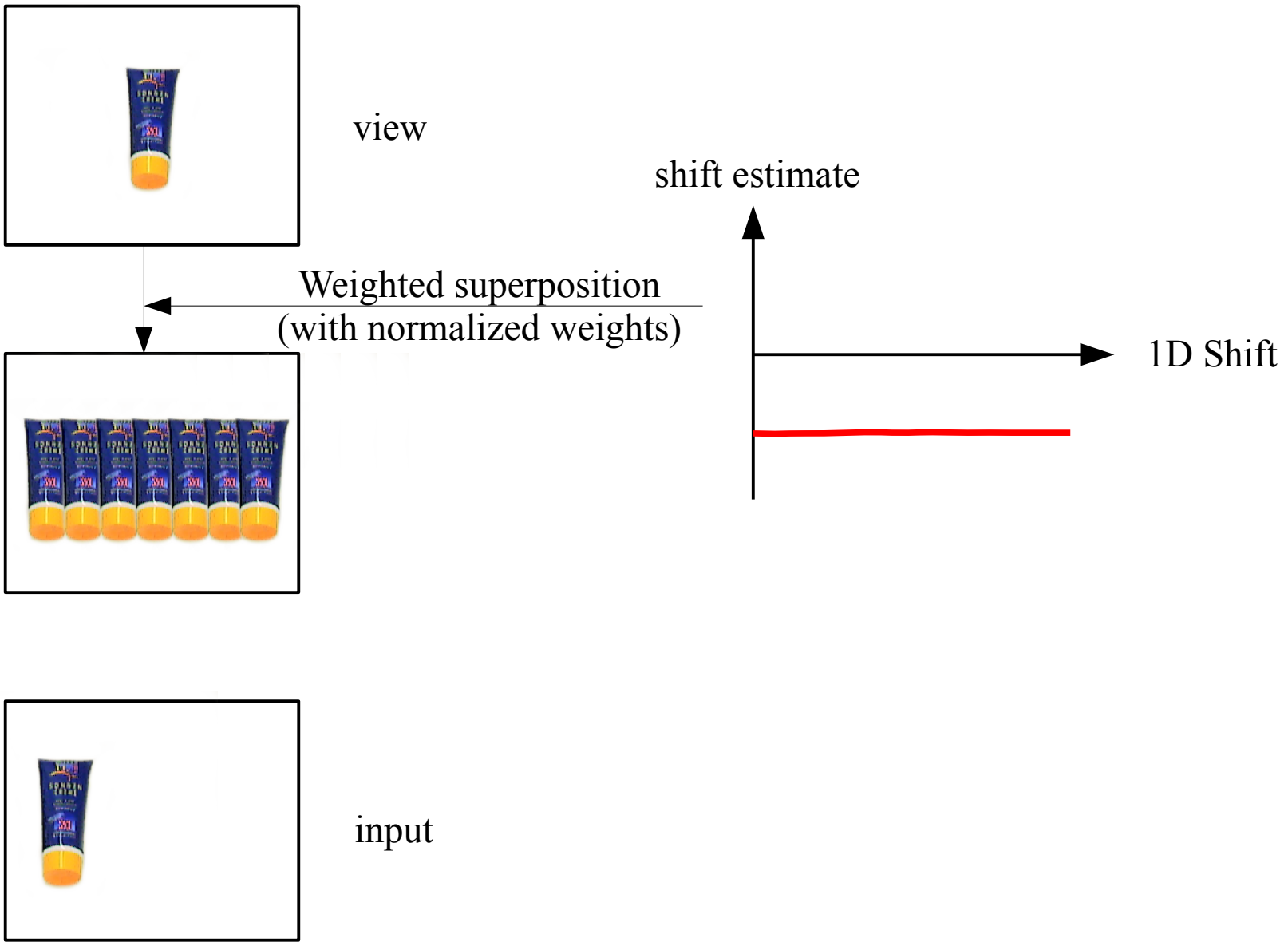
Estimating (horizontal) shift

view

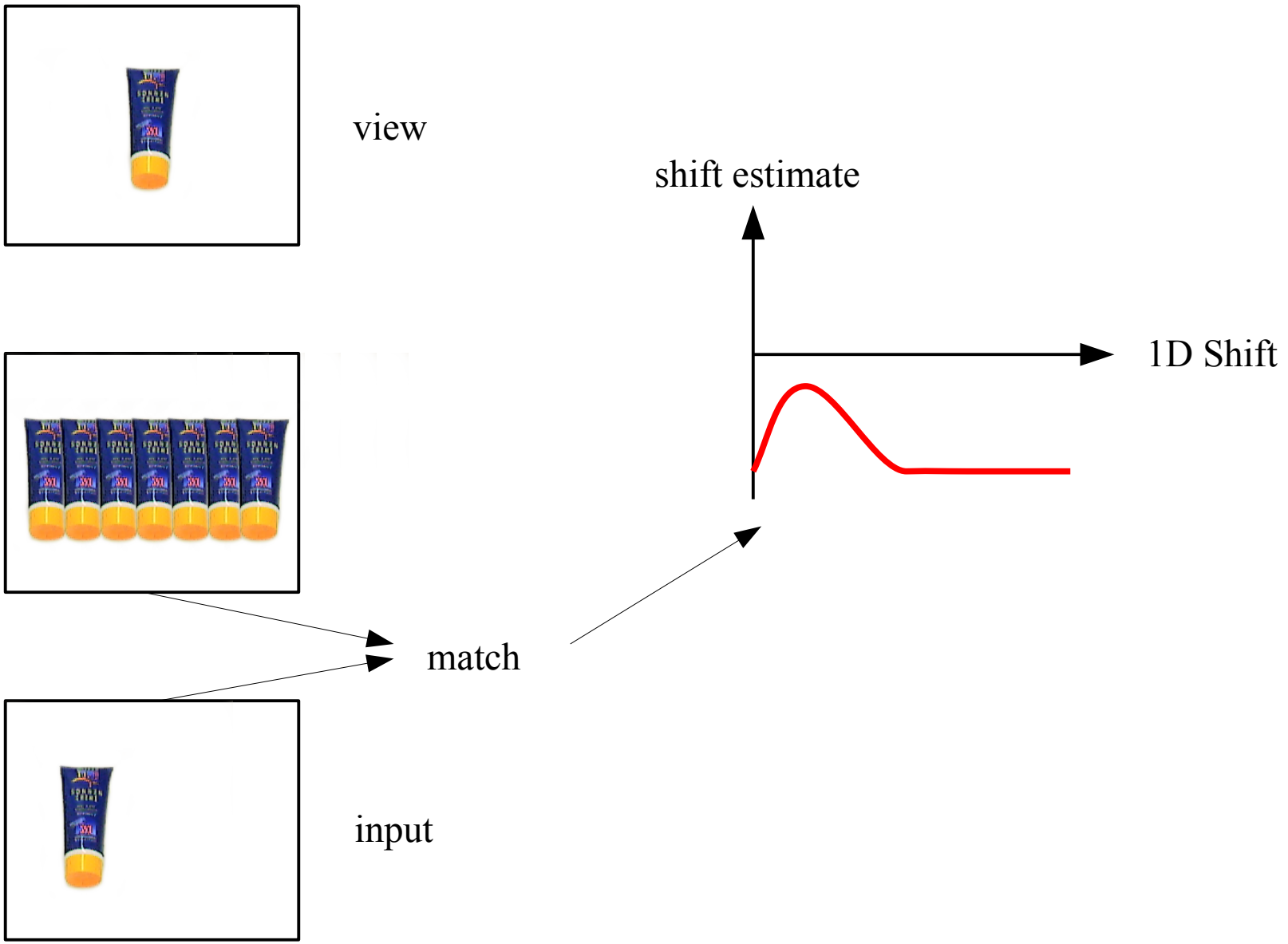


input

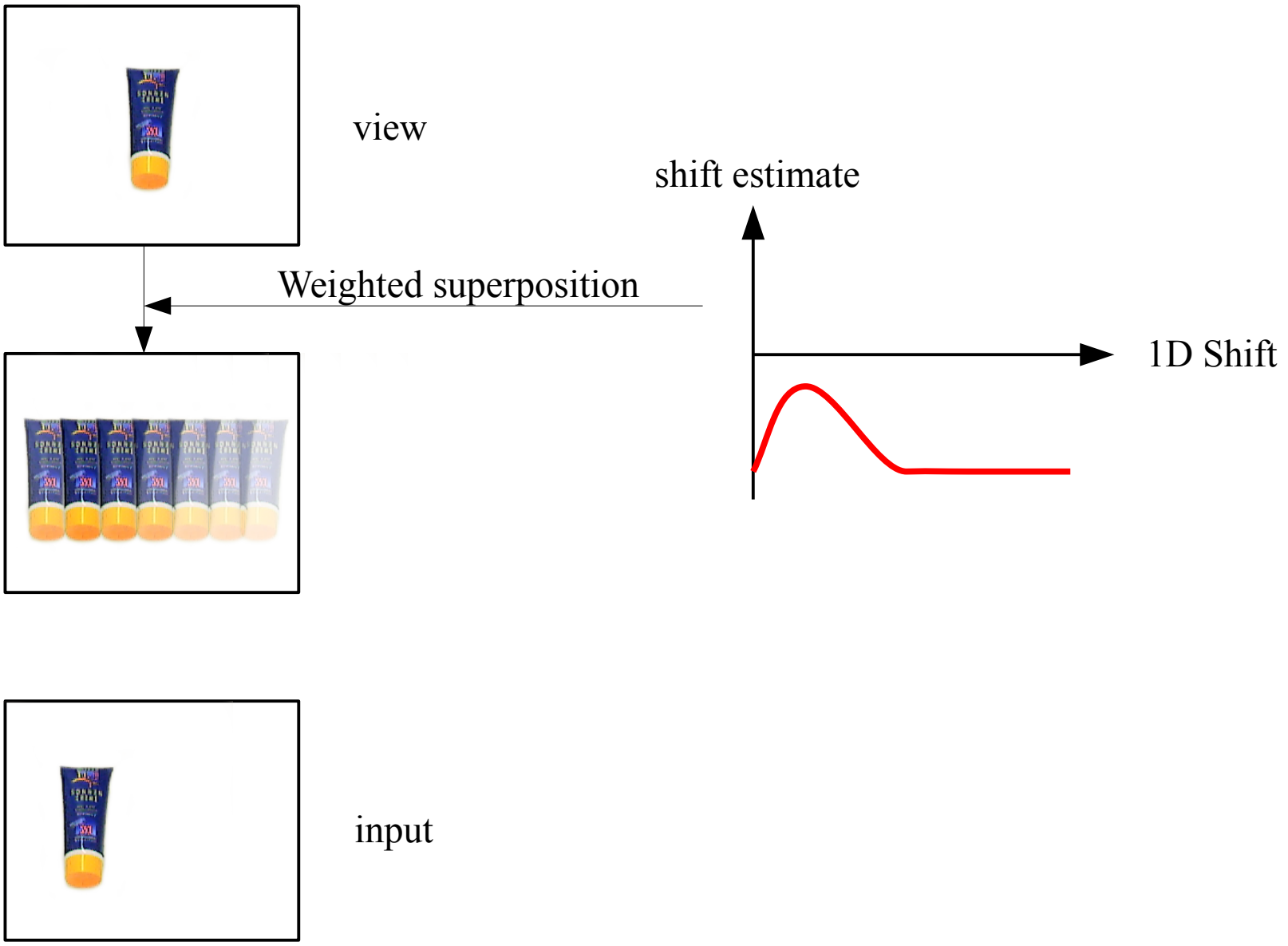
Estimating (horizontal) shift



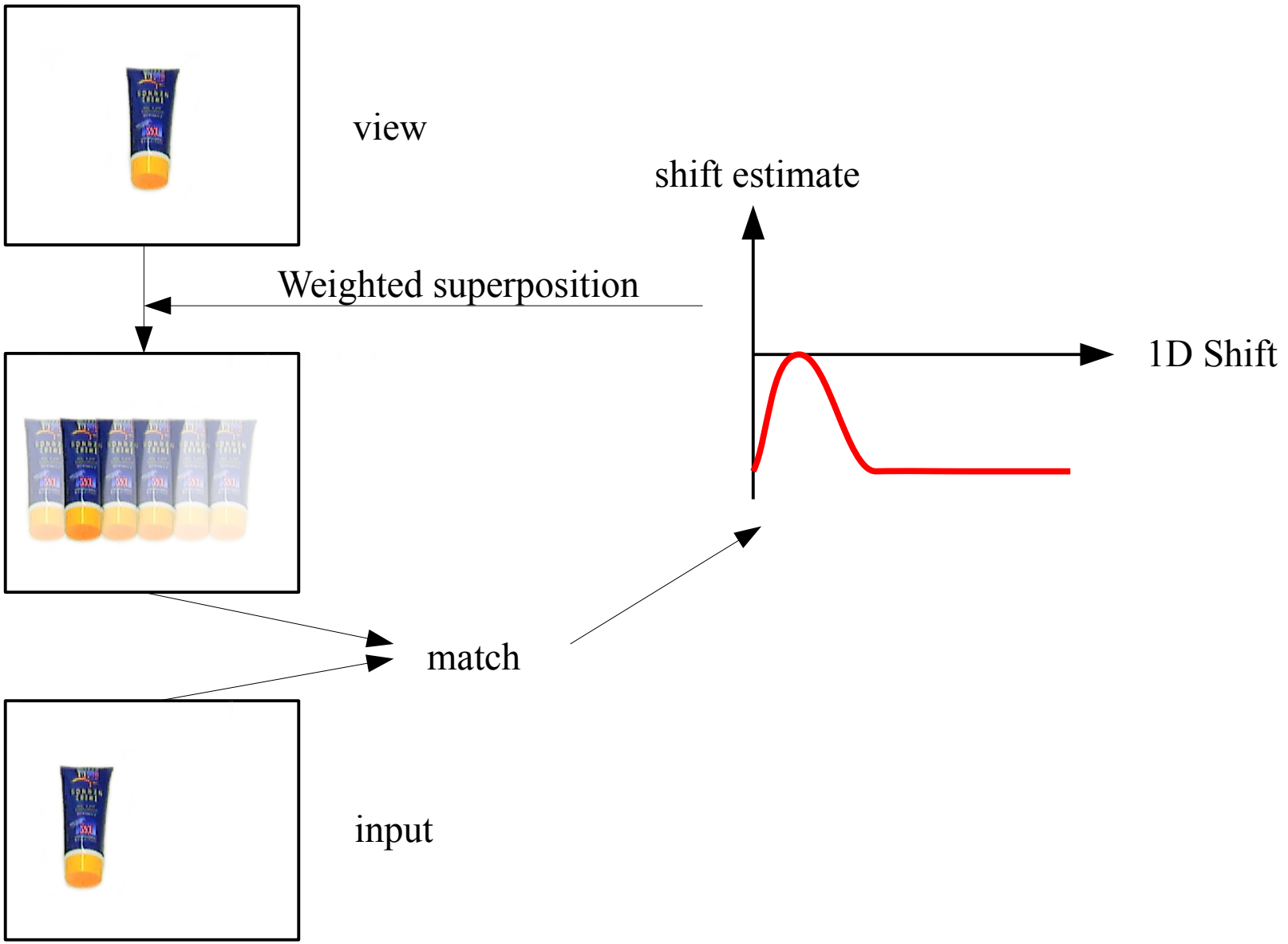
Estimating (horizontal) shift



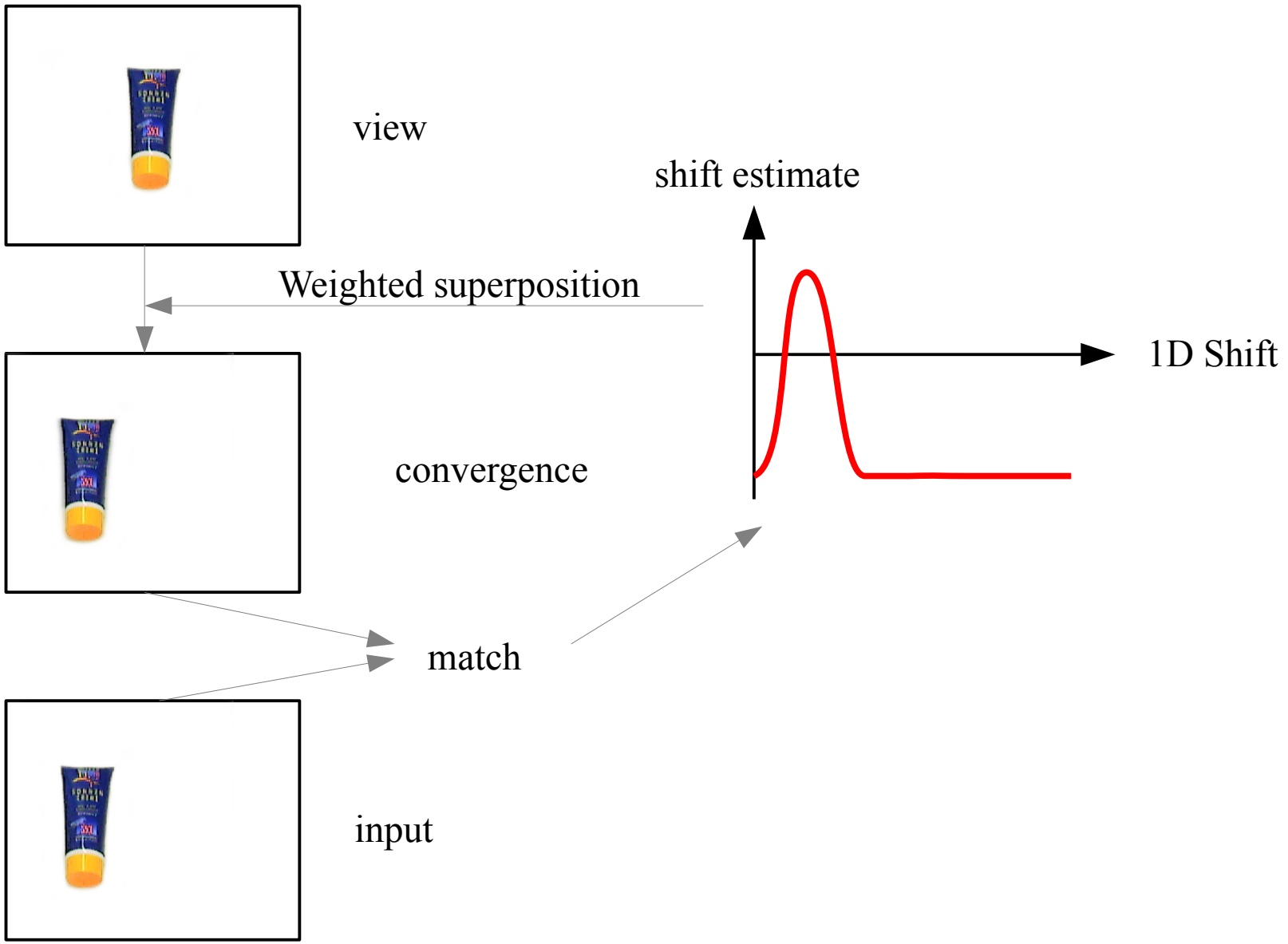
Estimating (horizontal) shift



Estimating (horizontal) shift



Estimating (horizontal) shift



The principle: matching views

Recognizing an object

Known objects

Input image

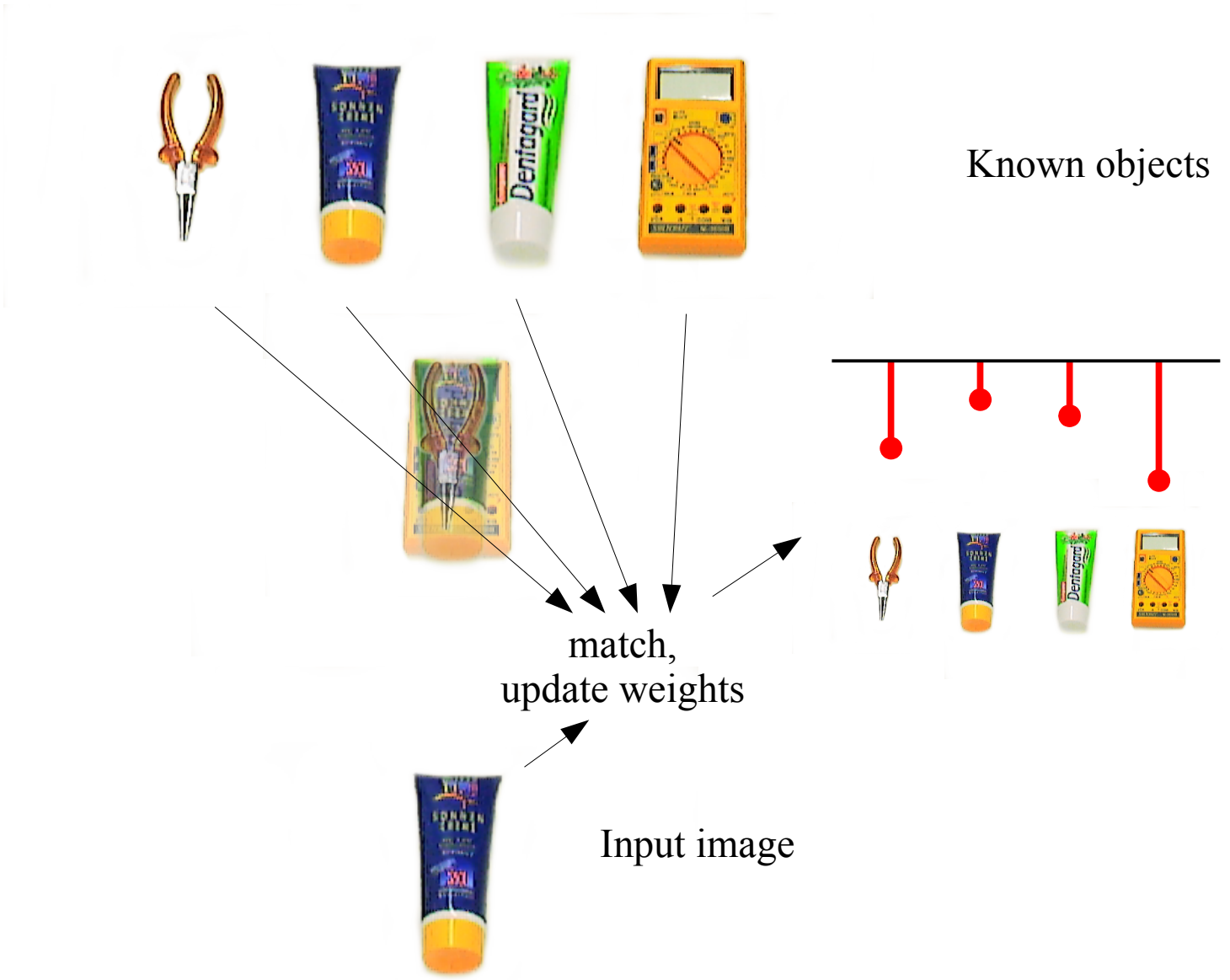
Recognizing an object

Known objects

weighted
superposition

Input image

Recognizing an object



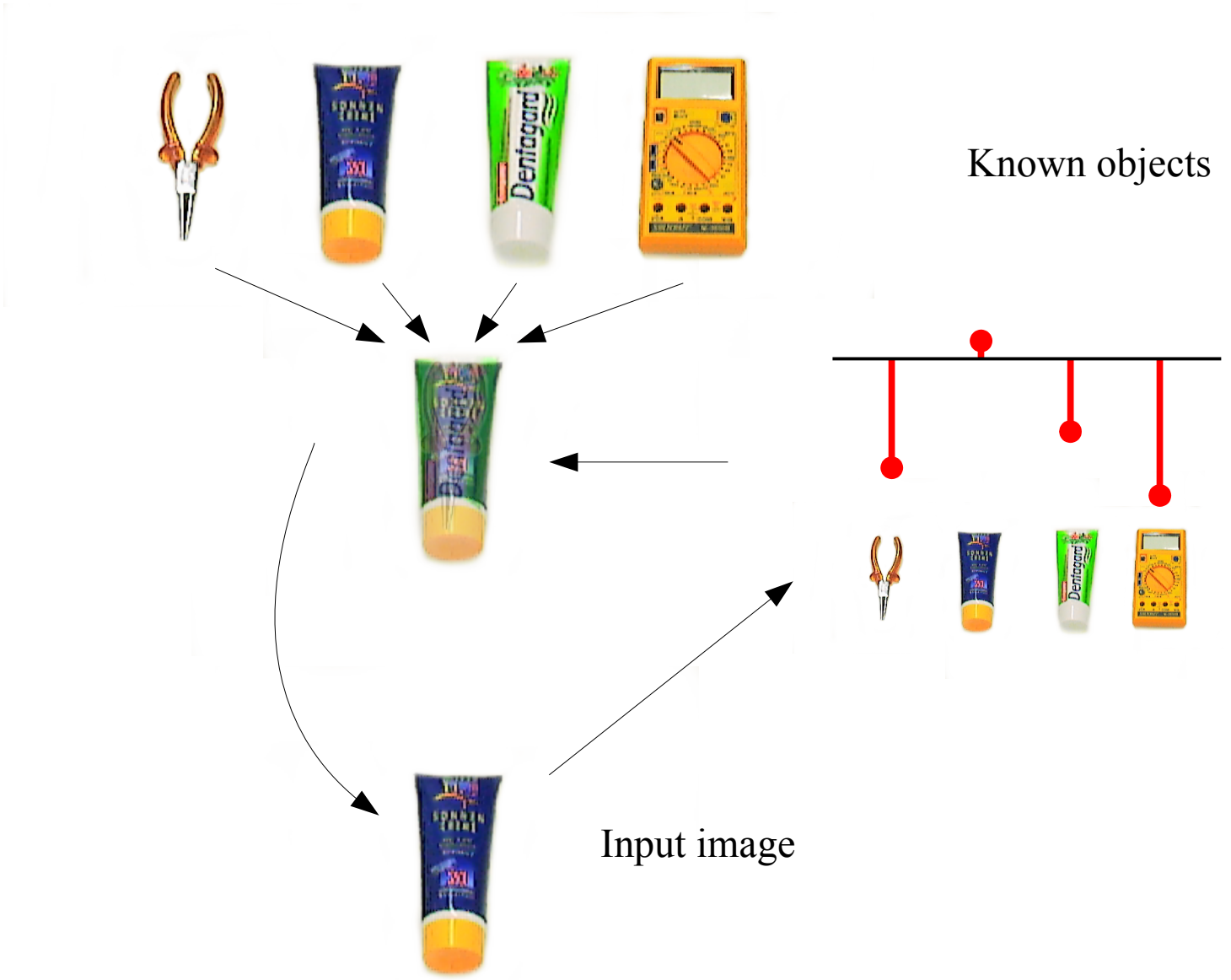
Recognizing an object

Known objects

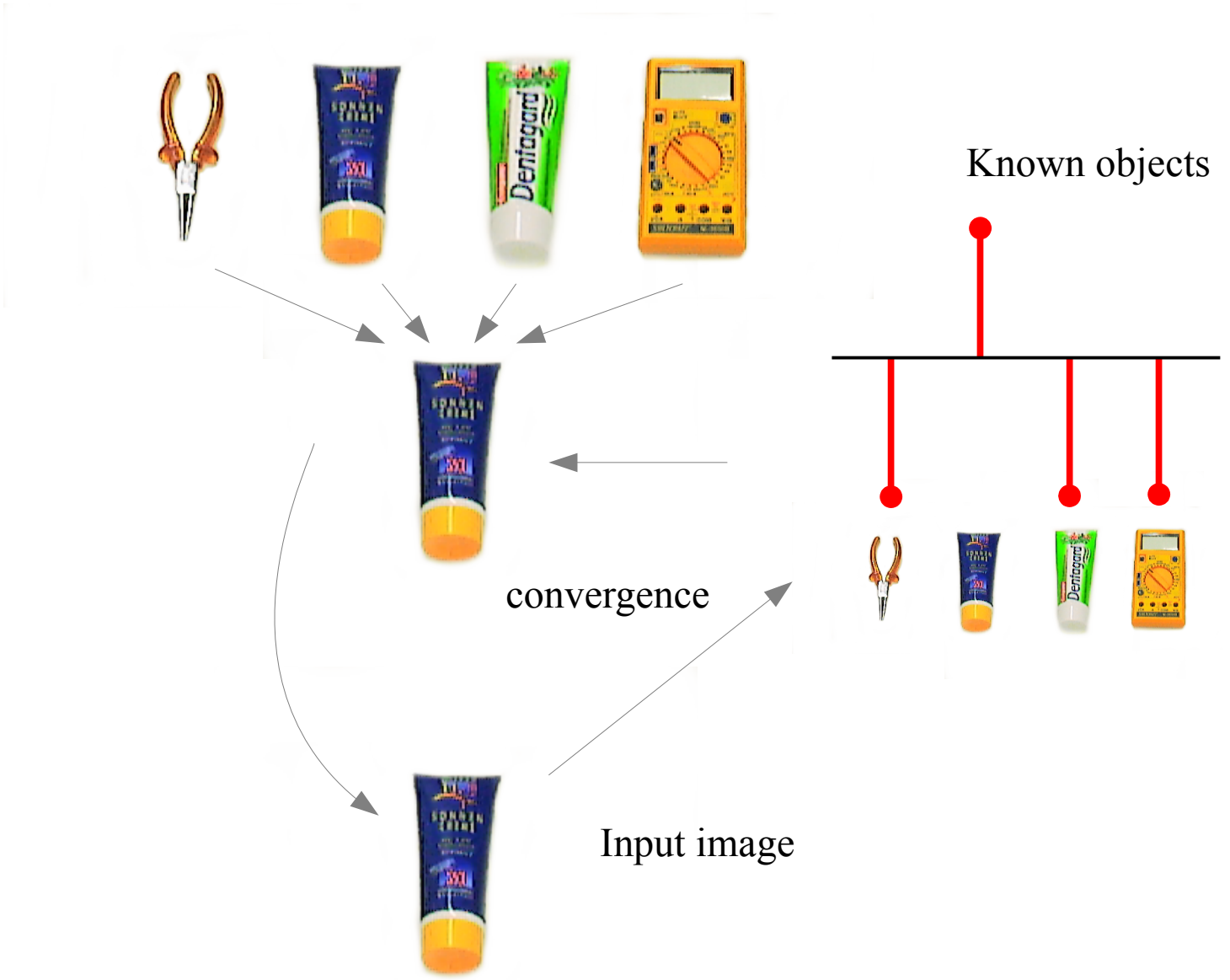
update
superposition

Input image

Recognizing an object



Recognizing an object

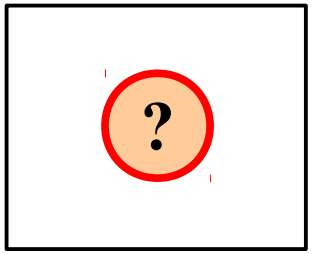


Combining the recognition and pose matching

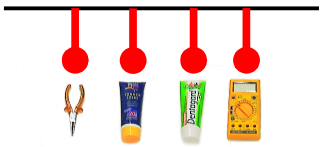
Recognition and pose estimation

View matching

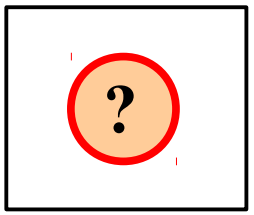
Shift matching

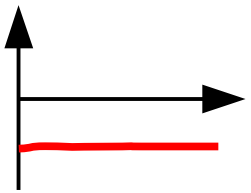


view to match



transformed input



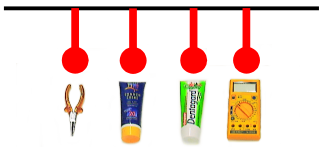


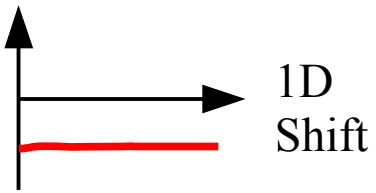
1D Shift

Input image

Recognition and pose estimation

View matching

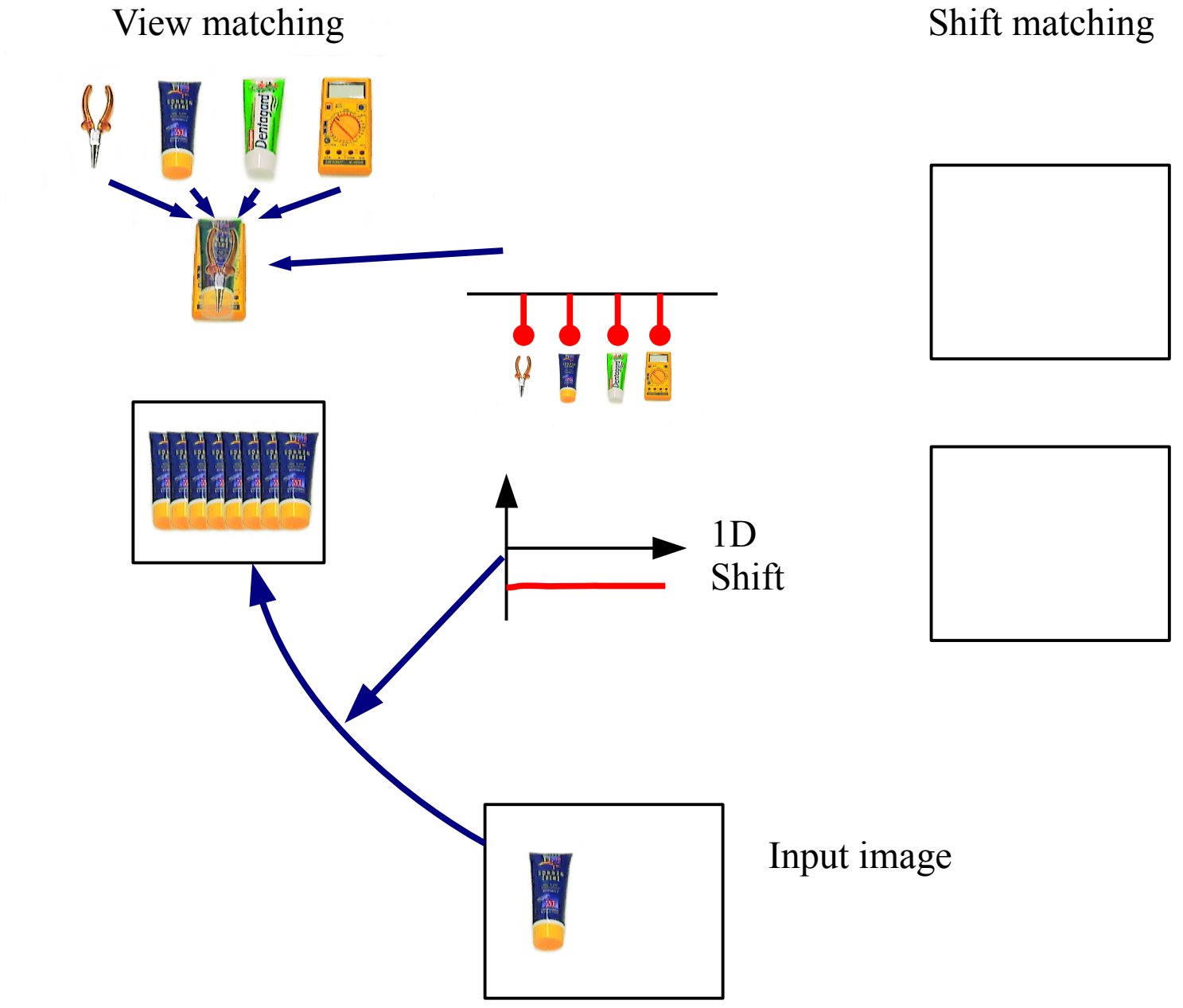




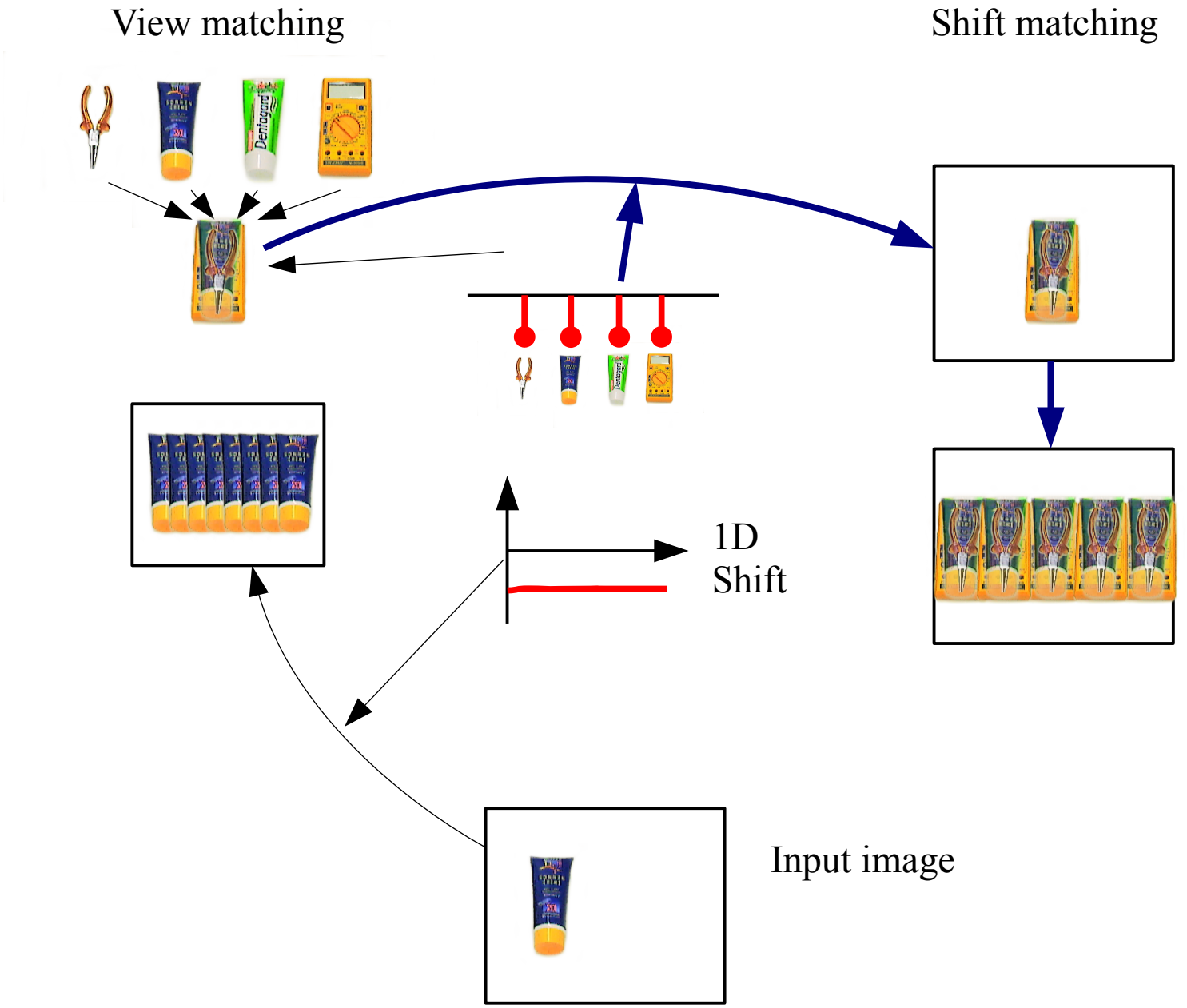
Shift matching

Input image

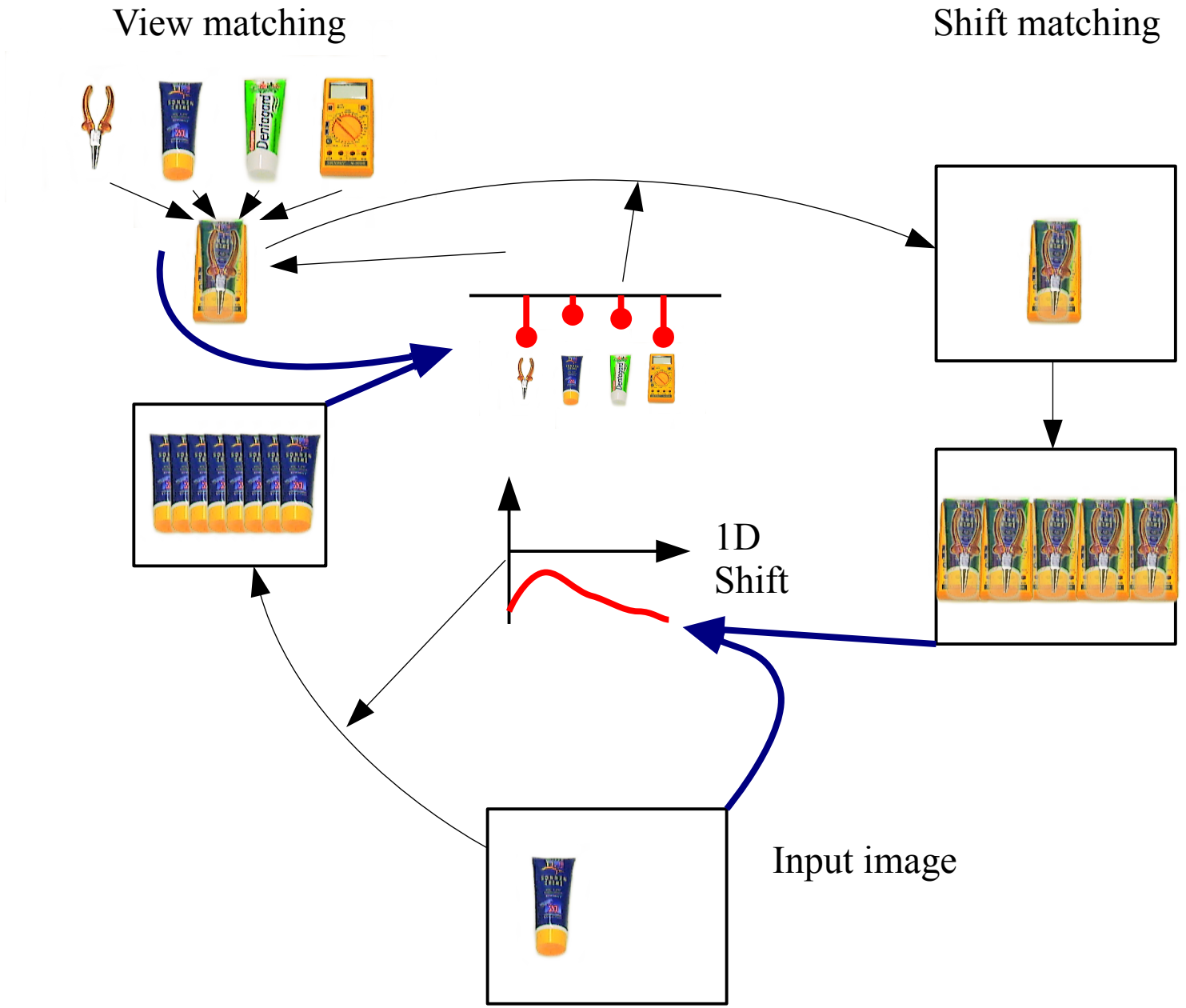
Recognition and pose estimation



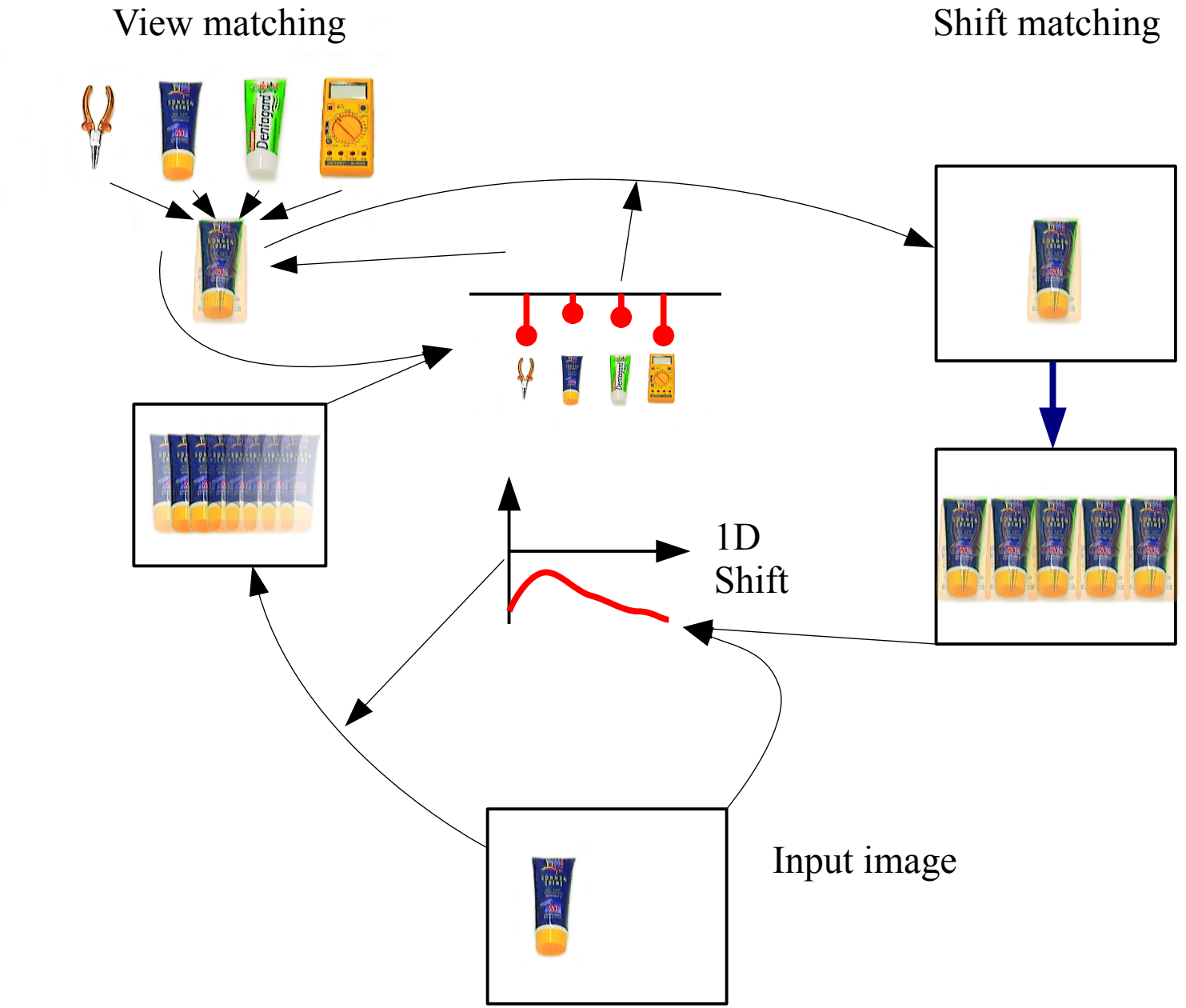
Recognition and pose estimation



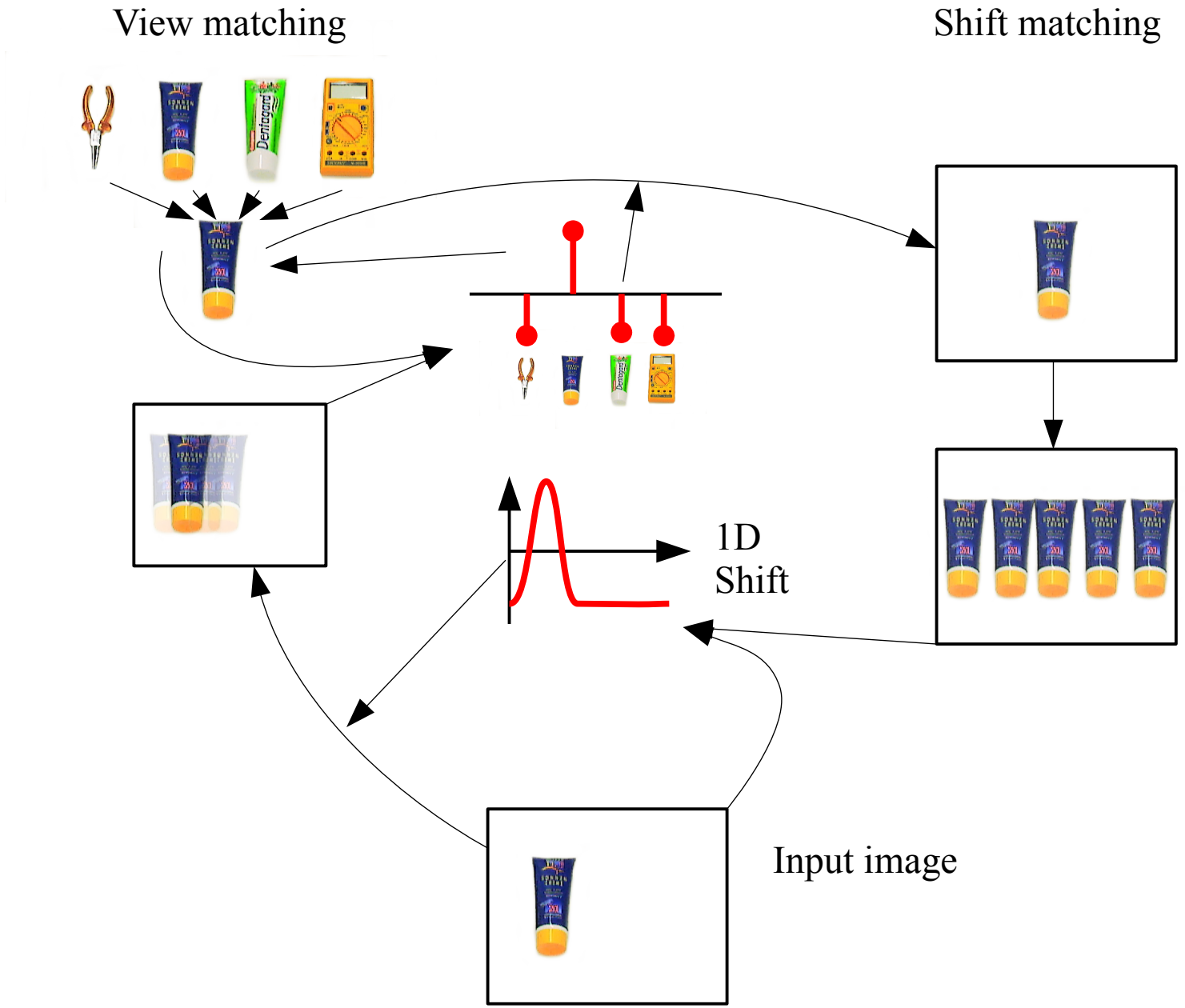
Recognition and pose estimation



Recognition and pose estimation

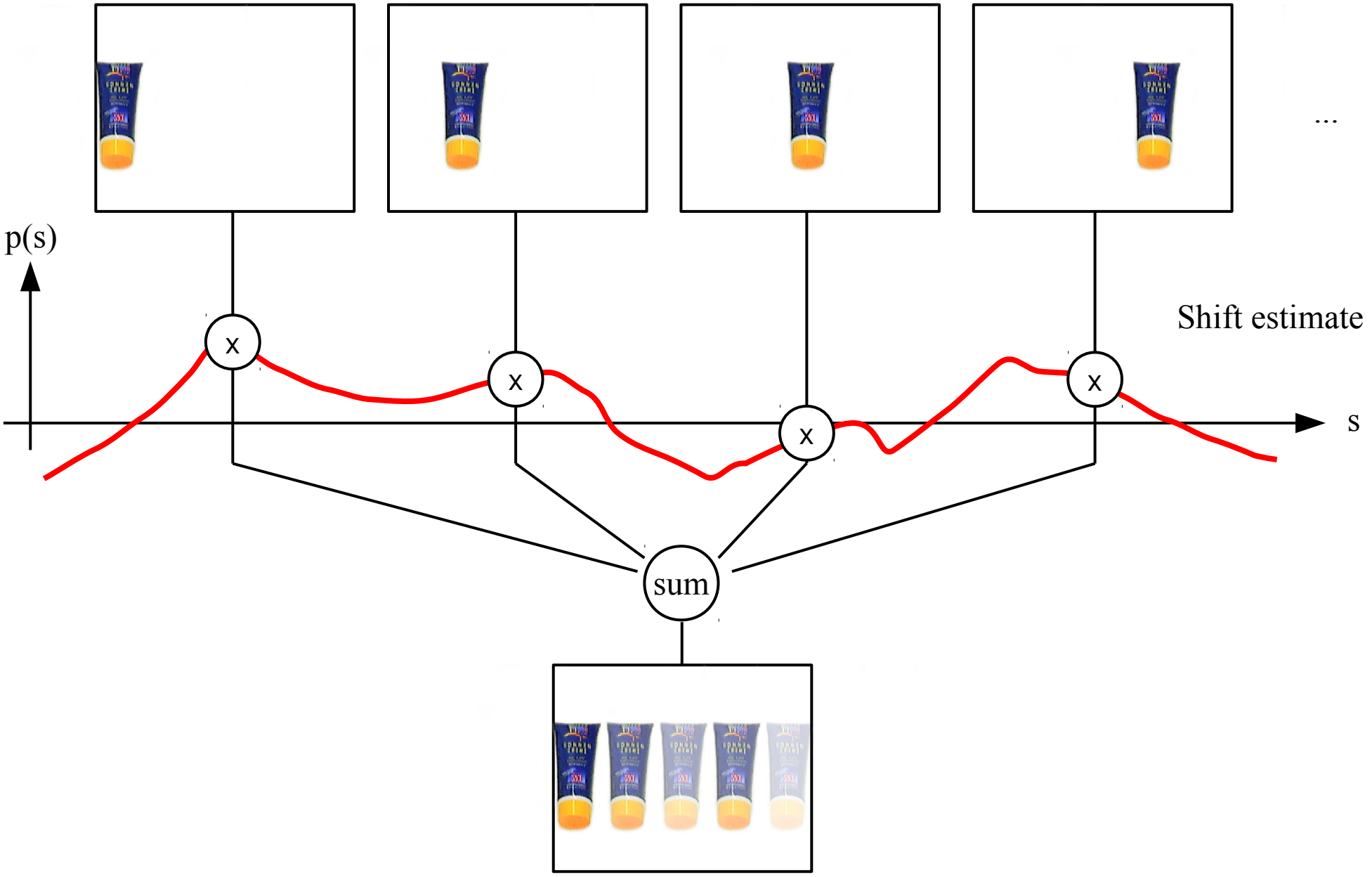


Recognition and pose estimation

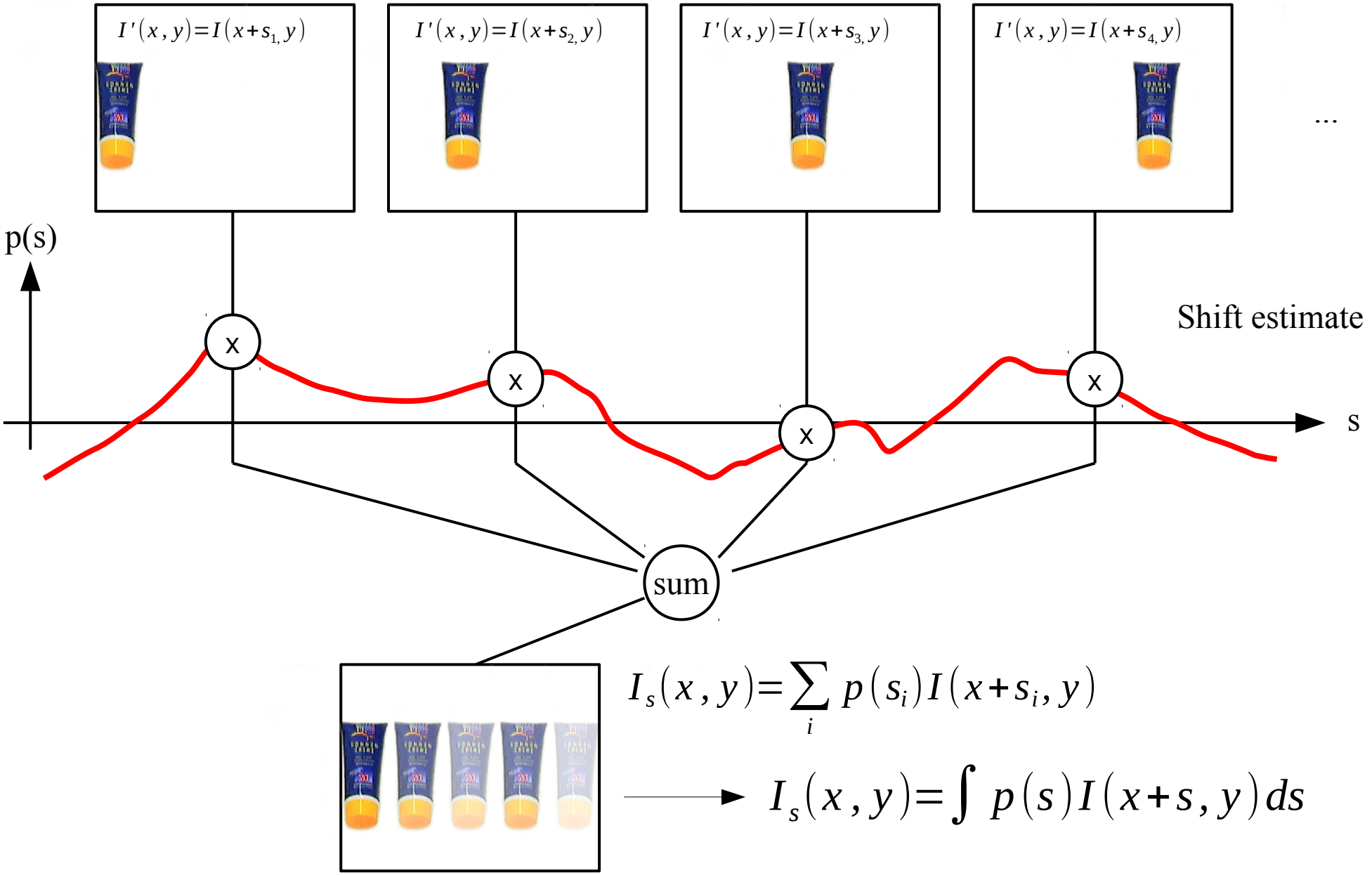


Implementation

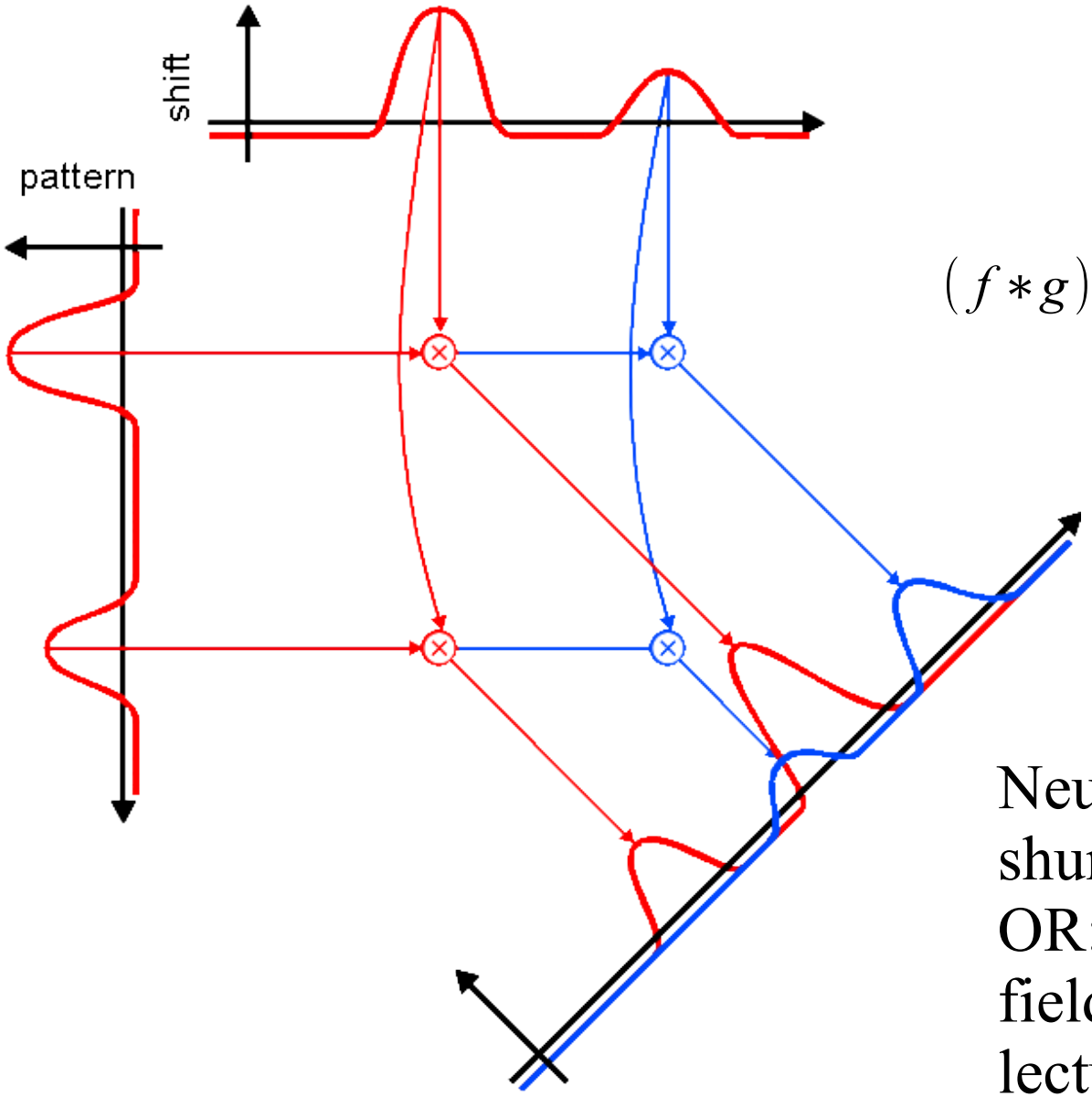
Transformations



Transformations



Transformations

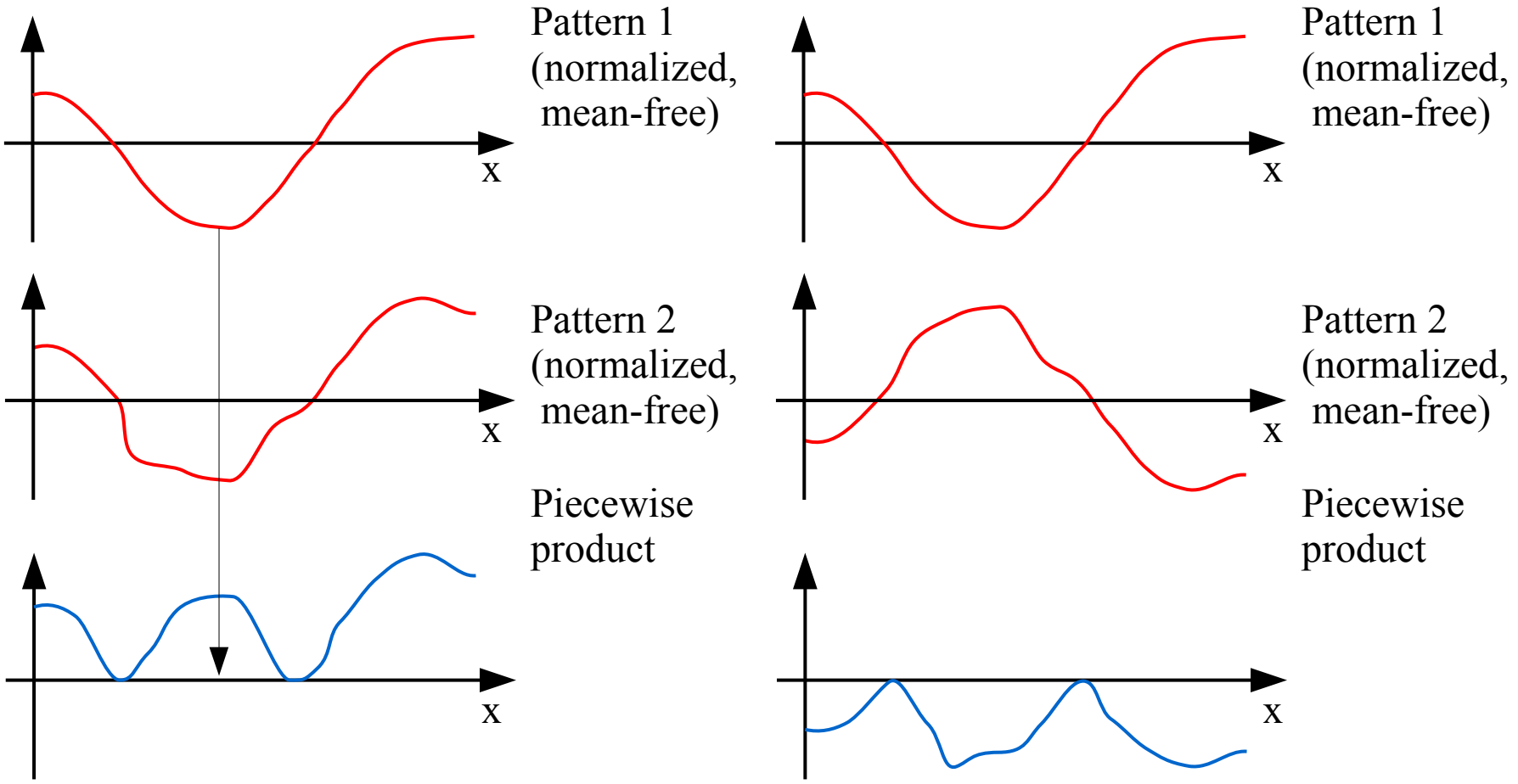


$$(f * g)(x) = \int f(y)\, g(x - x')\, dx'$$

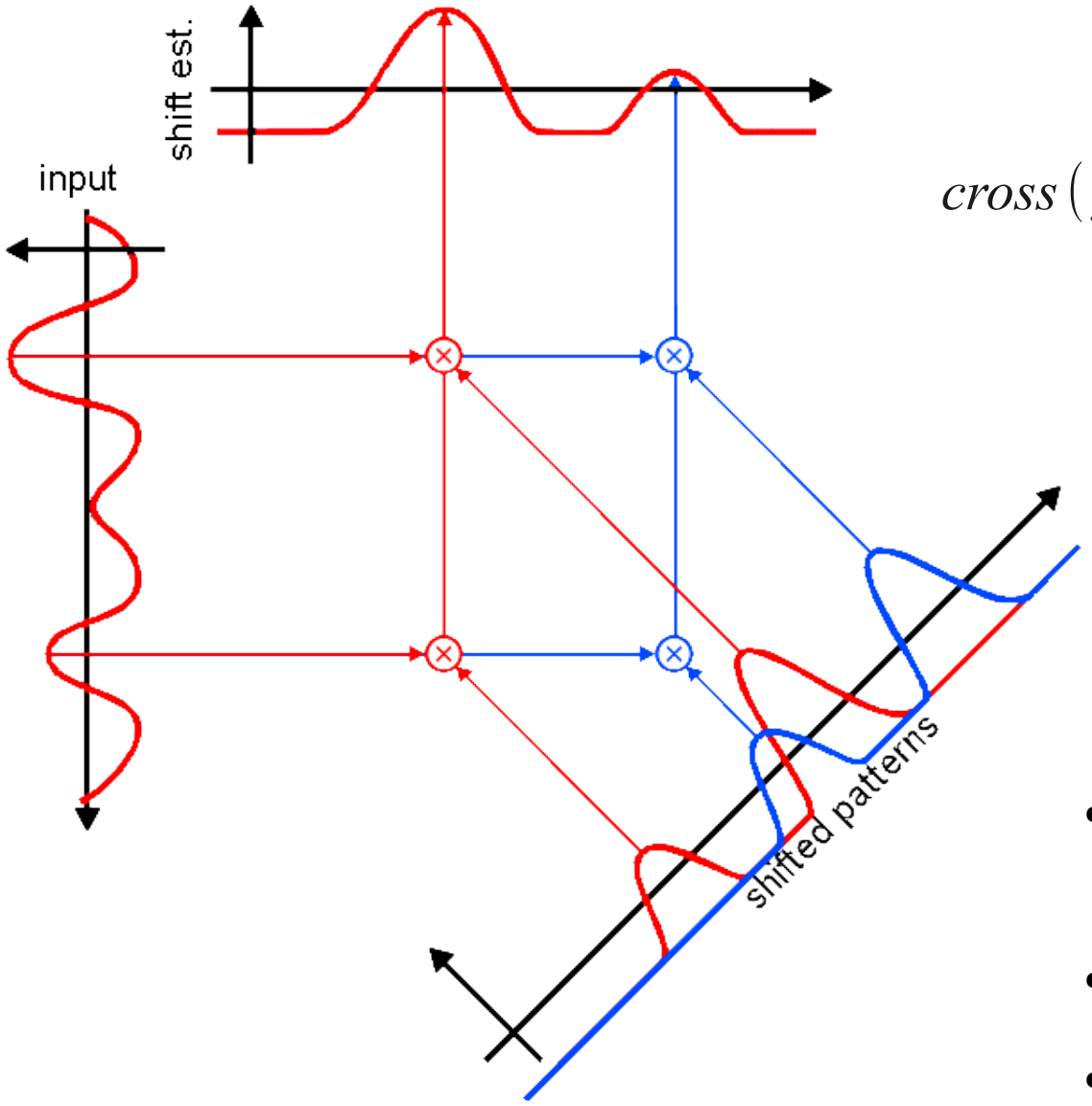
Neural implementation: shunting synapses
OR: transformation fields (cf. DFT core lecture 2)

Matching views

$$match(p_1, p_2) = \int p_1(x) \cdot p_2(x)\, dx$$



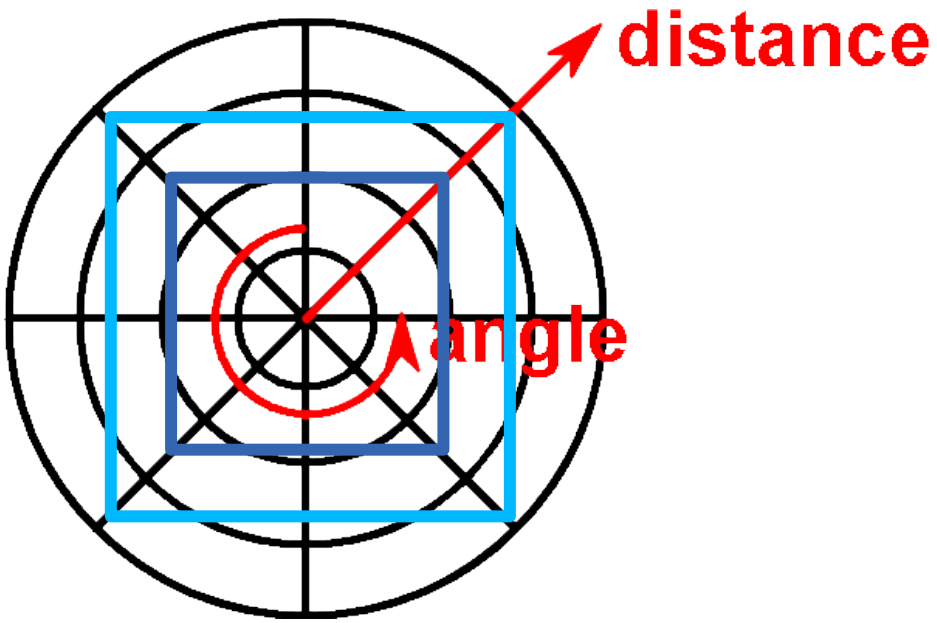
Matching poses



$$cross(f, g, x) = \int \overline{f}(y)\, g(x+y)\, dy$$

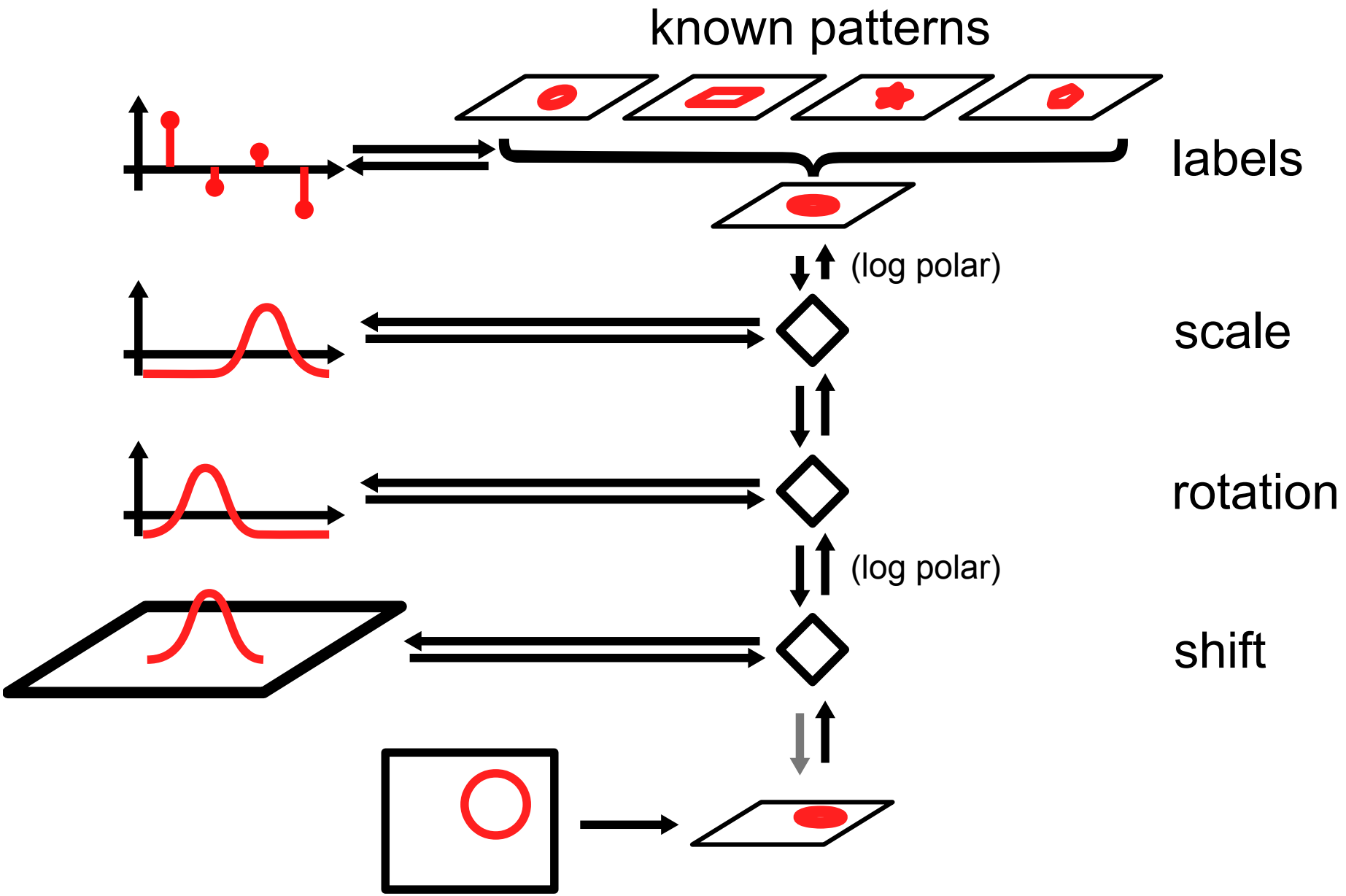
- Similar to transformation, different direction
- Normalized, mean-free
- Requires shunting synapses

Other transformations



- Shift in log-polar space is uniform scaling and rotation
- Log-polar is neurally plausible (retinal space)

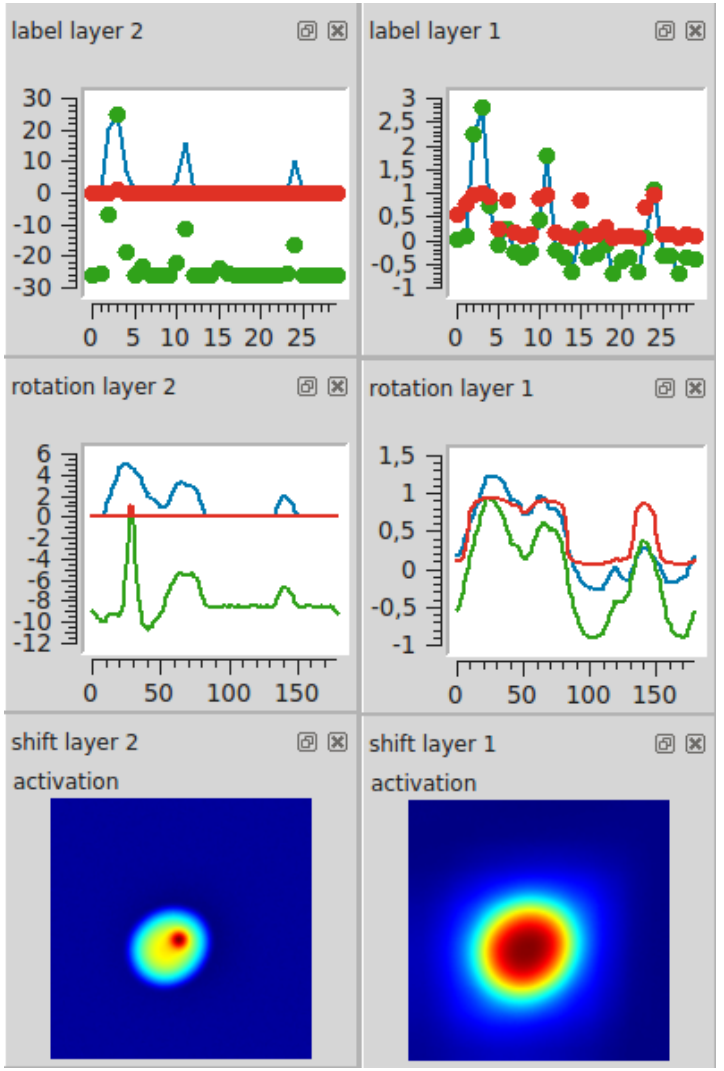
Cascading transformations



Pose parameter encoding, continued

- Two layers
  - First layer detects
  - Second one selects
- Pose estimate

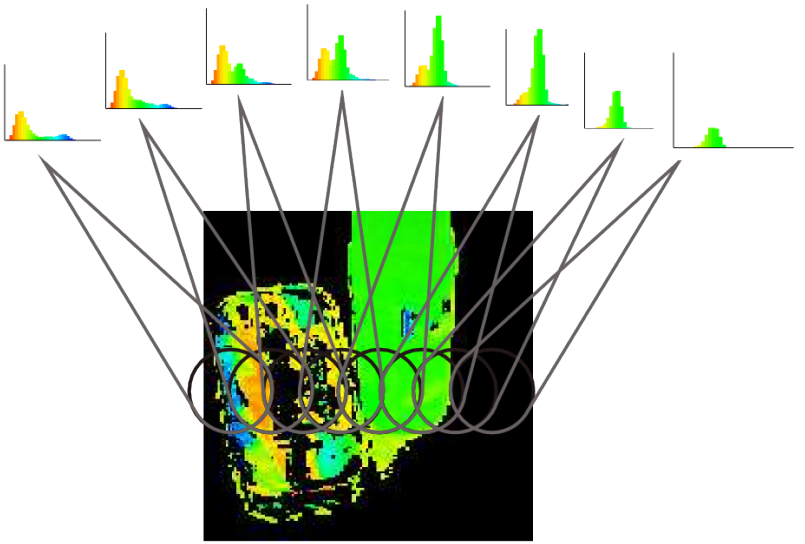
  = layer 2 if above threshold, layer 1 otherwise



Feature channels

- Full system has several feature channels
  - Spatial pattern (shape)
  - Localized histograms
    - Color (hue)
    - Edge orientations

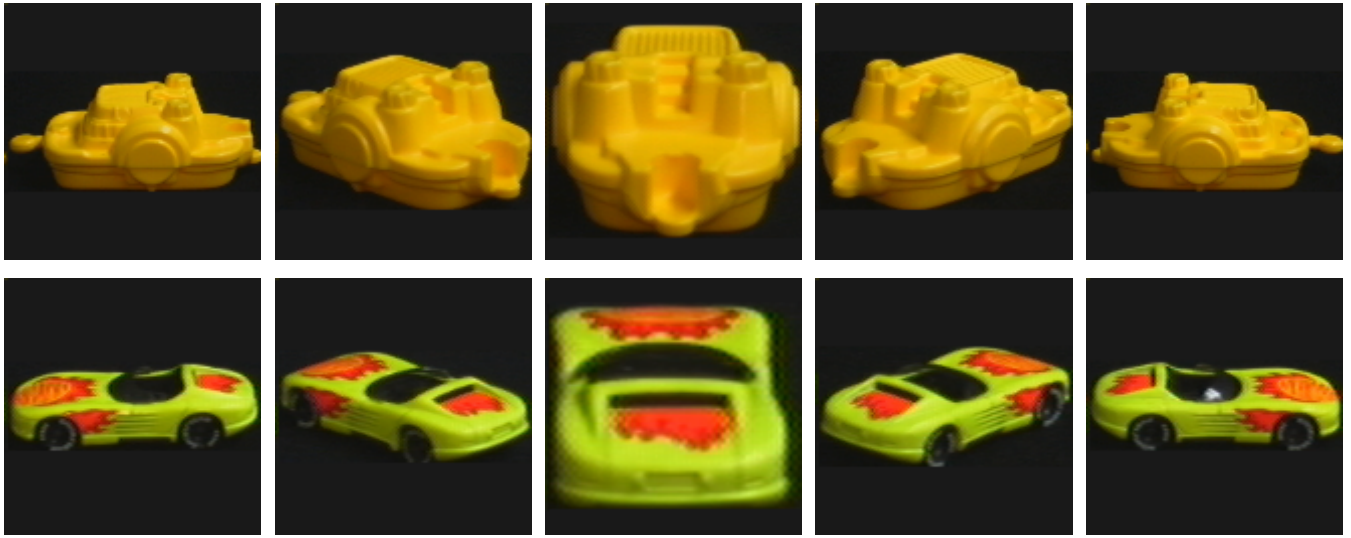
Localized Histograms



from Faubel, Schöner (2009)

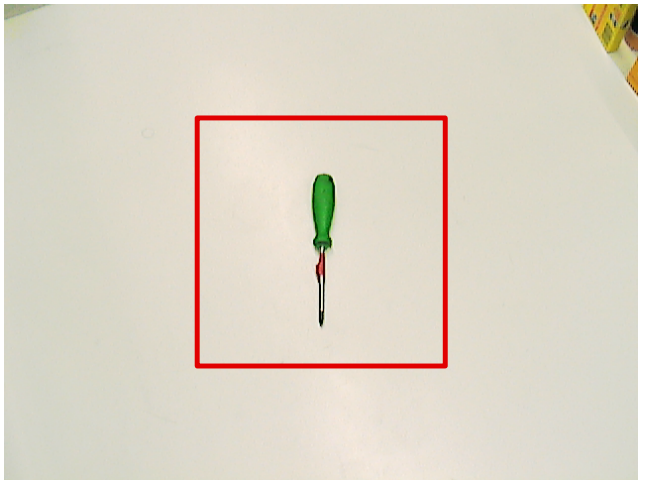
- Shape alone (as presented before) is not very powerful for recognition
- Additional feature channels: localized histograms (color, edge orientations)
- All channels provide information about pose, object
- Scale cannot be estimated

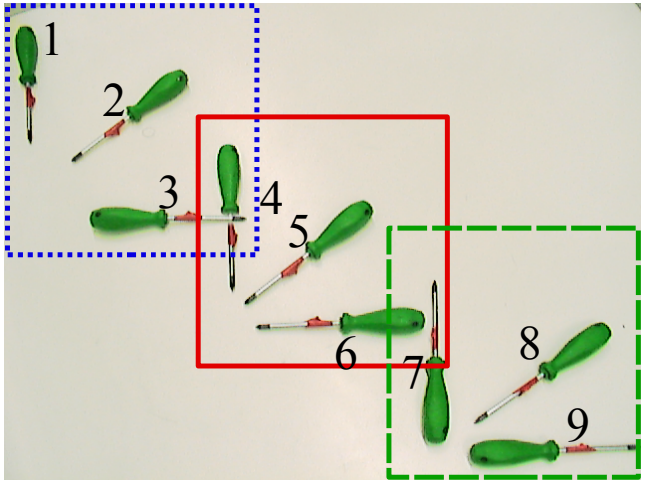
Results



- Recognition rates on COIL (with shape + localized histograms):
  - 85% with one training view
  - 94% with four training views
  - See Faubel, Schöner (2009)

Results II

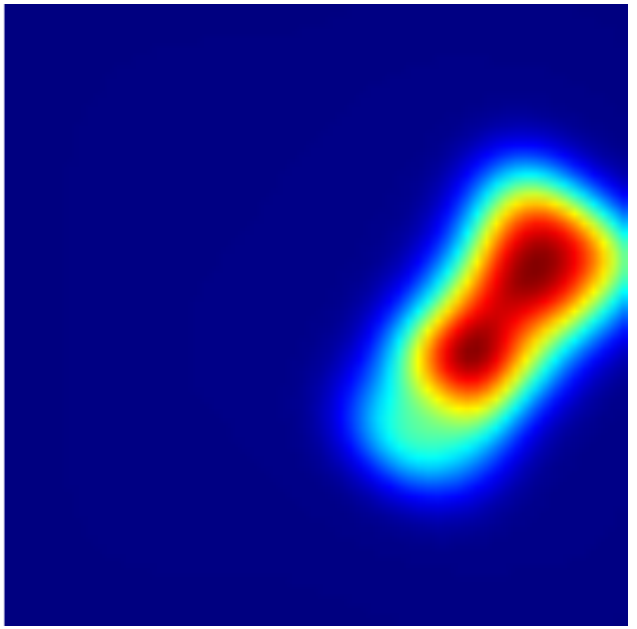




- Recognition rates on tabletop dataset (with shape + localized histograms):
  - 90% with one training view/single object
  - See also Faubel, Schöner (2009)

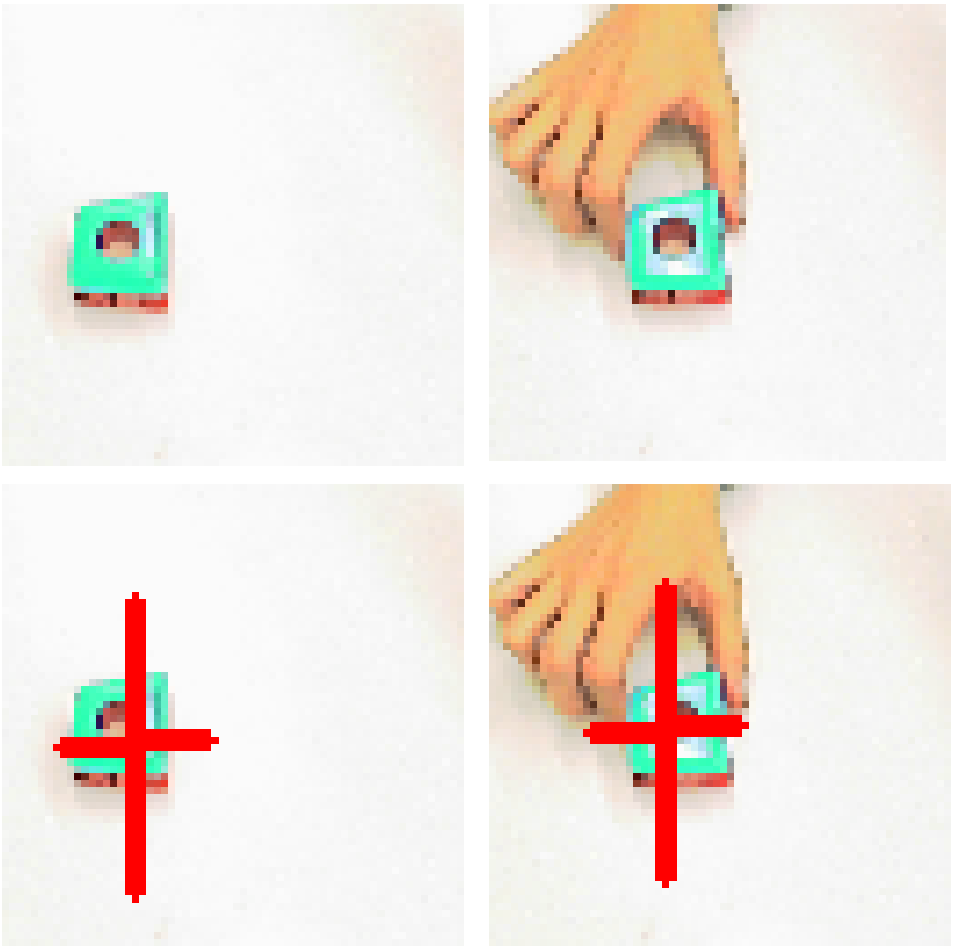
Video

Masking: what is it good for?

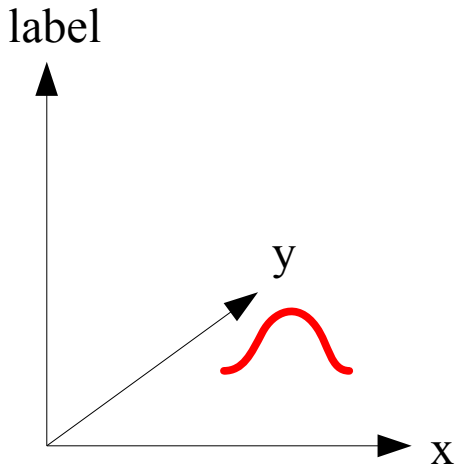
- Masking input allows to focus on single object

Masking: what (else) is it good for?

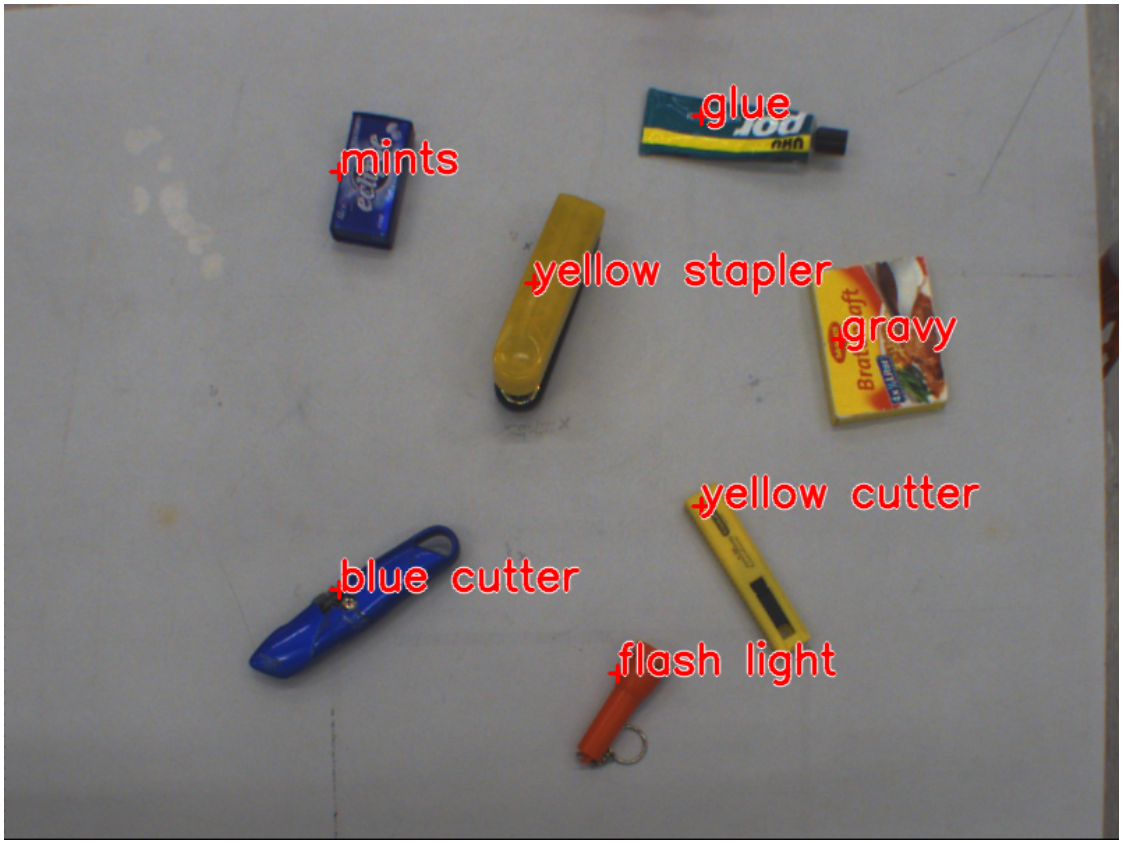


(from Faubel, Schöner 2009)

Outlook: attentional control



scene representation
+ object recognition



Thank you for your attention!

Questions?

For more: Faubel, C., & Schöner, G. (2009). *A neuro-dynamic architecture for one shot learning of objects that uses both bottom-up recognition and top-down prediction*. In Proc. of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009. IEEE Press.
(also contains references to other work mentioned here)