Computational Vision
U. Minn. Psy 5036
Daniel Kersten
Lecture 24: Object Recognition

Initialize

- **Spell check off**

```
Off[General::spell1];
```

Outline

Today

Object recognition

Role of geometric modeling in theories of object recognition

Example of an ideal observer that does discounting

High-level vision, visual tasks

Intermediate-level vision

How much can vision achieve with only generic object and surface properties?

Generic, global organizational processes

Domain overlap, occlusion

Surface grouping, selection

Gestalt principles

Cue integration

Cooperative computation

Attention

Intermediate-level processes could be useful for interpreting novel objects and scenes. These processes could also be useful for feature extraction to be used to store in memory, and later for testing match against stored representations. The idea is that more abstract features and relations would be more robust to image variations, of the sort discussed below.

High-level vision

Functional tasks

Object recognition--familiar objects

entry-level, subordinate-level

Object-object relations

Scene recognition

Spatial layout

Viewer-object relations

Object manipulation

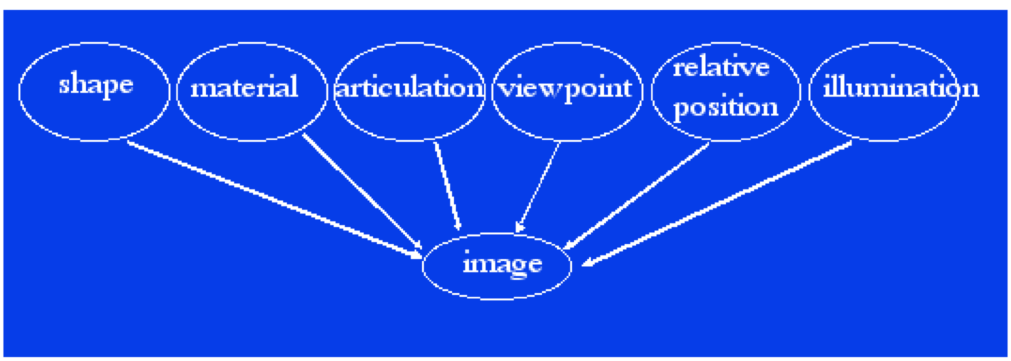
reach & grasp

Heading

Task dependency: explicit (primary) and generic (secondary, nuisance) variables

One can't think of invariance without considering what has to be discounted for a given type of task to achieve it. Consider several classes of scene causes of image pattern I.

I=f(shape, material, articulation,viewpoint,relative position, illumination)



Which variables are more important to estimate precisely for various tasks?

■ Task: Object Recognition (labelling)

I=f(**shape**, **material**, articulation,viewpoint,relative position, illumination)

■ Task: Absolute depth (e.g. for reaching)

I=f(shape, material, articulation,**viewpoint**,relative position, illumination)

■ Task: grasp

I=f(**shape**, material, **articulation**,viewpoint,relative position, illumination)

Problem: *all* the scene variables contribute to the variations in the image

■ Preview of *Mathematica* application below

Shape-based object recognition

estimate geometrical shape (primary variables)

discount sources of image variation not having to do with shape (secondary variables)

e.g. integrating out geometrical variables such as translation, rotation, and scale to estimate shape for object recognition

(Variations due to illumination, background clutter, and within-category shape also need to be taken into account.)

Object recognition

Analysis of image variation

Which variables are important to estimate for recognition depends on the level of abstraction required.

■ Variation within subordinate-level category

What distinguishes a mallard from a wood duck? Honeycrisp from a Braeburn? Doberman from a Alsation?

Here's a mental exercise: think of all the ways the images (or "appearances") of an object, like a mallard duck might vary. List the generative causes.

illumination

level, direction, source arrangement, shadows, spectral content

view

scale

translation

2D & 3D rotation

articulation

non-rigid,

e.g. joints, hinges, facial expression, hair, cloth, wings,

physical size

small and big apples, shoes, dogs, ...

background (segmentation)

bounding contour (affected by variation in pattern of intensities over the boundary regions)

occlusion (segmentation)

What is left?

■ Variation within basic-level category

What distinguishes a duck from a dog, or from a chair or an apple?

Part types and their spatial relationships

How to achieve part-relation invariance within a category? I.e. most coffee cups have a hollow cylinder and a handle. While the

shapes of parts and relative positions can vary somewhat, there is a level of invariance that constrains an instance of a cup to be labeled as a “cup”.

We will look at a structural relations theory, and diagnostic “fragment” theory

■ Variation across super-ordinate category

(e.g. bird, mammal, furniture, fruit)

more cognitive than perceptual, non-pictorial

Human vision

■ Basic-level vs. subordinate-level

Is the distinction relevant to human behavior?

Behavioral experiments (Rosch et al.),

Neuropsychological (Damasio and Damasio)

temporal lobe lesions disrupt object recognition

fine-grain distinctions more easily disrupted than coarse-grain ones

e.g. Boswell patient—can't recognize faces of family, friends, unique objects, or unique places. Can assign names like face, house, car, appropriately.

Also superordinate categories: "tool"

prosopagnosics

faces vs. subordinate-level?

neural evidence for distinction? IT hypercolumns?

■ Basic-level

Shape-particularly critical -- but qualitative, rather than metric aspects important.

E.g. geons and geon relations (Biederman).

Material, perhaps for some natural classes?

Issue of prototypes with a model for variation vs. parts.

e.g. average image face, the most familiar

priming methods to tease apart distinct representations

Fragment-based methods or "features of intermediate complexity": Ullman, S., Vidal-Naquet, M., & Sali, E. (2002). Learning informative features for class categorization

■ Subordinate-level

geometric variations important for subordinate--e.g. sensitivity to configurational etc..

material

Prototypes -> what kind of model for variation?

Problem: With only a discrete set of views, how does vision generalize to other views?

Other factors

Object ensemble

E.g imagine only one red object in your world – no need to process its shape.

Computational complexity

E.g. a crumpled piece of paper -- difficult to compute and remember exact shape.

Context is important

Small red thing flying past the window.

High "cue validity" for male Cardinal bird

■ Context can even over-ride local cues for identity

Sinha and Poggio, Nature 1996

Democrat coalition.

See too: Cox, D., Meyers, E., & Sinha, P. (2004). Contextually evoked object-specific responses in human visual cortex. *Science, 304*(5667), 115-117.

Getting a good image representation

Overview of problems and processes that seem to be necessary

For object recognition, the contributions due to the secondary or "generic variables", (e.g. illumination and viewpoint) need to be discounted, and object features such as shape and material need to be estimated. How?

o Measurements of image information likely to belong to the object. This principle should constrain segmentation.

regions with similar textures, super-pixels (Shi and Malik, 200; Sharon et al., 2006)

problems with: specularities, cast shadows, attached shadows (from shading).

edge detection is really noisy, and ambiguous as to cause, so what are these image "features"?

although noisy, are edges/groupings sufficiently reliable to determine object class?

o "Sensor fusion" or cue integration to improve estimates of where object boundaries are located:

combine stereo, motion, chromatic, luminance, etc..

o Incorporate intermediate-level constraints to help to find object boundaries or "silhouettes".

Gestalt principles of perceptual organization (symmetry, similarity, proximity, closure, common fate, continuity,..)

long smooth lines (David & Zucker, 1989; Shashua & Ullman, 1988; Field and Hess, 1993)

o "Cooperative computation" for object shape, reflectance and lighting.

"intrinsic images" of Barrow and Tenenbaum

explaining away, e.g. for occlusion

Problem: Still no bottom-up procedure for perfect segmentation or edge-parsing. Some improvements with top-down processes in limited domains.

Open questions regarding shape representations

Object geometry--Surfaces & shape, small scale surface structure

How can we describe objects themselves in terms of their geometry?

Are objects represented in terms of a view-dependent, dense local representation of shape, e.g. local surface orientation?

Or intrinsic properties, such as curvature?

Or in terms of parts? What is the relationship of parts of objects to each other?

Compositional representations

Role in object recognition, e.g. structural descriptions

Storing information about an object and matching stored information to new appearances

In the next section, we take a closer look at severak ways in which to store information about 3D objects in a way that is useful for recognizing novel instances or views of these previously seen objects.

Structural description: high-level features or parts plus relations. How would you describe the letters

A, A, A, A, and A that is independent of font? Strokes and their spatial relationships.

Image-based: low-level features plus metric comparisons, transformations?

2D views?

3D object-centered?

Given a representation of the image information likely to be due to 3D object in memory, how does the brain store, then later when given another view, index and verify? Consider two extremes:

Nearest neighbor to 2D views?

Transformation of 3D model to fit 2D view?

Or something in between?

Two broad classes of models for object recognition

View independent, structural description

Structural description theories. Use invariants to find parts (assumption is that this is easier than for the whole object), build up description of the relations between the parts, this description specifies the object. E.g. a triangle shape, the letter "A" (three parts, with two "cross relations" and one " cotermination" relation".

(Could be based on 2.5 D sketch => object-centered representation that is independent of viewpoint? e.g. Marr's generalized cylinders)

predicts view-point independence

(Biederman, 1987) extraction of invariants, "non-accidental properties", such as:

co-linearity of points or lines => colinearity in 3D

cotermination of lines=>cotermination in 3D (e.g. Y and arrow vertices)

skewed symmetry in 2d=>symmetry in 3D

curved line in 2D =>curved line in 3D

parallel curves in 2D => parallel in 3D (over small regions)

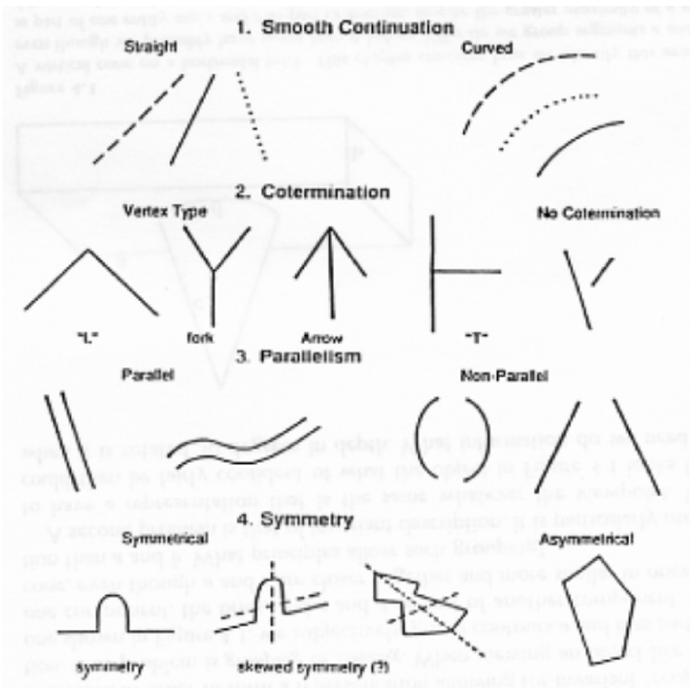
The considerations of non-accidental image properties lead to the idea

of objects being represented in terms of elementary "parts" or

=> geons (box, cylinder, wedge, truncated cone, etc..)

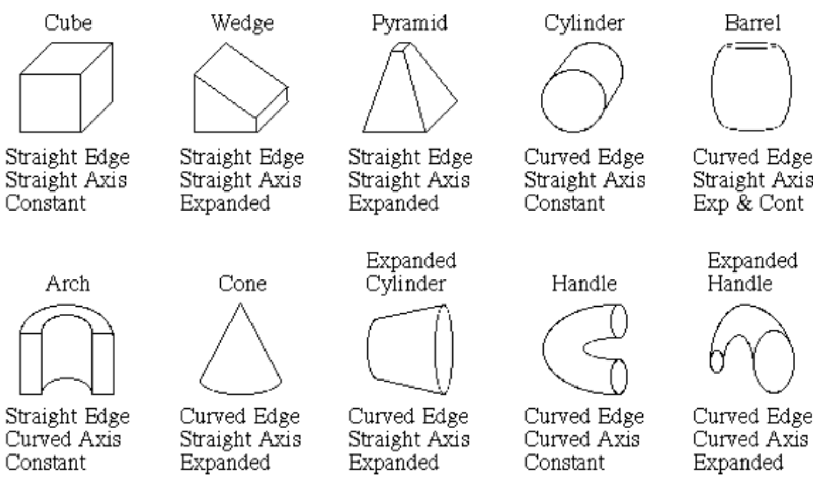
and a description of their spatial relationships to each other.

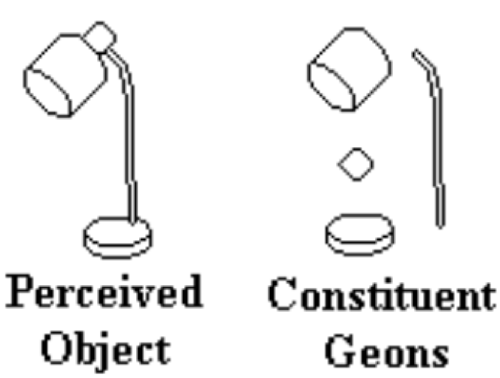
partial independence of viewpoint

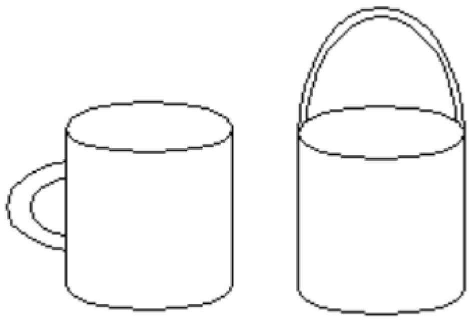


Biederman, I. (1987). Recognition-by-components: A theory of human image understanding.

Psychological Review, 94, 115-147.







Geon theory: example of coding an object into a (partially) view-independent description

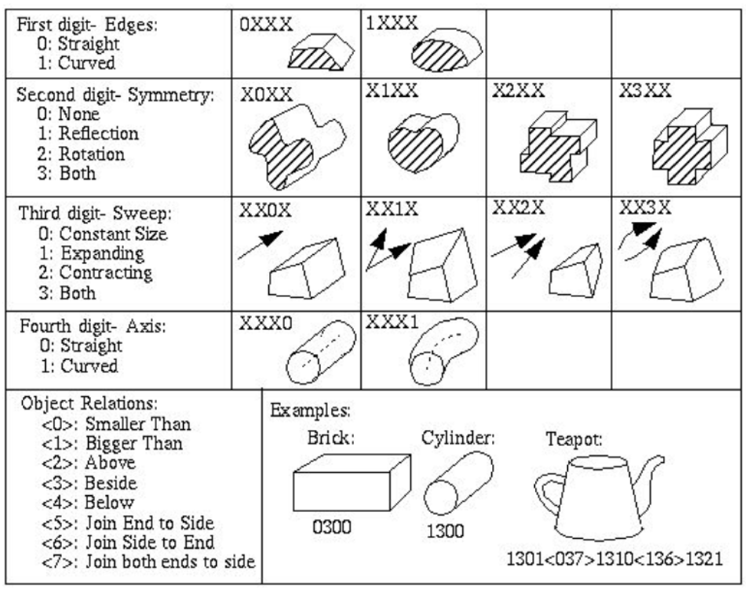


Image-based, alignment methods.

Relation to view-dependency

E.g. Ullman, alignment of pictorial descriptions

Image description S, memory model M

-Matching

TEST: S = F(M)? Here one imagines the brain using its generative model F, to test for inputs.

or one could try to test: F^{-1} (S) = M? Here one imagines the brain has processes that could operate feedforward to get the input in the right invariant format to test against memories.

What are possible representations of M?

And how to model F?

->Image-based or "Exemplar" theories

view-specific features are stored in memory predicts view-point dependence (e.g. Rock & DiVita (1987)

Poggio & Edelman, 1990; Bülthoff & Edelman, 1992;

Tarr & Bülthoff, 1995; Liu, Knill & Kersten (1995)

Troje & Kersten (1999)

Model may depend on object class and task (e.g. basic vs. subordinate)?

How sophisticated are transformation processes in human recognition?

(Liu, Knill & Kersten, 1995; Liu & Kersten, 1998)

Ideal observer analysis applied to the problem of view-dependency in 3D object recognition

One can imagine two quite different ways of verifying whether an unfamiliar view of an object belongs to the object or not. One way is to simply test how close the new view is to the set of stored views, without any kind of "intelligent" combination of the stored views. Given a sufficiently good representation, a simple measure of similarity could produce good recognition performance over restricted sets of views (i.e. not too much self-occlusion).

Another way is to combine the stored views in a way that reflects knowledge that they are from a 3D object, and compare the new view to the combined view. The second approach has the potential for greater accuracy than the first.

An example of the second approach would be to use the familiar views to interpolate the unfamiliar views. Given sufficient views and feature points, this latter approach has a simple mathematical realization (Ullman, 1996). An optimal verification algorithm would verify by rotating the actual 3D model of the object, projecting it to 2D and testing for an image match.

Liu et al. (1995) were able to exclude models of the first class (comparisons in 2D) and the last class (comparisions with a full 3D model) in a simple 3D classification task using ideal observer analysis. The ideal observer technique was developed in the context of our studies of quantum efficiency in early vision.

Review of types of image modeling (from Lecture 7)

Generative models for images: rationale

Generative vs. discriminative models

Discriminative models don’t explicitly model how the image results from the object description. In Bayesian terms, an algorithm is based on:

p(object | image). For example, the posterior could be constructed as a look-up table: input image, check probabilities on various object descriptions, and pick the one with the biggest posterior probability.

Generative models characterize the range of variations in the image. In Bayesian terms, algorithms explicitly model the likelihood:

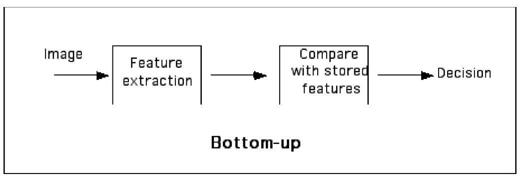
$p(\text{object} \mid \text{image}) \propto p(\text{image} \mid \text{object})\, p(\text{object})$

The pros for a generative are, if the visual system has built-in knowledge that can recapitulate the generative process, then recognition should be able to better generalize to novel appearances of a learned object. E.g. to deal with 3D rotations, occlusion. Cons are that modeling generative processes can be complex, and take time.

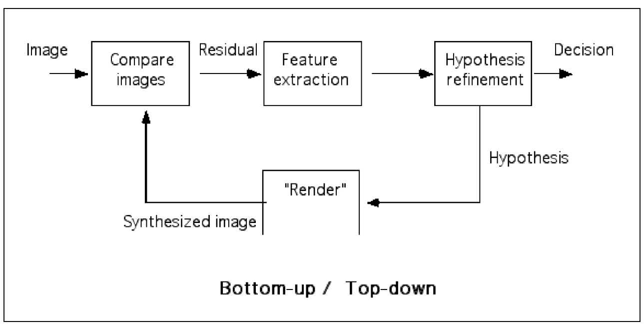
We review the types of generative models with a view to asking whether human vision can be modeled as incorporating a particular kind of generative knowledge for object recognition.

■ Characterize the knowledge required for inference

Feedforward procedures:



Pattern theory perspective: "analysis by synthesis"--synthesis phase explicitly incorporates generative model



Note that a top-down generative model can be used in more than one way. The above figure shows it being used to find errors in the top-down predictions. It could also be used to find consistent features.

Generative models can make it easier to conceptualize information flow: Mapping is many-to-one.

But as pointed out above, necessarily easy to compute. For example, realistic 3D graphics rendering is computationally intensive.

■ Two basic concepts: Photometric & geometric variation

■ Two more basic concepts: 3D scene-based & 2D image-based models of geometric variation

3D Scene-based modeling: Computer graphics models

■ Objects & surfaces: Shape, Articulations, Material & texture

■ Illumination: Points and extended, Ray-tracing, Radiosity

■ Viewpoint/Camera

Projection geometry, homogeneous coordinates:

perspective, orthographic

Application to viewpoint variation: Does human vision compensate for variations using "built-in" knowledge of 3D?

Image-based modeling

■ Linear intensity-based

Basis sets:

I = m1*I1 + m2*I2 + m3*I3 + ...

application: optics of the eye

application: illumination variation for fixed views of an object, useful in object recognition

■ Linear geometry-based

Affine:

rigid translations, rotations, scale and shear

Application to viewpoint variation: 2D approximations to 3D variations?

■ Non-linear geometry-based

Morphs

Application:within-category variation for an object,or objects

finding the "average" face

Both linear and non-linear based methods raise the general question:

Does human recognition store object prototypes together with some

perhaps image-based model of possible transformations to look for

a match of incoming image data with a stored template?

Modeling geometric variation: 3D scene-based modeling

Let's assume that the 2D spatial locations of certain features are stored for a given object. The math that describes how the 3D locations can be transformed and mapped on to the retinal coordinates has been known for many years, and is built into virtually all 3D graphics engines. The math will give us a handle on how to quantitatively think about matching image features to stored memory.

Let's review the math. There are four basic types of transformations: rotation, scale, translation, and projection. First rotations. Then we'll put rotations together with the other transformations using homogeneous coordinates.

Representing Rotations

See Rotate[] and RotationMatrix[] for built-in *Mathematica* functions for doing rotations.

■ Euler angles

Euler angles are a standard way of representing rotations of a rigid body.

A rotation specified by the Euler angles *psi*, *theta*, and *phi* can be decomposed into a sequence of three successive rotations: first by angle *psi* about the *z* axis, the second by angle *theta* about the *x* axis, and the third about the *z* axis (again) by angle *phi*. The angle *theta* is restricted to the range 0 to π.

```
RotationMatrix3D[ψ_, θ_, ϕ_] := ( Cos[ϕ] Cos[ψ] - Cos[θ] Sin[ϕ] Sin[ψ]    Cos[θ]
                                  -Cos[ψ] Sin[ϕ] - Cos[θ] Cos[ϕ] Sin[ψ]   Cos[θ]
                                             Sin[θ] Sin[ψ]
```

```
RotationMatrix3D[ψ, θ, ϕ] // MatrixForm
```

```
( Cos[ϕ] Cos[ψ] - Cos[θ] Sin[ϕ] Sin[ψ]    Cos[θ] Cos[ψ] Sin[ϕ] + Cos[ϕ] Sin[ψ]   Sin
 -Cos[ψ] Sin[ϕ] - Cos[θ] Cos[ϕ] Sin[ψ]   Cos[θ] Cos[ϕ] Cos[ψ] - Sin[ϕ] Sin[ψ]   Cos
            Sin[θ] Sin[ψ]                            -Cos[ψ] Sin[θ]
```

```
RotationMatrix3D[ψ, 0, 0] // MatrixForm
```

```
( Cos[ψ]   Sin[ψ]  0 )
( -Sin[ψ]  Cos[ψ]  0 )
(   0        0     1 )
```

```
RotationMatrix3D[0, θ, 0] // MatrixForm
```

```
( 1     0        0    )
( 0   Cos[θ]   Sin[θ] )
( 0  -Sin[θ]   Cos[θ] )
```

```
RotationMatrix3D[0, 0, ϕ] // MatrixForm
```

```
( Cos[ϕ]   Sin[ϕ]  0 )
( -Sin[ϕ]  Cos[ϕ]  0 )
(   0        0     1 )
```

Homogeneous coordinates

Rotation and scaling can be done by linear matrix operations in three-space. Translation and perspective transformations do not have a three dimensional matrix representation. By going from three dimensions to four dimensional coordinates, all four of the above basic operations can be represented within the formalism of matrix multiplication.

Homogeneous coordinates are defined by: {xw, yw, zw, w }, (w not equal to 0). To get from homogeneous coordinates to three-space coordinates, {x,y,z}, divide the first three homogeneous coordinates by the fourth, {w}. For more information, see references on 3D graphics, e.g. Foley, J., van Dam, A., Feiner, S., & Hughes, J. (1990).

The rotation and translation matrices can be used to describe object or eye-point changes of position. The scaling matrix allows you to squash or expand objects in any of the three directions. Any combination of the matrices can be multiplied together or concatenated. But remember, matrices do not in general commute, so the order is important. The translation, rotation, and perspective transformation matrices can be concatenated to describe general 3-D to 2-D perspective mappings.

We will use these definitions later when we develop a theory for computing structure from motion, spatial layout and direction of heading.

```
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, 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, 0, -N[1/focal]}, {0, 0, 0, 1}};
ZOrthographic[vec_] := Take[vec, 2];
```

■ Translation by {d_x, d_y, d_z} can be found by applying the matrix

```
Clear[d];
TranslateMatrix[dx, dy, dz] // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_x & d_y & d_z & 1 \end{pmatrix}$$

```
{x, y, z, 1}.TranslateMatrix[dx, dy, dz]
```

```
{x + dx, y + dy, z + dz, 1}
```

to {x,y,z,1}

$$(x+d_x,\quad y+d_y,\quad z+d_z,\quad 1) = (x,\quad y,\quad z,\quad 1)\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_x & d_y & d_z & 1 \end{pmatrix}$$

■ **The scaling matrix is:**

```
ScaleMatrix[sx, sy, sz] // MatrixForm
```

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

■ **There are three matrices for general rotation:**

z-axis (moving the positive x-axis towards the positive y-axis)

```
ZRotationMatrix[θ] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] & \text{Sin}[\theta] & 0 & 0 \\ -\text{Sin}[\theta] & \text{Cos}[\theta] & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

x-axis (moving the positive y towards the positive z)

```
XRotationMatrix[θ] // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}[\theta] & \text{Sin}[\theta] & 0 \\ 0 & -\text{Sin}[\theta] & \text{Cos}[\theta] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

y-axis (moving positive z towards positive x):

```
YRotationMatrix[θ] // MatrixForm
```

$$\begin{pmatrix} \text{Cos}[\theta] & 0 & -\text{Sin}[\theta] & 0 \\ 0 & 1 & 0 & 0 \\ \text{Sin}[\theta] & 0 & \text{Cos}[\theta] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

■ Perspective

Perspective transformation is the only one that requires extracting the three-space coordinates by dividing the homogeneous coordinates by the fourth component w. The projection plane is the x-y plane, and the focal point is at z = d. Then {x, y, z, 1 } maps onto {x, y, 0, -z/d + 1 } by the following transformation:

```
Clear[d]
ZProjectMatrix[d] // MatrixForm
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{d} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

After normalization, the image coordinates {x',y',z'} are read from:

$$\left(x', \quad y', \quad z', \quad 1 \right) = \left(\frac{xd}{d-z,} \quad \frac{yd}{d-z,} \quad 0, \quad 1 \right)$$

The steps can be seen here:

```
Clear[x, y, z, d]
{x, y, z, 1}.ZProjectMatrix[d]
{x, y, z, 1}.ZProjectMatrix[d] / %[[4]]
HomogeneousToThreeD[{x, y, z, 1}.ZProjectMatrix[d]]
Simplify[
 ZOrthographic[HomogeneousToThreeD[{x, y, z, 1}.ZProjectMatrix[d]]]]
```

$$\left\{x, y, 0, 1-\frac{z}{d}\right\}$$

$$\left\{\frac{x}{1-\frac{z}{d}}, \frac{y}{1-\frac{z}{d}}, 0, 1\right\}$$

$$\left\{\frac{x}{1-\frac{z}{d}}, \frac{y}{1-\frac{z}{d}}, 0\right\}$$

$$\left\{\frac{d\,x}{d-z}, \frac{d\,y}{d-z}\right\}$$

The matrix for orthographic projection has d-> infinity.

```
Limit[ZOrthographic[HomogeneousToThreeD[{x, y, z, 1}.ZProjectMatrix[d]]],
 d → ∞]
```

$$\{x, y\}$$

The perspective transformation is the only singular matrix in the above group.This means that, unlike the others its operation is not invertible. Given the image coordinates, the original scene points cannot be determined uniquely.

Example: transforming, projecting a 3D object

We are going to generate a "view" of random 3D object. Imagine that you've seen this view (**threeDtemplate**) and stored it in memory. Later you get a view (**newvertices**) of either the same object or a different one, and you want to check if it is the same object as before. You need to make some kind of comparison test.

We'll keep it simple and do orthographic projection.

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

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

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

■ First view

■ View from along Z-direction

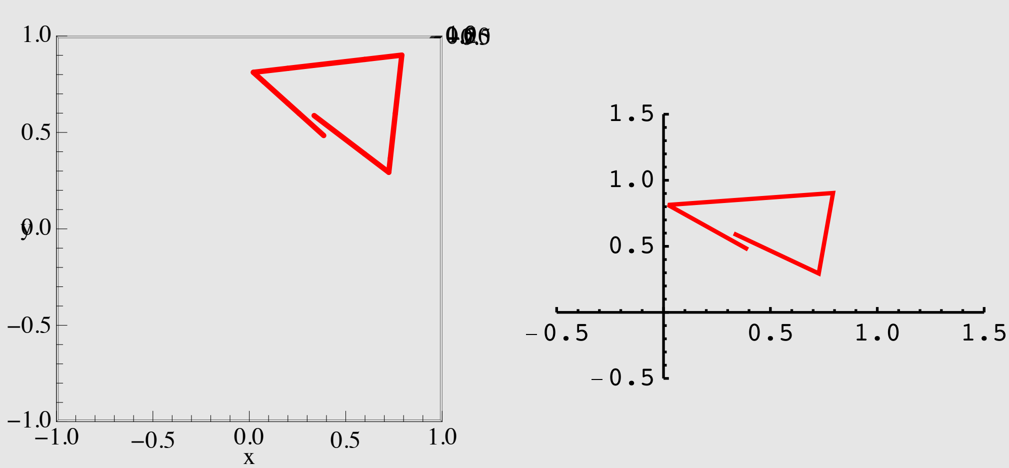
```
lines = Partition[threeDtemplate , 2, 1];
fv3d = Graphics3D[{Thick, Red, Line[lines]}, ViewPoint → {0, 0, 100},
  PlotRange → {{-1, 1}, {-1, 1}, {-1, 1}}, AspectRatio → 1,
  Axes → True, AxesLabel → {"x", "y", "z"}, ImageSize → Small,
  PreserveImageOptions → True];
```

■ ListPlot view

We can also do the projection ourselves:

```
ovg = ListPlot[orthoproject[threeDtemplate], Joined -> True,
   PlotStyle -> {Thickness[0.01], RGBColor[1, 0, 0]},
   PlotRange -> {{-.5, 1.5}, {-.5, 1.5}}];
```

```
GraphicsRow[{fv3d, ovg}]
```



■ New View

We pick an arbitrary view of the above object.

■ **Use Homogeneous coordinates**

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

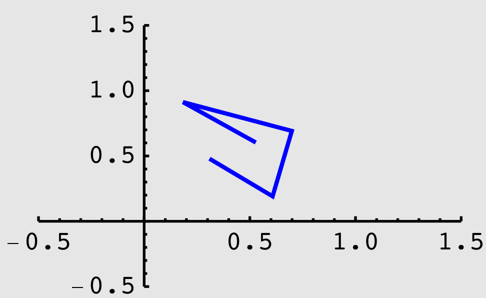
```
homovertices = Transpose[Map[ThreeDToHomogeneous, threeDtemplate]];
newtransformMatrix = TranslateMatrix[.3, 0, 0].XRotationMatrix[N[π/2 * .3]].
   YRotationMatrix[N[-π/2 * .2]].ScaleMatrix[swidth, sheight, slength];
```

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

■ **Take a look at the new view**

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

```
ListPlot[orthoproject[newvertices], Joined -> True,
 PlotStyle -> {Thickness[0.01], RGBColor[0, 0, 1]},
 PlotRange -> {{-.5, 1.5}, {-.5, 1.5}}, ImageSize → Small]
```



■ **Exercise: look at new view by coding the orthographic projection yourself**

Modeling geometric variation: 2D image-based modeling as an approximation to 3D scene-based variation

Suppose that we encounter a new view of the 3D object, i.e. from some new arbitrary viewpoint. This new viewpoint can be modeled as a 3D rotation and translation of the object.

If we want to see if the new and old images are of the same object, we could try to rotate a 3D representation of the object. But this would require knowledge of 3D.

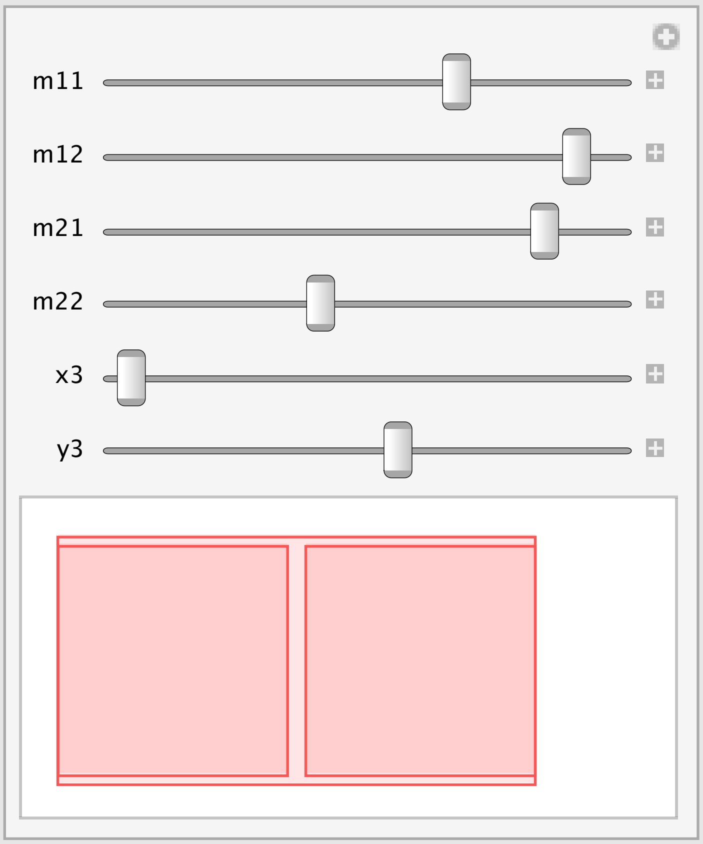
Alternatively, if one projects a rotation in 3D onto a 2D view, we can try to approximate the rotation by a 2D affine transformation. A 2D affine transformation is a simple 2D operation, perhaps it is sufficient to account for the generalization of familiar to unfamiliar views?

■ Affine transformation preserves parallel lines.

We know that rotations, scale and shear transformations will preserve parallel lines. 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. Lecture 7 had a simple demo of the parallel line preservation for transformations of a square.

- **Try to find values of a 2D matrix (M2={{m11,m12},{m21,m22}}) and 2D translations (x3,y3) that bring newvertices as close as possible to the threeDtemplate stored in memory.**

```
Manipulate[
 x1 = orthoproject[threeDtemplate];
 M2 = {{m11, m12}, {m21, m22}};
x2 = (M2.#1 &) /@ orthoproject[newvertices];
 x2b = # + {x3, y3} & /@ x2;
GraphicsRow[{Graphics[Line[x1], PlotRange → {{-.5, 1.5}, {-.5, 1.5}}],
   Graphics[{Line[x1], Line[x2b]}, PlotRange → {{-.5, 1.5}, {-.5, 1.5}}]},
  ImageSize → Small],
 {{m11, 1}, -2, 2}, {{m12, 2}, -2, 2}, {{m21, 2}, -2, 2},
 {{m22, 0}, -2, 2}, {x3, -1, 1}, {y3, -1, 1}]
```



Compute closest least squares affine match with translation

```
aff = {{aa, bb}, {cc, dd}}; tra = {ff, gg};
errorsum :=
  Apply[Plus,
   Flatten[
    (# + tra & /@ (aff.#1 &) /@ orthoproject[newvertices] -
       orthoproject[threeDtemplate]) ^2]];
temp = FindMinimum[errorsum, {aa, .8}, {bb, .2}, {cc, .16}, {dd, .8},
   {ff, 0.0}, {gg, 0.0}, MaxIterations -> 200];
minvals = Take[temp, -1][[1]]; minerr = Take[temp, 1][[1]];
naff = aff /. minvals; ntra = tra /. minvals;
minerr
```

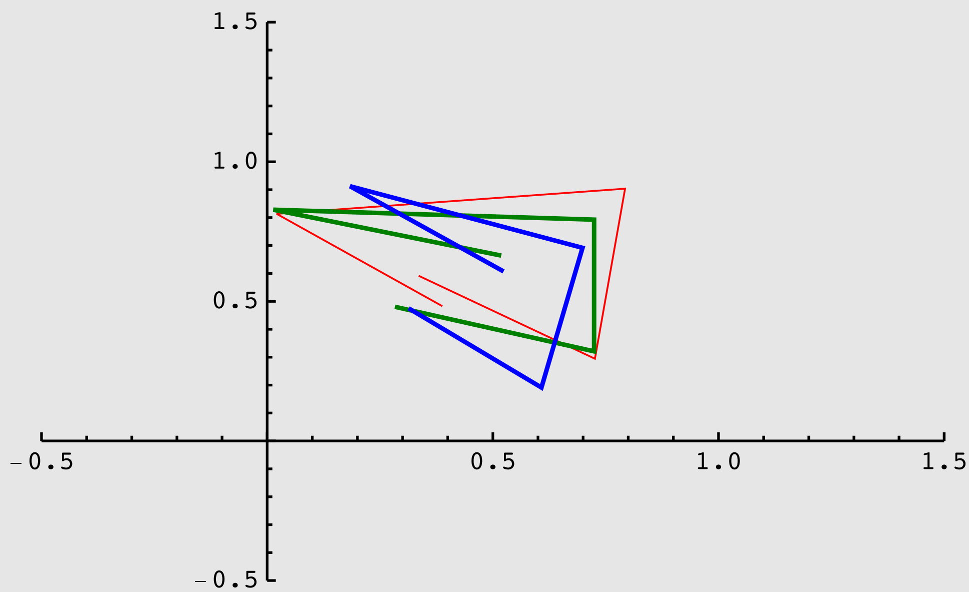
```
0.0818952
```

Check match with estimated view

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

■ **Plot first original view, new view and the affine estimate of the first from the new**

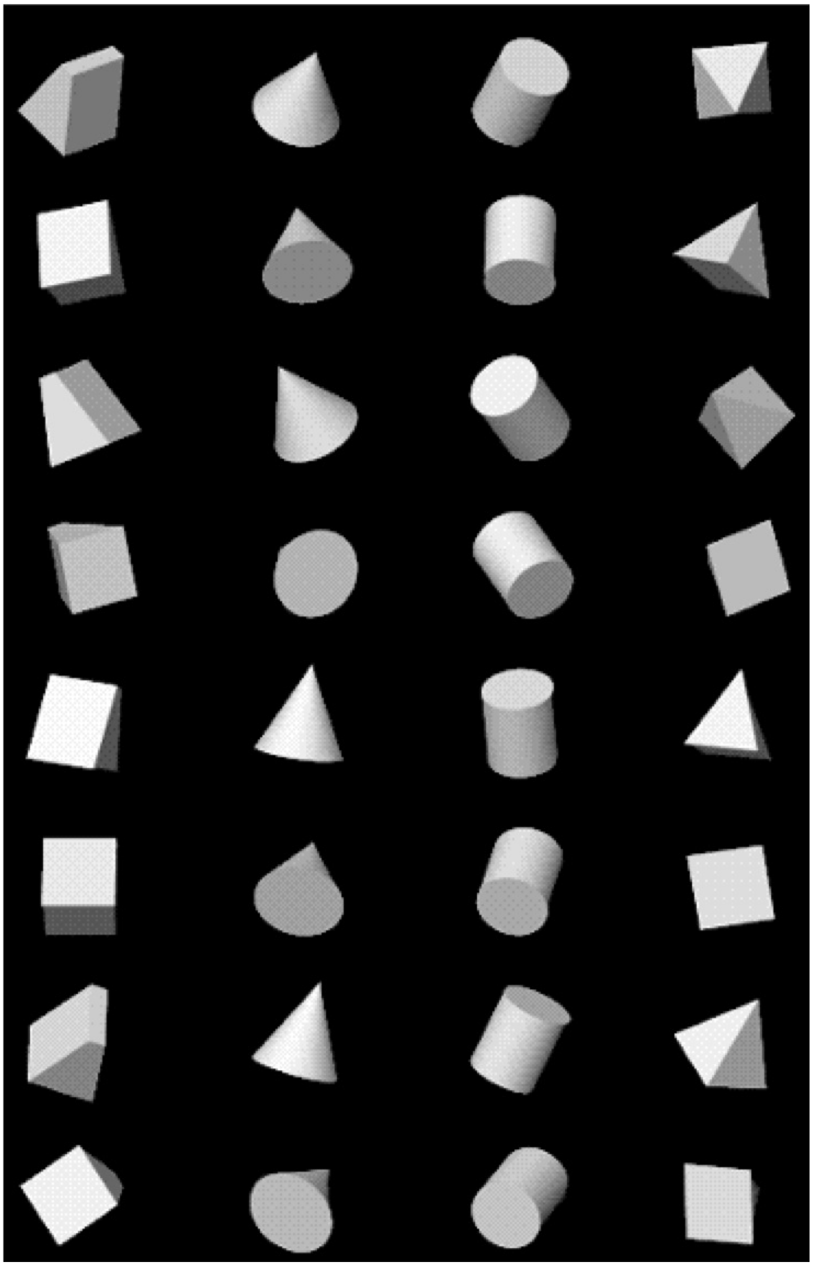
```
evg = ListPlot[{orthoproject[threeDtemplate], Transpose[estim],
   orthoproject[newvertices]}, Joined -> True,
  PlotStyle -> {{Thickness[0.002], RGBColor[1, 0, 0]},
                {Thickness[0.005], RGBColor[0, .5, 0]},
                {Thickness[0.005], RGBColor[0, 0, 1]}},
   PlotRange -> {{-.5,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



Liu & Kersten (1998) compared human recognition performance with 2D affine observers. The targets were paper-clip like objects as above, except thicker with some shading. Human performance was somewhat better than the affine observer, suggesting that people can incorporate additional 3D information, perhaps from the shading/occlusion information, together with a "smarter" model.

Ideal observer for the "snap shot" model of visual recognition: Discounting views

Tjan et al. describe an application to object recognition in noise.



Eight views of four objects. (See Tjan B., Braje, W., Legge, G.E. & Kersten, D. (1995) Human efficiency for recognizing 3-D objects in luminance noise.Vision Research, 35, 3053-3069.)

Let **X** = the vector describing the image data. Let O_i represent object i, where i = 1 to N. Suppose that O_i is represented in memory by M "snap shots" of each object, call them views (or templates) V_{ij}, where j = 1, M.

$$p(O_i \mid X) = \sum_{j=1}^{M} p\left(V_{ij} \mid X\right)$$

$$= \sum_{j=1}^{M} \frac{p(X \mid V_{ij})\, p(V_{ij})}{p(X)}$$

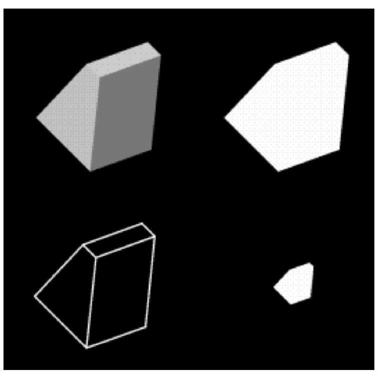
Given image data, Ideal observer chooses i that maximizes the posterior $p(O_i \mid X)$. If we assume that the p(X) is uniform, the optimal strategy is equivalent to choosing i that maximizes:

$$L(i) = \sum_{j=1}^{M} p(X \mid V_{ij})\, p(V_{ij})$$

If we assume i.i.d additive gaussian noise (as we did for the signal-known-exactly detection ideal), then

$$p(X \mid V_{ij}) = \frac{1}{\left(\sigma \sqrt{2\pi}\right)^{p}} \exp\left(-\frac{1}{2\sigma^2} \|X - V_{ij}\|^2\right)$$

where p is the number of pixels in the image.



Tjan et al. showed that size, spatial uncertainty and detection efficiency played large roles in accounting for human object recognition efficiency. Interestingly, highest recognition efficiencies (~7.8%) were found for small silhouettes of the objects. (The small silhouettes were 0.7 deg, vs. 2.4 deg for the large silhouettes).

Appendix: Neuropsychological and neurophysiological studies

Neuropsychological Studies

Category-specific breakdowns

Inferomedial occipito-temporal region, (right hemi), fusiform and lingual gyri--> prosopagnosia. Can recognize other objects (even with comparable structural complexity), and can recognize a face as a face, and can name its parts.

...but is it a problem with individuation in a class? Evidence suggesting prosopagnosics have a problem distinguishing fruits, playing cards, autos, etc.. Bird-watcher lost ability. Farmer couldn't identify his cows.

Damasio's patients could recognize horses, owls, elephants, but had problems with dollar sign, British pound sign, musical clef. --> perhaps a problem with inter-category discriminations (subordinate-level), rather than complexity per se.

Corroboration--patient with car agnosia could still identify ambulance and fire engine (distinct entry point attributes)

BUT, propospagnosia does seem sometimes to occur without any of the subordinate-level deficit. Patients impaired for living, but not non-living things.

<<20 questions and recognition>>

Summary: Two types of visual memory:

recognition that involves representing and distinguishing prototypes

<<Different protypes in different IT hypercolumns?>>

recognition that involves distinguishing deviations between members with the same prototype (inferomedial occipito-temporal)

<<processing within hypercolumn?>>

Deficits in recognizing facial expressions

Dissociation between face recognition and recognizing facial expressions.

Some proso's can't recognize an individual face, but can recognize the expression.

Damasio reports bilateral amygdala lesion patient could recognize individual faces, but did not do well with expressions of happiness, surprise, fear, anger, etc.. Monkeys too (Weiskrantz, 1956)

Metamorphopsia with faces. Another patient experiences metamorphopsia with objects other than faces.

Visuomotor

DF

Electrophysiological Studies

V1-> V2 -> V4 -> IT-> TEO (PIT) -> TE

not strictly serial

V2, V3, V4 , corpus callosum-> IT

TE, TEO connected to thalamus, hypothalamus,...

Object information might even skip IT and go to limbic structures or striatum...

>abstract categorizations (with high cue validity) perhaps possible even with damage to TE

■ *Physiological properties of IT neurons*

Physiological properties of IT neurons

Gross. IT as last exclusive visual area.

Posterior TEO, cells similar to V4, visuotopic, repres. contralateral vis. field, rf.s larger than V4. (small as 1.5 - 2.5 deg)

anterior TE, complex stimuli required. TE not visutopic, large ipsi, contra or bilat. rfs.

30 to 50 deg rfs.

Cells often respond more vigorously to Fovea stimulation

Shape selectivity (some in V4), lots in IT. natural objects, walsh functions, faces, hands.

Invariance? Rare to find size or position constancy--but selectivity falls off slowly over size and position. Thus in this sense roughly 50% of cells show size and position invariance.

Cue invariant--motion, texture or luminance defined shape boundaries. BUT, contrast polarity sensitive. >>shape from shading?

Two mechanisms? 1) prototypes of objects that can be decomposed into parts.

parts important.

2) holistic, configurational. Part features not useful for discrimination, but whole is.

■ *Combination encoding*

Tanaka & modules for similar shapes, columnar organization. \

>1300 prototype modules?? RBC?

Sufficient for representing an exemplar of a category? Or when holistic information is required?

L&S suggest combination encoding not used for holistic representation. Evidence: Many celss in TE and STS code overall shape of biologically important objects--not features or parts. Novel wire objects too.

■ *Selectivity for biologically important stimuli*

Face cells - TEa, TEm, STS, amygdala, inf. convexity of prefrontal cortex.

Some cells like features (e.g. eyes). Other like the whole face, or face-view, or even highly selective for face-gaze angle, head direction, and body posture.

Face cells, invariant over size and position, less so over orientation--upright preferred.

Face identity cells in IT,

but facial expression, gaze direction , and vantage point in STS

PET, posterior fusiform gyrus for face matching, gender disc.

mid-fusiform for unique face

IT cells for whole human body, mostly viewer centered cells. 20% holistic

■ *Configurational selectivity for novel objects*

L et al., and L&S's work. on wires, etc.

ant. medial temporal sulcus

view-selective "blurred templates"

enantiomorphic views undistinguished

many showed broad size tuning

Action-related

MT -> parietal MST, FST, LIP, 7,

LIP cells sensitive to grasp shape of hand

Compute closest least squares affine match without translation

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

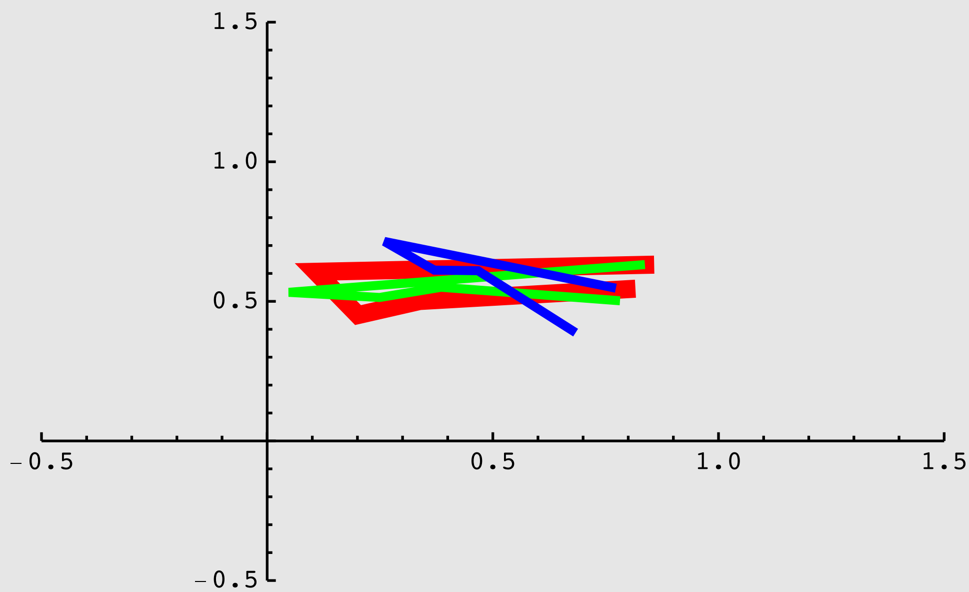
```
{{1.40376, -0.443053}, {0.391479, 0.602826}}
```

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 = ListPlot[{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}}]
```



Test set of newvertices and threeDtemplate

(*threeDtemplate=

0.23981762582649485`	0.14312418380466885`	0.03003120544761813`
0.2624091279705781`	0.4565009537332048`	0.1221875974954246`
0.019392922865028396`	0.016530310373452352`	0.5906147114395374`
0.06481020981636326`	0.6548152420848915`	0.40459291550719`
0.642248220665317б`	0.7719461816974882`	0.22053936016974654`

*)

(*newvertices=

0.2215818503538964`	0.09974436717830631`	−0.09854211812818665`
0.26452283137288907`	0.3891455135818127`	−0.16199300398045482`
0.18952784909991596`	0.25183584120022917`	0.45991480775746235`
0.17676551774440416`	0.7093221832702079`	0.026256226849368618`
0.5640697249874365`	0.5756717475918378`	−0.28280208011255054`

*)

References

Recognition

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

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.

Clark, J. J., & Yuille, A. L. (1990). Data Fusion for Sensory Information Processing. Boston: Kluwer Academic Publishers.

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

Field, D. J., Hayes, A., & Hess, R. F. (1993). Contour integration by the human visual system: evidence for a local "association field". Vision Research, 33, 173-193.

Kersten, D. J. (1991). Transparency and the Cooperative Computation of Scene Attributes. In M. Landy & A. Movshon (Eds.), Computational Models of Visual Processing, (pp. 209-228). Cambridge, Massachusetts: M.I.T. Press.

Kersten, D. & Madarasmi, S. (1995). The Visual Perception of Surfaces, their Properties, and Relationships. In I. J. Cox, P. Hansen, & B. Julesz (Ed.), *Partitioning Data Sets: With applications to psychology, vision and target tracking* . (pp. 373-389). American Mathematical Society.

Kersten, D. (1999). High-level vision as statistical inference. In M. S. Gazzaniga (Ed.), *The New Cognitive Neurosciences -- 2nd Edition* (pp. 353-363). Cambridge, MA: MIT Press.

Kersten, D., & Schrater, P. W. (2000). Pattern Inference Theory: A Probabilistic Approach to Vision. In R. Mausfeld & D. Heyer (Eds.), *Perception and the Physical World*. Chichester: John Wiley & Sons, Ltd.

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.

Mohan, R. (1989). Perceptual organization for computer vision (IRIS 254): University of Southern California.

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

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

Sharon, E., Galun, M., Sharon, D., Basri, R., & Brandt, A. (2006). Hierarchy and adaptivity in segmenting visual scenes. *Nature*, *442*(7104), 810–813. doi:10.1038/nature04977

Shashua, A., & Ullman, S. (1988,). Structural Saliency: The detection of globally salient structures using a locally connected network. Paper presented at the 2nd International Conference on Computer Vision, Washington, D.C.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *22*(8), 888–905.

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.

Troje, N. F., & Kersten, D. (1999). Viewpoint dependent recognition of familiar faces. *Perception, 28*(4), 483 - 487.

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

Ullman, S., Vidal-Naquet, M., & Sali, E. (2002). Visual features of intermediate complexity and their use in classification. Nat Neurosci, 5(7), 682-687.

Vetter, T., Poggio, T., & Bülthoff, H. H. (1994). The importance of symmetry and virtual views in three-dimensional object recognition. Current Biology, 4(1), 18-23.