Computational Vision
U. Minn. Psy 5036
Daniel Kersten
Lecture 23: Overview of high-level visual tasks. Object Recognition

Initialize

■ Spell check off

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

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

Outline

Last time

Perception as "puzzle solving"--cooperative computation of scene attributes given image cues

Today

Tasks

Object recognition

Role of geometric modeling in theories of object recognition

High-level vision, visual tasks

Intermediate-level vision

Generic, global organizational processes

Domain overlap, occlusion

Surface grouping, selection

Gestalt principles

Cue integration

Cooperative computation

Attention

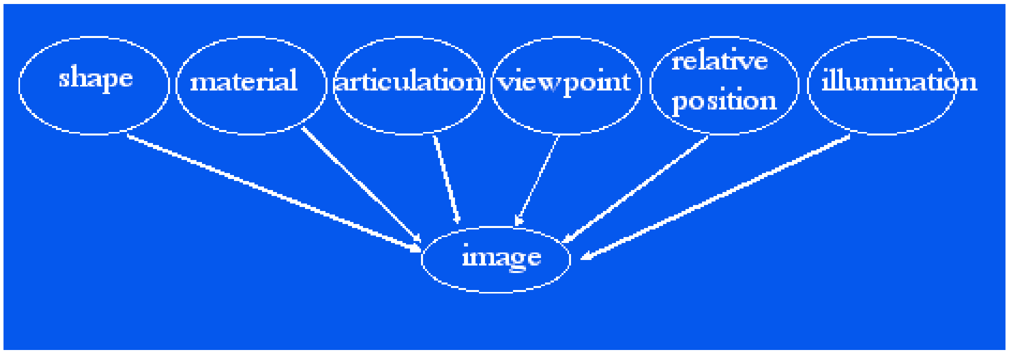
High-level vision

Functional tasks

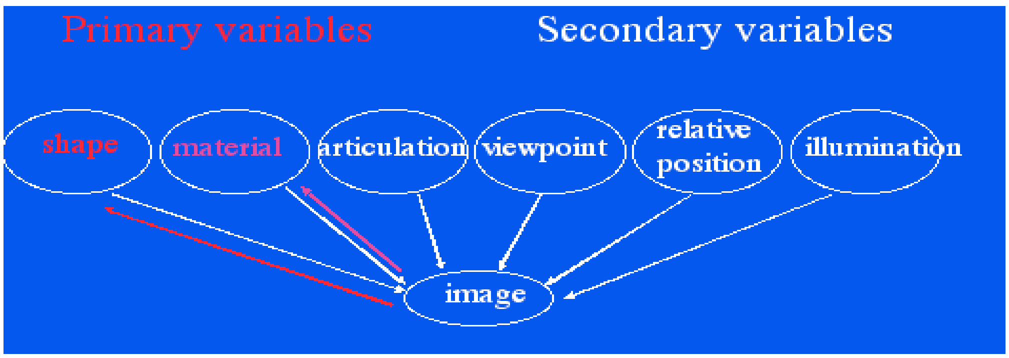
- Object recognition
    - entry-level, subordinate-level
- Object-object relations
    - Scene recognition
    - Spatial layout
- Viewer-object relations
    - Object manipulation
        - reach & grasp
    - Heading

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

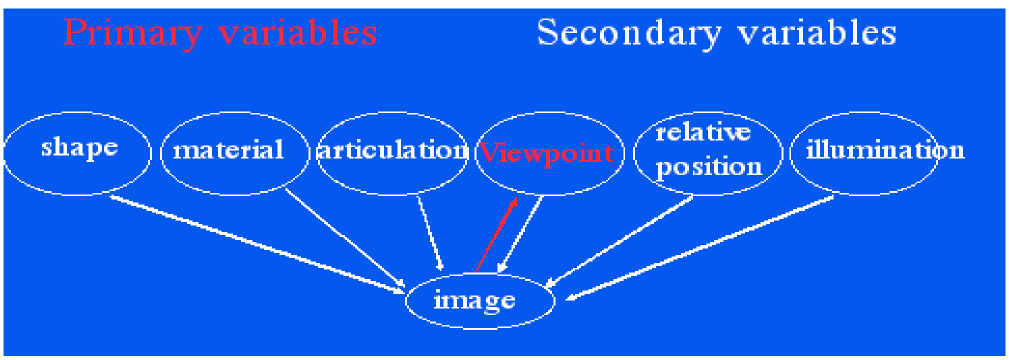
I=f(shape, material, articulation,viewpoint,relative position, illumination)



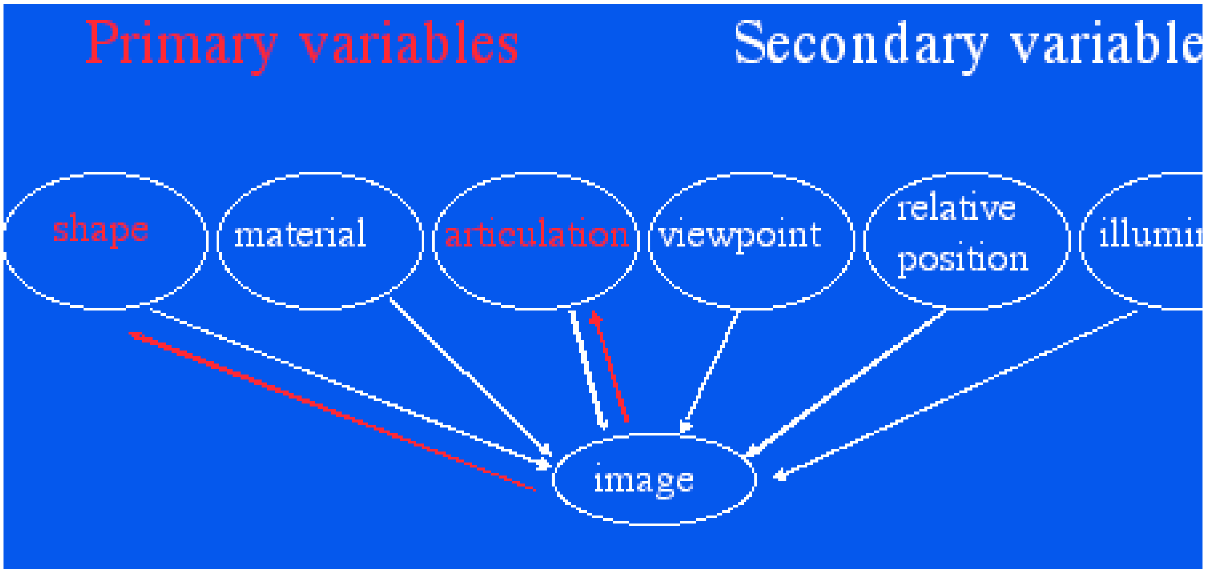
■ **Task: Object Recognition**



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



■ **Task: grasp**



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

Bayes framework for high-level vision

Task specifies what scene variables to integrate out:

$$p(S_{prim} \mid I) = \int p(S_{prim}, S_{sec} \mid I) dS_{sec}$$

Bayes' rule factors posterior probability into likelihood and prior terms:

$$p(S_{prim}, S_{sec} \mid I) \propto p(I \mid S_{prim}, S_{sec}) p(S_{prim}, S_{sec})$$

■ **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

Object recognition

Sources of image variation

We'll work from lower to higher levels of object abstraction

■ Variation within subordinate-level category

(subordinate level, e.g. mallard, Doberman, Braeburn)

- 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
- background (segmentation)
  - bounding contour/edge variation
- occlusion (segmentation)

■ Variation within basic-level category

(e.g. duck, dog, chair, apple)

"entry-level", "basic-level"

structural relation invariance?

■ Variation across super-ordinate category

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

more cognitive than perceptual, non-pictorial

■ Variation across context

ball on tennis court vs. billiard table

Basic-level vs. subordinate-level

Psychological (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

■ 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?

Consider viewpoint

Features invariant to viewpoint change?

Material (surface color, texture)

2D image features correlated with 3D Shape

Object ensemble is important

E.g only red object amidst others – no need to process its shape

Context is important

Small red thing flying past the window.

High "cue validity" for Cardinal

Overview of processes that seem to be necessary -- i.e. what we think we know

Getting a good image representation

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.

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

edge detection is really noisy, so what are these image "features"?

maybe although noisy, edges are 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

Mohan (1988); Zhu (1999); Geiger – figure/ground talk

symmetry (Vetter et al., 1994)

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

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

There is no single local cue to edge identity

"intrinsic images" of Barrow and Tenenbaum to extract

Clark & Yuille (1990); Knill &Kersten, Kersten (1991); Kersten and Madarasmi (1995).

Problem: Still no bottom-up procedure for perfect segmentation or edge-parsing.

Solutions?: Domain-specific processing; class-specific recognition.

Classification -> edge detection. E.g dalmation dog demonstration

Storing, matching information about objects

How does the brain store information about 3D objects?

Structural description: high-level features or parts plus relations

Image-based: low-level features plus 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?

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)

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

partial independence of viewpoint

View dependent, alignment methods

E.g. Ullman, alignment of pictorial descriptions

Selection

Segmentation

Image description

-Alignment

-Matching

TEST: S = F(M)?

->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?

How sophisticated are the 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. 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.

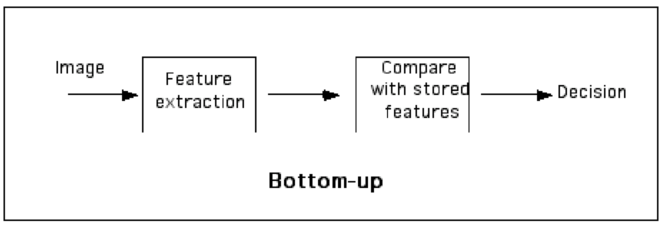
Liu et al. (1995) were able to exclude models of the first class in a simple 3D classification task using ideal observer analysis. The ideal observer technique was developed in the context of our studies of quantum efficiency in early vision.

Review of types of image modeling (from Lecture 7)

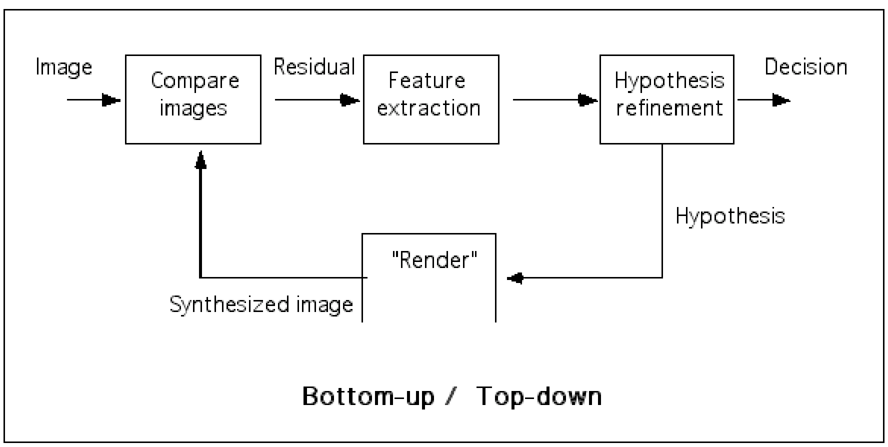
Generative models for images: rationale

■ Characterize the knowledge required for inference

Feedforward procedures:



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



- **Easier to characterize information flow: Mapping is is many-to-one**

- **Provides tools to specify the independent variables in psychophysics, and vision models**

- **Two basic concepts: Photometric & geometric variation**

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

Scene-based modeling: Computer graphics models

- **Scenes, Variable classes : extended surfaces, solid objects, diffuse "stuff"/particles, lights, and camera (eye).**

- **Objects & surfaces: Shape, Articulations, Material & texture**

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

- **Viewpoint/Camera**

Projection geometry, homogeneous coordinates:

perspective, orthographic

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

■ Non-linear intensity-based

e.g. Contrast normalization

■ Linear geometry-based

Affine:

rigid translations, rotations, scale and shear

Application: viewpoint variation

■ Non-linear geometry-based

Morphs

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

finding the "average" face

Applications of small-scale geometry

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

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

What is the relationship of parts of objects to each other?

Extrinsic vs. intrinsic geometrical descriptions

Role in object recognition

Applications of large scale geometry

Scene geometry--Spatial layout, large-scale surface structure

Where are objects relative to the viewer?

Where are they relative to each other?

Role in spatial layout, structure from motion, and heading-> in final lecture

Modeling geometric variation: 3D scene-based modeling

Rotations

```
<< Geometry`Rotations`
```

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. (See *Mathematica* Help menu)

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

$$\begin{pmatrix} \cos(\phi)\cos(\psi)-\cos(\theta)\sin(\phi)\sin(\psi) & \cos(\theta)\cos(\psi)\sin(\phi)+\cos(\phi)\sin(\psi) & \sin(\theta)\sin(\phi) \\ -\cos(\psi)\sin(\phi)-\cos(\theta)\cos(\phi)\sin(\psi) & \cos(\theta)\cos(\phi)\cos(\psi)-\sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta) \\ \sin(\theta)\sin(\psi) & -\cos(\psi)\sin(\theta) & \cos(\theta) \end{pmatrix}$$

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

$$\begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

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

$$\begin{pmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

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 homogeneous 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: Foley, J., van Dam, A., Feiner, S., & Hughes, J. (1990). {Computer Graphics Principles and Practice}, (2nd ed.). Reading, Massachusetts: Addison-Wesley Publishing Company.

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.

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

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

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:**

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

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

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

$$\begin{pmatrix} cos\theta & sin\theta & 0 & 0 \\ -sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & sin\theta & 0 \\ 0 & -sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} cos\theta & 0 & -sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

■ 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:

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

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

$$\begin{pmatrix} x', & y', & z', & 1 \end{pmatrix} = \begin{pmatrix} \frac{xd}{d-z,} & \frac{yd}{d-z,} & 0, & 1 \end{pmatrix}$$

The matrix for orthographic projection has d-> infinity.

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.

Packages--homogeneous.m

Homogeneous.m

<<"Geometry`Homogeneous`"

<<"Graphics`Polyhedra`"

<<Graphics`Graphics3D`

<<Graphics`MultipleListPlot`

```
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}};
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`

XRotationMatrix[phi] gives the matrix for rotation about x−axis by phi degrees in radians

YRotationMatrix[phi] gives the matrix for rotation about y−axis by phi degrees in radians

ZRotationMatrix[phi] gives the matrix for rotation about z−axis by phi degrees in radians

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[x,y,z] gives the matrix to translate coordinates by x,y,z.

ThreeDToHomogeneous[sx,sy,sz] converts 3D coordinates to 4D homogeneous coordinates.

HomogeneousToThreeD[4Dvector] converts 4D homogeneous coordinates to 3D coordinates.

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[vector] projects vector on to the x−y plane.

Geometry`Homogeneous`private`

Geometry`Homogeneous`private`

Example: transforming, projecting a 3D object

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

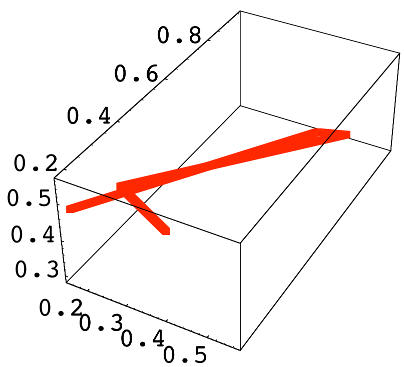
■ Define 3D target object - Wire with randomly positioned vertices

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

```
MatrixForm[threeDtemplate]
```

$$\begin{pmatrix} 0.398115 & 0.183863 & 0.513307 \\ 0.272483 & 0.21363 & 0.560933 \\ 0.567738 & 0.686339 & 0.498648 \\ 0.38164 & 0.94558 & 0.28734 \\ 0.154773 & 0.18974 & 0.473253 \end{pmatrix}$$

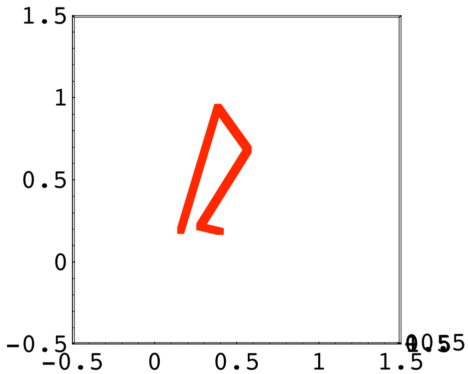
```
ScatterPlot3D[threeDtemplate, PlotJoined -> True,
  PlotStyle -> {{Thickness[0.02], RGBColor[1, 0, 0]}}];
```



First view

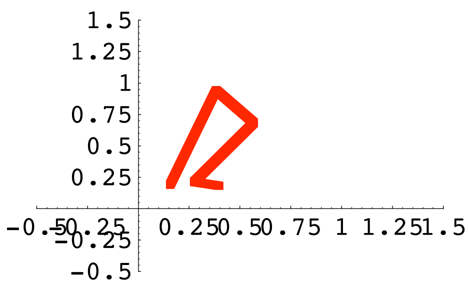
View from along Z-direction

```
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;
```

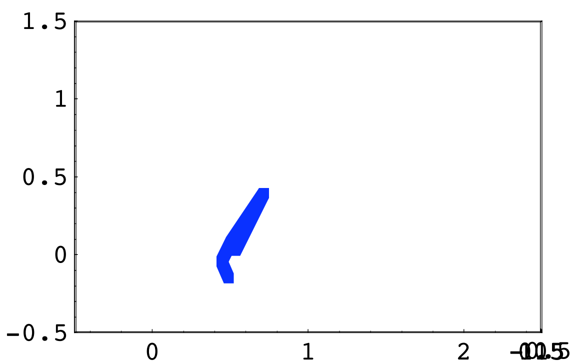
```
homovertices = Transpose[Map[ThreeDToHomogeneous, threeDtemplate]];
newtransformMatrix = TranslateMatrix[1, 0, 0].XRotationMatrix[N[π/4]].
   YRotationMatrix[N[-π/2]].ScaleMatrix[swidth, sheight, slength];
```

```
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}}];
```



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

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

If one projects a 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.

Affine transformation preserve parallel lines.

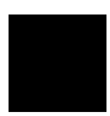
We know that rotations, scale and shear transformations will preserve parallel lines. So will translations. It is not immediately apparent, that any matrix operation is an affine transformation, although one has to remember that translations are not represented by matrix operations unless one goes to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

■ Rotate square

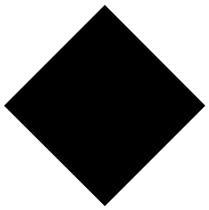
```
M[theta_] := {{Cos[theta], Sin[theta]}, {-Sin[theta], Cos[theta]}};
```

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

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



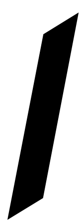
```
Show[Graphics[Polygon[(M[π/4].#1 &) /@ square]], AspectRatio → 1,
  PlotRange → {{-2, 2}, {-2, 2}}];
```



■ Random matrix operation on square

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

$$\begin{pmatrix} 0.130709 & 0.129477 \\ 0.674555 & 0.0784184 \end{pmatrix}$$



- Graphics -

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
```

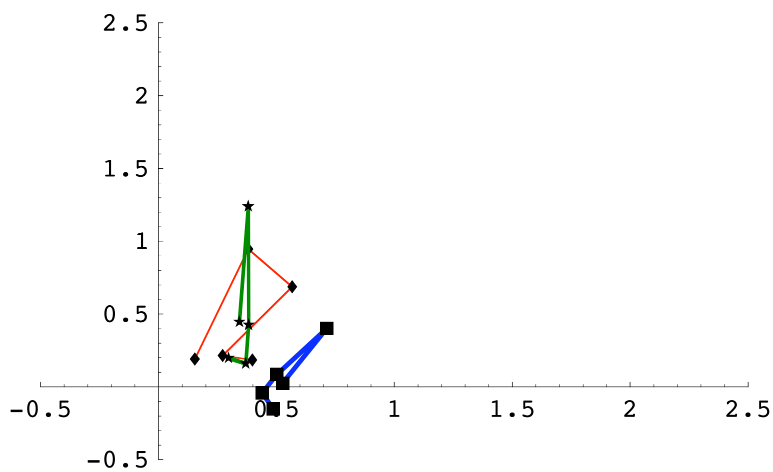
```
0.272389
```

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 = 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, 2.5}, {-.5, 2.5}}];
```



Compute closest least squares affine match without translation

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

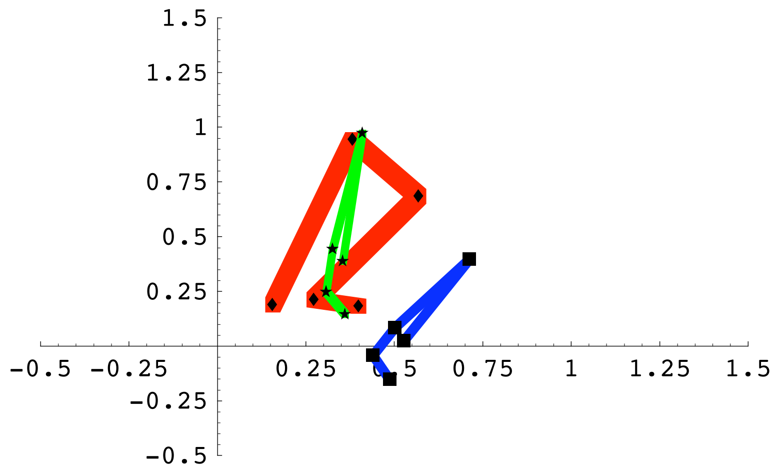
$$\begin{pmatrix} 0.680998 & -0.190259 \\ 0.680998 & 1.22395 \end{pmatrix}$$

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}}];
```



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.

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 othe 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

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.

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.

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

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.

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.

© 2000, 2004, 2006 Daniel Kersten, Computational Vision Lab, Department of Psychology, University of Minnesota.
kersten.org