Computational Vision
U. Minn. Psy 5036
Daniel Kersten
Lecture 25: Object recognition, background

Initialize

```
Off[General::spell];
```

```
<< Graphics`Graphics3D`
<< Graphics`MultipleListPlot`
<< "Graphics`Polyhedra`"
```

Outline

Last time

Object recognition overview

Today

Object recognition: compensating for viewpoint changes

Recognition, background variation, segmentation & learning objects

Variation over view: review

From the previous lecture...

Geometric variation: 3D scene-based modeling

Homogeneous coordinates

```
XRotationMatrix[theta_] :=
  {{1, 0, 0, 0}, {0, Cos[theta], -Sin[theta], 0},
   {0, Sin[theta], Cos[theta], 0}, {0, 0, 0, 1}};
YRotationMatrix[theta_] :=
  {{Cos[theta], 0, Sin[theta], 0}, {0, 1, 0, 0},
   {-Sin[theta], 0, Cos[theta], 0}, {0, 0, 0, 1}};
ZRotationMatrix[theta_] :=
  {{Cos[theta], -Sin[theta], 0, 0}, {Sin[theta], Cos[theta], 0, 0},
   {0, 0, 1, 0}, {0, 0, 0, 1}};
ScaleMatrix[sx_, sy_, sz_] :=
  {{sx, 0, 0, 0}, {0, sy, 0, 0}, {0, 0, sz, 0}, {0, 0, 0, 1}};
(*TranslateMatrix[x_,y_,z_]:=
   {{1,0,0,x},{0,1,0,y},{0,0,1,z},{0,0,0,1}};*)
TranslateMatrix[x_, y_, z_] :=
  {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {x, y, z, 1}};
ThreeDToHomogeneous[vec_] := Append[vec, 1];
HomogeneousToThreeD[vec_] := Drop[vec/vec[[4]], -1];
ZProjectMatrix[focal_] :=
  {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, N[1/focal], 0}};
ZOrthographic[vec_] := Take[vec, 2];
```

Example: transforming, projecting a 3D object

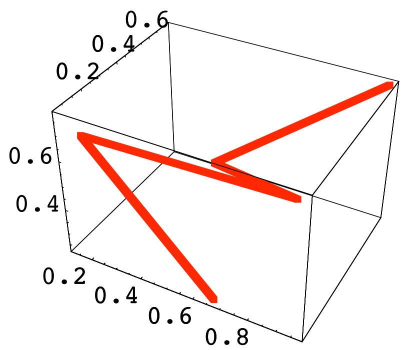
```
orthoproject[x_] := Delete[x, Table[{i, 3}, {i, 1, Length[x]}]];
```

■ Define 3D target object - Wire with randomly positioned vertices

```
threeDtemplate = Table[{Random[], Random[], Random[]}, {5}];
```

```
MatrixForm[threeDtemplate];
```

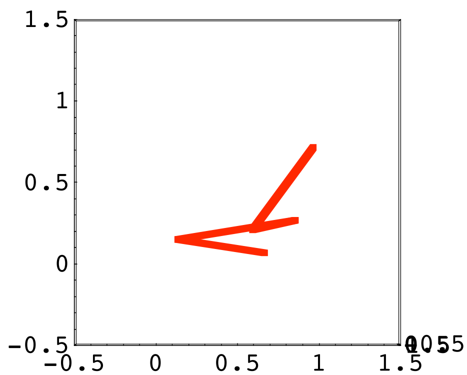
```
ScatterPlot3D[threeDtemplate, PlotJoined -> True,
  PlotStyle -> {{Thickness[0.02], RGBColor[1, 0, 0]}}];
```



■ First view

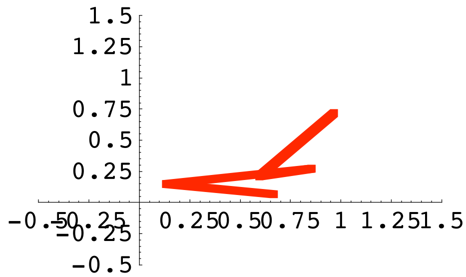
■ View from along Z-direction

```
ScatterPlot3D[threeDtemplate, ViewPoint -> {0, 0, 100}, PlotJoined -> True,
  PlotStyle -> {{Thickness[0.02], RGBColor[1, 0, 0]}},
  PlotRange -> {{-.5, 1.5}, {-.5, 1.5}, {-.5, 1.5}}];
```



■ ListPlot view

```
ovg = ListPlot[orthoproject[threeDtemplate], PlotJoined -> True,
    PlotStyle -> {Thickness[0.02], RGBColor[1, 0, 0]},
    PlotRange -> {{-.5, 1.5}, {-.5, 1.5}}];
```



■ New View

■ Use Homogeneous coordinates

```
swidth = 1.0; sheight = 1.0; slength = 1.0; d = 0;
```

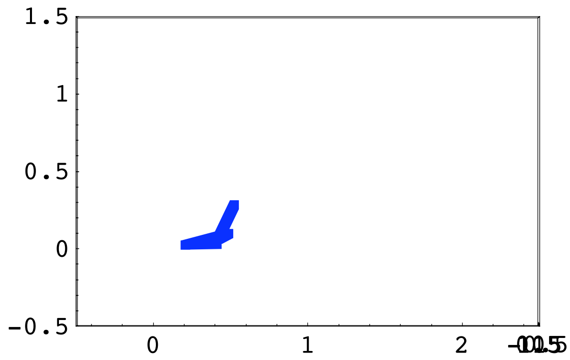
```
homovertices = Transpose[Map[ThreeDToHomogeneous, threeDtemplate]];
newtransformMatrix = TranslateMatrix[1, 0, 0].XRotationMatrix[N[π/16]].
    YRotationMatrix[N[π/16]].ScaleMatrix[swidth, sheight, slength];
```

```
temp = N[newtransformMatrix.homovertices];
```

■ Take a look at the new view

```
newvertices = Map[HomogeneousToThreeD, Transpose[temp]];
```

```
ScatterPlot3D[newvertices, ViewPoint -> {0, 0, 100}, PlotJoined -> True,
  PlotStyle -> {{Thickness[0.02], RGBColor[0, 0, 1]}},
  PlotRange -> {{-.5, 2.5}, {-.5, 1.5}, {-.5, 1.5}}];
```



Geometric variation: 2D image-based modeling

Suppose one wants to check to see if the blue image above is indeed a 3D rotated version of the original familiar object. One could look for rotation parameters that minimize the squared error of the image projection.

Is there a 2D image approximation that provides an alternative, even if not perfect ?

If one projects a small rotation in 3D onto a 2D view, the rotation can be approximated by a 2D affine transformation. Because a 2D affine transformation is a simple 2D operation, perhaps it is sufficient to account for the generalization of familiar to unfamiliar views in human observers (Liu and Kersten).

Affine transformation preserve parallel lines.

We know that rotations, scale and shear transformations will do this. So will translations. It is not immediately apparent, that any matrix operation is an affine transformation, although one has to remember that translations are not represented by matrix operations unless one goes to homogeneous coordinates. Here is a simple demo of the parallel line preservation for transformations of a cube.

```
M[theta_] := {{Cos[theta], Sin[theta]}, {-Sin[theta], Cos[theta]}};
```

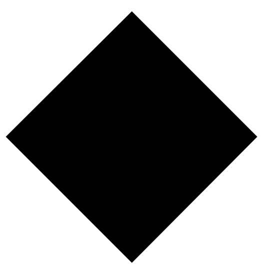
■ Define a square

```
square = {{1, 1}, {-1, 1}, {-1, -1}, {1, -1}};
```

```
Show[Graphics[Polygon[square]], AspectRatio → 1,
  PlotRange → {{-2, 2}, {-2, 2}}];
```

- **A rotation**

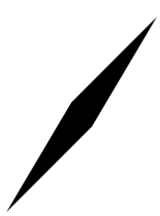
```
Show[Graphics[Polygon[(M[π/4].#1 &) /@ square]], AspectRatio → 1,
  PlotRange → {{-2, 2}, {-2, 2}}];
```



- **Let's try a random matrix to see if it preserves parallel lines**

```
MR = Table[Random[], {2}, {2}]
Show[Graphics[Polygon[(MR.#1 &) /@ square]], AspectRatio → 1,
  PlotRange → {{-2, 2}, {-2, 2}}];
```

```
{{0.737607, 0.561893}, {0.741276, 0.948084}}
```



Compute closest least squares affine match with translation

```
aff = {{a, b}, {c, d}};
tra = Transpose[{{f, g}, {f, g}, {f, g}, {f, g}, {f, g}}];
errorsum :=
  Apply[Plus,
   Flatten[
    (aff.Transpose[orthoproject[newvertices]] + tra -
       Transpose[orthoproject[threeDtemplate]])^2]];
temp = FindMinimum[errorsum, {a, .8}, {b, .2}, {c, .16}, {d, .8},
   {f, 0.0}, {g, 0.0}, MaxIterations -> 200];
minvals = Take[temp, -1][[1]]; minerr = Take[temp, 1][[1]];
naff = aff /. minvals; ntra = tra /. minvals;
minerr
```

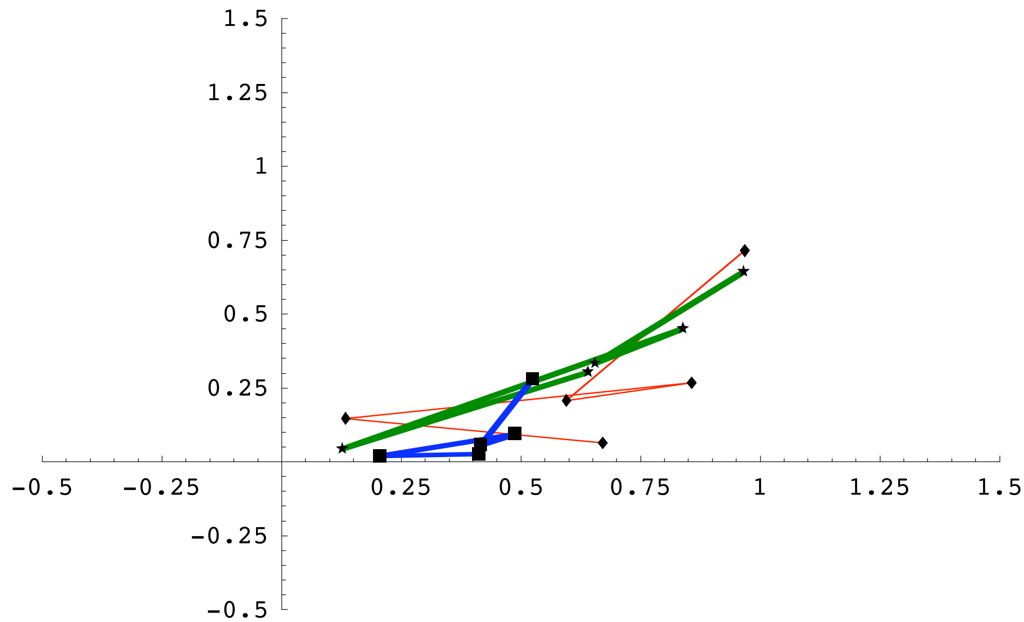
```
0.171834
```

Check match with estimated view

```
estim = naff.Transpose[orthoproject[newvertices]] + ntra;
```

■ **Plot familiar view, new view and the affine estimate of the old from the new**

```
evg = MultipleListPlot[orthoproject[threeDtemplate], Transpose[estim],
   orthoproject[newvertices], PlotJoined -> True,
   PlotStyle -> {{Thickness[0.002], RGBColor[1, 0, 0]},
                 {Thickness[0.005], RGBColor[0, .5, 0]},
                 {Thickness[0.005], RGBColor[0, 0, 1]}},
    PlotRange -> {{-.5, 1.5}, {-.5, 1.5}}];
```



Note that we haven't used 3D rotations to try to get a match between the original familiar view (red) and the new view (blue)--we've assumed that an affine match might get us close. Comparing the red and the green shows how close.

Compute closest least squares affine match without translation

If we have some other means to compensate for translation, we can look for the subset of affine parameters (i.e. matrix parameters) that minimize the total squared error in the reconstruction. Then we can use the built-in **PseudoInverse[]** function to find the solution, essentially in one line:

```
naff2 = Transpose[orthoproject[threeDtemplate]].
  PseudoInverse[Transpose[orthoproject[newvertices]]]
```

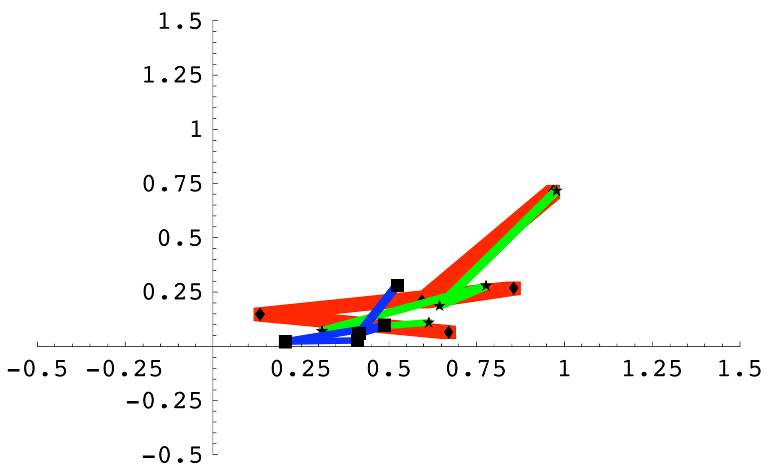
```
{{1.43981, 0.798144}, {0.11845, 2.33565}}
```

Check match with estimated view

```
estim2 = naff2.Transpose[orthoproject[newvertices]];
```

■ **Plot familiar view, new view and the affine estimate of the old from the new**

```
evg = MultipleListPlot[orthoproject[threeDtemplate], Transpose[estim2],
   orthoproject[newvertices], PlotJoined -> True,
   PlotStyle -> {{Thickness[0.02], RGBColor[1, 0, 0]},
                 {Thickness[0.01], RGBColor[0, 1, 0]},
                 {Thickness[0.01], RGBColor[0, 0, 1]}},
    PlotRange -> {{-.5, 1.5}, {-.5, 1.5}}];
```



Note that we again haven't used 3D rotations to try to get a match between the original familiar view (red) and the new view (blue)--we've assumed that an 2D matrix operation might get us close to a match. Comparing the red and the green shows how close.

Background context, clutter

■ Background/context for "indexing"

Background can provide prior information, that could be called "index" cues, to narrow down the space of possible objects to be recognized. E.g see: Oliva et al. (2003), Torralba A, Sinha P (2001)

One of the first demonstrations of the role of background for human perception is:

Biederman I (1972) Perceiving real-world scenes. Science 177:77-80.

■ Background (clutter) as a confound

Variation over background (clutter) is challenging, very important, yet poorly understood.

Need a better understanding of local image cues, as well as how high-level models can be used to disambiguate local information

Natural image statistics:

Brady, M. J., Legge, G., & Kersten, D. (2004). Effects of natural backgrounds on spatial filter responses near object contours [Abstract]. *Journal of Vision*, *4*(8), 535a, http://journalofvision.org/4/8/535/, doi:10.1167/4.8.535

The same image of an object appearing at different locations will produce quite different local responses in spatial filters.

Place the antlers

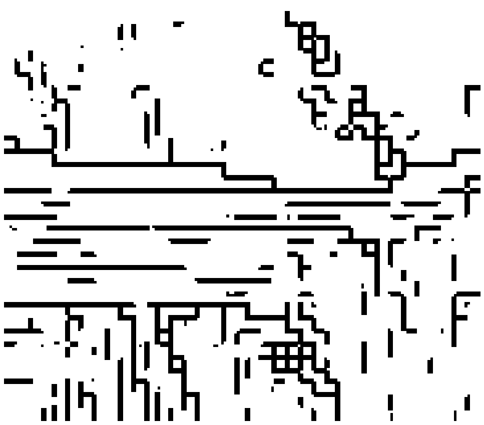
on background location 1

or on background location 2

Compare the local information in the following blow ups for location 1

and location 2

Here are examples of edge detector outputs for the two conditions:



Konishi SM, Yuille AL, Coughlan JM, Zhu SC (2003) Statistical edge detection: Learning and evaluating edge cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 25:57-74.

High-level information:

Cavanagh P (1991) What's up in top-down processing? In: Representations of Vision: Trends and tacit assumptions in vision research (Gorea A, ed), pp 295-304. Cambridge, UK: Cambridge University Press.

Sinha P, Poggio T (2001) High-level learning of early perceptual tasks. In: Perceptual Learning (Fahle M, ed). Cambridge, MA: MIT Press.

Bootstrapped learning of object models in clutter

Brady MJ, Kersten D (2003) Bootstrapped learning of novel objects. J Vis 3:413-422.

http : // vision.psych.umn.edu / www / kersten – lab / camouflage / digitalembryo.html

Next & wrapping up...

■ Spatial layout

■ Theories of neocortex: How is high-level information combined with local features in the brain?

Feedforward, predictive coding, resonance, ...

Grossberg (1980),, Carpenter and Grossberg (1986)

Rao and Ballard (1999)

Bullier (2001)

Friston (2003)

Lee & Mumford (2003)

■ Learning concepts

Appendix

Load in: **Homogeneous.m**

```
Get[Experimental`FileBrowse[False]];
```

```
Experimental`FileBrowse[False]
```

■ Writing Packages

The basic format is straightfoward:

```
BeginPackage["Geometry`Homogeneous`"]
XRotationMatrix::"usage" =
 "XRotationMatrix[phi] gives the matrix for rotation about x-
   axis by phi degrees in radians"
YRotationMatrix::"usage" =
 "YRotationMatrix[phi] gives the matrix for rotation about y-
   axis by phi degrees in radians"
ZRotationMatrix::"usage" =
 "ZRotationMatrix[phi] gives the matrix for rotation about z-
   axis by phi degrees in radians"
ScaleMatrix::"usage" =
 "ScaleMatrix[sx,sy,sz] gives the matrix to scale a vector by
   sx,sy, and sz in the x, y and z directions, respectively."
TranslateMatrix::"usage" =
 "TranslateMatrix[x,y,z] gives the matrix to translate coordinates
   by x,y,z."
ThreeDToHomogeneous::"usage" =
 "ThreeDToHomogeneous[sx,sy,sz] converts 3D coordinates to 4D
   homogeneous coordinates."
HomogeneousToThreeD::"usage" =
 "HomogeneousToThreeD[4Dvector] converts 4D homogeneous coordinates
   to 3D coordinates."
ZProjectMatrix::"usage" =
 "ZProjectMatrix[focal] gives the 4x4 projection matrix to map
   a vector through  the origin to an image plane at focal
   distance from the origin along the z-axis."
ZOrthographic::"usage" =
 "ZOrthographic[vector] projects vector on to the x-y plane."
Begin["`private`"]
XRotationMatrix[theta_] :=
  {{1, 0, 0, 0}, {0, Cos[theta], -Sin[theta], 0},
   {0, Sin[theta], Cos[theta], 0}, {0, 0, 0, 1}};
YRotationMatrix[theta_] :=
  {{Cos[theta], 0, Sin[theta], 0}, {0, 1, 0, 0},
   {-Sin[theta], 0, Cos[theta], 0}, {0, 0, 0, 1}};
ZRotationMatrix[theta_] :=
  {{Cos[theta], -Sin[theta], 0, 0}, {Sin[theta], Cos[theta], 0, 0},
   {0, 0, 1, 0}, {0, 0, 0, 1}};
ScaleMatrix[sx_, sy_, sz_] :=
  {{sx, 0, 0, 0}, {0, sy, 0, 0}, {0, 0, sz, 0}, {0, 0, 0, 1}};
(*TranslateMatrix[x_,y_,z_]:=
   {{1,0,0,x},{0,1,0,y},{0,0,1,z},{0,0,0,1}};*)
TranslateMatrix[x_, y_, z_] :=
  {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {x, y, z, 1}};
ThreeDToHomogeneous[vec_] := Append[vec, 1];
```

```
HomogeneousToThreeD[vec_] := Drop[vec/vec[[4]], -1];
ZProjectMatrix[focal_] :=
  {{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, N[1/focal], 0}};
ZOrthographic[vec_] := Take[vec, 2];
End[]
EndPackage[]
```

```
Geometry`Homogeneous`
```

References

Biederman I (1972) Perceiving real-world scenes. Science 177:77-80.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. Psychological Review, **94,** 115-147.

Brady MJ, Kersten D (2003) Bootstrapped learning of novel objects. J Vis 3:413-422.

Brady, M. J., Legge, G., & Kersten, D. (2004). Effects of natural backgrounds on spatial filter responses near object contours [Abstract]. *Journal of Vision*, *4*(8), 535a, http://journalofvision.org/4/8/535/, doi:10.1167/4.8.535

Bullier J (2001) Integrated model of visual processing. Brain Res Brain Res Rev 36:96-107.

Bülthoff, H. H., & Edelman, S. (1992). Psychophysical support for a two-dimensional view interpolation theory of object recognition. Proc. Natl. Acad. Sci. USA, 89, 60-64.

Carpenter GA, Grossberg S (1986) A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. In: Computer Vision, Graphics and Image Processing.

Cavanagh P (1991) What's up in top-down processing? In: Representations of Vision: Trends and tacit assumptions in vision research (Gorea A, ed), pp 295-304. Cambridge, UK: Cambridge University Press.

Cohen MA, Grossberg S (1984) Neural dynamics of brightness perception: features, boundaries, diffusion, and resonance. Percept Psychophys 36:428-456.

David, C., & Zucker, S. W. (1989). Potentials, Valleys, and Dynamic Global Coverings (TR-CIM 98-1): McGill Research Centre for Intelligent Machines, McGill University.

Friston K (2003) Learning and inference in the brain. Neural Netw 16:1325-1352.

Grossberg S (1980) How does a brain build a cognitive code? Psychological Review 87:1-51.

Grossberg S, Mingolla E (1985) Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. Percept Psychophys 38:141-171.

Grossberg S (1986) Competitive Learning: From Interactive Activation to Adaptive Resonance. In: Cognitive Science.

Konishi SM, Yuille AL, Coughlan JM, Zhu SC (2003) Statistical edge detection: Learning and evaluating edge cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 25:57-74.

Lee TS, Mumford D (2003) Hierarchical Bayesian inference in the visual cortex. J Opt Soc Am A Opt Image Sci Vis 20:1434-1448.

Liu, Z., Knill, D. C. & Kersten, D. (1995). Object Classification for Human and Ideal Observers. *Vision Research*, ***35,*** 549-568.

Liu, Z., & Kersten, D. (1998). 2D observers for 3D object recognition? In *Advances in Neural Information Processing Systems* Cambridge, Massachusetts: MIT Press.

Logothetis, N. K., Pauls, J., Bulthoff, H. H. & Poggio, T. (1994). View-dependent object recognition by monkeys. *Current Biology*, *4 No 5*, 401-414.

Logothetis, N. K., & Sheinberg, D. L. (1996). Visual Object Recognition. Annual Review of Neuroscience, 19, 577-621.

Mumford D (1994) Neuronal architectures for pattern-theoretic problems. In: Large-Scale Neuronal Theories of the Brain (Koch C, Davis JL, eds), pp 125-152. Cambridge, MA: MIT Press.

Oliva Aude , Torralba Antonio , Castelhano Monica S. , and Henderson John M. . (2003) Top-Down Control of Visual Attention in Object Detection. , International Conference on Image Processing (ICIP). Vol. I, pages 253-256. September 14-17, in Barcelona, Spain

Poggio, T. & Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature*, ***343*,** 263-266.

Rao RP, Ballard DH (1999) Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects [see comments]. Nat Neurosci 2:79-87.

Rock, I. & Di Vita, J. (1987). A case of viewer-centered object perception. *Cognitive Psychology*, ***19*,** 280-293.

Sinha P, Poggio T (2001) High-level learning of early perceptual tasks. In: Perceptual Learning (Fahle M, ed). Cambridge, MA: MIT Press.

Tanaka, K. (1996). Inferotemporal cortex and object vision. Annual Review of Neuroscience, 19, 109-139.

Tarr, M. J., & Bülthoff, H. H. (1995). Is human object recognition better described by geon-structural-descriptions or by multiple-views? Journal of Experimental Psychology: Human Perception and Performance, **21**(6), 1494-1505.

Torralba A, Sinha P (2001) Statistical Context Priming for Object Detection. In: Proceedings of the International Conference on Computer Vision, ICCV01, pp 763-770. Vancouver, Canada.

Torralba A, Oliva A (2003) Statistics of natural image categories. Network 14:391-412.

Ullman, S. (1996). High-level Vision: Object Recognition and Visual Cognition. Cambridge, Massachusetts: MIT Press.