\[ f^m = \sum_{j=1}^{n} g^m_j \cdot t^m_j (f^{m-1}) \]

\[ b^m = \begin{cases} \\
\sum_{j=1}^{m} g^m_j \cdot t^m_j (b^{m+1}) & \text{for } m = 1 \ldots L \\
\sum_k z \langle w_k, f^L \rangle \cdot w_k & \text{for } m = L + 1 \\
\end{cases} \]

\[ q^m_i = \langle t^m_i (f^{m-1}), b^{m+1} \rangle \]

\[ g^m_i = \max \left( 0, g^m_i - k_1 \cdot \left( 1 - \frac{q^m_i}{\max{q^m}} \right)^{k_2} \right) \]
An inverse method for transformation discovery

Map-seeking circuits or algorithm...
In machine vision terms, segmentation and recognition of 3D object in arbitrary pose is a form of the “correspondence problem.”
In map-seeking terms...
image and model are related by an aggregate transformation $T$

$$P(o) = T(m_i) \quad m_i \in M$$

$T$ may be decomposed into a sequence of transformations

$$T = t_L \circ t_{L-1} \cdots \circ t_1$$
Define **correspondence** \( c \) between vectors \( r \) and \( w \) through a composition of \( L \) transformations \( t^1_{j_1}, t^2_{j_2}, \cdots, t^L_{j_L} \) where \( t^l_{j_l} \in t^1_{l}, t^2_{l}, \cdots, t^n_l \)

\[
c(j) = \left\langle \bigcirc_{i=1}^L t^i_{j_i}(r), w \right\rangle
\]

where the composition operator is defined

\[
\bigcirc_{i=\emptyset,1}^L t^l_{j_i}(r) = \begin{cases} t^L_{j_L} \circ t^{L-1}_{j_{L-1}} \cdots \circ t^1_{j_1}(r) & l = 1 \cdots L \\ r & l = \emptyset \end{cases}
\]
\[ c(j) = \left\langle \bigcirc_{i=1}^{L} t_{j_i}^i (r), w \right\rangle \]

If we treat this as a discrete problem, let et \( \mathbf{C} \) be an \( L \) dimensional matrix of values of \( c(j) \) whose dimensions are \( n_1 \ldots n_L \).

The problem, then is to find

\[ \mathbf{x} = \text{arg max} \ c(j) \]

That is, the indices \( \mathbf{x} \) specify the sequence of transformations that best correspondence between vectors \( \mathbf{r} \) and \( \mathbf{w} \).
Why is this problem not amenable to solution by brute force search?

Space of aggregate transformations is huge.

e.g. with reasonable discretization, transformations $>10^{10}$
How has this problem been tackled?

1) Find feature correspondences between image and model, calculate transformation from geometry of feature locations

2) Limit space of considered transformations, e.g. using “Bayesian” guesses at “probable” transformations, and then gradient descent or ascent

what’s a feature here?
**Map-seeking method** provides a means to search the entire space with tractable resources (in time or material) and with high probability of finding

\[ x = \arg \max_i c(j) = \arg \max_i \left( \bigoplus_{i=1}^{L} t_{j_i} (r), w \right) \]

Requirements for effective application of map-seeking method (MSC, for map-seeking circuit)

1) **decomposability** of transformation space
2) data representation satisfies **ordering property of superpositions** (OPS)
decomposability alone is not enough...

one-at-a-time
- translation
  - 120 × 120

one-at-a-time
- rotation in plane
  - -30° to +30°

one-at-a-time
- scale/aspect
  - one octave

one-at-a-time
- 3D projection
  - 180°h × 90°v

scale \sim R \left( \prod_{l=1}^{L} n_l \right)

goodness of match

\[ c(j) = \left< \bigodot_{i=1}^{L} t_{ji}^i \left( r \right), w \right> \]
MSC depends on...

**Ordering Property of Superpositions**

A superposition $s$ is formed from a set of sparse vectors $v_i \in V$, 

$$s = \sum v_i$$

Vector $v_k$ is not part of the set from which superposition is formed, $v_k \notin V$, 

Then, 

$$P(v_i \cdot s > v_k \cdot s) > 0.5$$

or

$$P_{\text{correct}} > P_{\text{incorrect}}$$
also true for two superposition vectors sharing a common contributor

Superpositions \( r = \sum u_i \), \( s = \sum v_j \) and \( s' = \sum v_k \)
from three sets of sparse vectors \( u_i \in R \), \( v_j \in S \) and \( v_k \in S' \)
where \( R \cap S = \emptyset \) and \( R \cap S' = v_q \)

Then, from the superposition ordering property
\[
P_{\text{correct}} \left( r \cdot s' > r \cdot s \right) > 0.5
\]
Case 1

Let \( v_i, v_j, v_k \in S \), where \( v_i \) and \( v_k \) are \( n \)-length vectors with \( m \) ones and \( n-m \) zeros.

\[ \sum_{i \in v_S} v_i = s_1 \]

If \( v_k \neq v_j \),

\[ (v_i, s_1) = m > (v_k, s_1) = 0 \]

\[ (v_i, s_2) = m + (v_i, v_j) > (v_k, s_2) = 0 + (v_k, v_j) \]

\[ s_1 = v_i \]

\[ s_2 = v_i + v_j \]
\[ q_x = v_x \cdot s \]
\[ q_y = v_y \cdot s \]

where \( s = v_x + \sum_{i=1}^{c} v_i \)

overlap:
\[ L = v_x \cdot v_y \]
OPS lower bound $0.5 < P(q_x - q_y > 0) < 1$
the *ordering property of superpositions (OPS)* allows...

one-at-a-time  *all-at-a-time*...

rotation in plane  -30° to +30°

scale/aspect  one octave

3D projection  180° h × 90° v

---

\[ R\left( \sum_{l=1}^{L} n_l \right) < \text{scale} < R\left( \prod_{l=1}^{L} n_l \right) \]

---

\[
q_j^m = \left\langle \sum_{l=m+1}^{L} g_l \cdot t_l \right\rangle \circ_{j} \left( \sum_{i=1}^{m-1} \left( g_i \cdot t_i \right) \right) (r), w \right\rangle
\]
or equivalently...

\[ q_j^m = \left\langle t_j^m \circ \left( \sum_{l=1}^{m-1} g_l^i \cdot t_l^i \right)(r), \left( \sum_{l=L}^{m+1} g_l^i \cdot t_l^i \right)(w) \right\rangle \]
Parallelize.....

- 3D projection
- rotation in plane
- scale/aspect
- 3D projection

- translation
- 120 × 120
- -30° to +30°
- one octave
- 180°h × 90°v

- edge filtered to sparsify

- indefinite fwd projection 2D to 3D
- match in 3D domain
\[ f_m = \sum_{j=1}^{m} g^m_j \cdot t^m_j (f_{m-1}^j) \]
\[ b_m = \sum_{j=1}^{m} g^m_j \cdot t^m_j (b_{m-1}) \]
\[ g^m = K(g^m_q, q^m) \]
\[ b_{L+1} = w^L \cdot w_k \]

The competition can be non-linear match operation.
a minimalist example...

\[ r = \begin{array}{c}
\Lambda \\
\end{array} \quad w = \begin{array}{c}
\Lambda \\
\end{array} \]

what transformations map \( r \) onto \( w \)?

\[ r \xrightarrow{t_x(r)} t_y \circ t_x(r) \xrightarrow{t_{rot} \circ t_y \circ t_x(r) = w} \]
modify q to reflect matches, then adjust g

BKWD

memory image

FWD

tr x

tr y

rot

input image
new g values modify f and b

FWD

BKWD

memory image

input image

tr x

tr y

rot
modify q to reflect matches, then adjust g
new g values modify f and b - final state
\[
\begin{align*}
g_1 &= (0, 1, 0) \\
g_2 &= (1, 0, 0) \\
g_3 &= (1, 0, 1) \\
g_4 &= (0, 0, 1)
\end{align*}
\]
Convergence by competition, in action...

(mapping coefficients indices $i$)

(Example from 3D layer 4 includes azimuth and elevation)
input image has arbitrary view of target and distractors

scene segmentation

edge filtered

3D projection
rotation in plane
scale/aspect
translation
composed transformations

180° h × 90° v
-30° to +30°
one octave
120 × 120
fwd     inverse
converges to determine 3D pose

two different views of target

backward path "edge plausibility states"
Geometry Mapping Parameter Ranges (Single convergence):
Scaling 0.7 to 1.4
Azimuth: -180 to +180 degrees
View Depression: 10 to 60 degrees
Axial Rotation: -5 to +5 degrees
Geometry Mapping Parameter Ranges (Single convergence):
Scaling 0.7 to 1.4
Azimuth: -180 to +180 degrees
View Depression: 40 to 70 degrees
Axial Rotation: -5 to +5 degrees
Tests with Fort Carson EO Imagery

Transformation parameters:
Translation: 120 x 120 pixels (out of 200 x 200 input image)
In plane rotation: -10 to +10 degrees by 1 degree increments
Scaling: one octave, 0.7 to 1.4 by increments of 0.025
3D projection: azimuth –90 to +90 degrees by 5 degree increments,
camera depression 0 to 30 degrees by 2.5 degree increments.
(note actual azimuth range in all tests -85 to +85 degrees, since –90 to +90
is functionally redundant at mapping resolution)
Convergence maximum 25 iterations.
M60 and distractors, input image is blurred to about 10 cycles on target
M60 and distractors, input image is blurred to about 12 cycles on target
Not just edges: interpretation from shadows...

model

pose determined by MSC

input image

source image
Stabilization of retinal motion induced drift and jitter can be done in real time (30 fps)

AOSLO = Adaptive Optics Scanning Laser Ophthalmoscope

Cone Photoreceptors near fovea

OC left eye

AOSLO Houston 02/28/2003

source

stabilized

frames 512h x 480 v

CfAO funded research with Curt Vogel, Austin Roorda (UC Berkeley), Al Parker
Image enhancement by co-adding registered frames
Adding Registered Frames to DeSpeckle and Denoise

motion track frame to frame

Filtered Motion Tracks for QC - dropped motion with corr<.3
Kinematic transformation (K) MSCs
One Basis is Skeletal Inverse Kinematics
The points in space reachable by the $i$th segment from $p_i$ is

$$S_i = \bigcup_j f_{i,j}(p_i) = F_i(p_i)$$

We will use the convention that the end effector is segment 1. For a three segment limb the points in space, $S_1$, reachable by the third segment from a given point $p_3$, at the root of the limb, is the composition of the mappings

- $S_1 = F_1(S_2)$
- $S_2 = F_2(S_3)$
- $S_3 = F_3(p_3)$

or

$$S_1 = F_1 \circ F_2 \circ F_3(p_3)$$
MSC Solution of Inverse Kinematics with Obstacles

<table>
<thead>
<tr>
<th>armx</th>
<th>armx</th>
<th>targx</th>
<th>targy</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>40</td>
<td>64</td>
<td>58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>segment</th>
<th>θ_{min}</th>
<th>θ_{max}</th>
<th>θ_{incr}</th>
<th>radius</th>
<th>θ_{relmin}</th>
<th>θ_{relmax}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>-175.0</td>
<td>180.0</td>
<td>5.0</td>
<td>9</td>
<td>-175.0</td>
<td>180.0</td>
</tr>
<tr>
<td>2:</td>
<td>-175.0</td>
<td>180.0</td>
<td>5.0</td>
<td>23</td>
<td>-175.0</td>
<td>180.0</td>
</tr>
<tr>
<td>3:</td>
<td>-80.0</td>
<td>80.0</td>
<td>5.0</td>
<td>31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Case 1: Obstacle at \(x, y\) =<45 to 55, 30 to 50>

*Figure 4-7. IK test including obstacles and constraints, case 1.*
Route planning is a version of kinematic "planning"
from *Compositional Connectionism in Cognitive Science* (AAAI 2004 Fall Symposium)
“legs” of the route solved by $n$-layer circuit...

finds composition of transformations that map start to end responding to terrain utility bias vector “$u$”

$$f_m = \circ_{l=1}^{m-1} \left( u \cdot \sum_j g_{l,j} \cdot t_{l,j} \right)(r)$$

for $m = 1 \ldots n$
route considering only avoiding upwind locations

route considering avoiding upwind locations and terrain “disutility”

- wolves
- elk
- upwind of elk
- gully
- hummock
Kinematic-Visual transformation (KV) MSCs
rotation in plane

edge filtered

2D visual domain

rotation in plane

3D-2D projection

3D kinematic domain

full spine generating process reflected in mappings

segment 6

segment 5

segment 4

segment 3

segment 2

segment 1

fwd

inverse

v

spine

oc

contours for each segment

occluding
How the visual (V) and kinematic (K) circuits couple
-in neurobiology this would be a “mirror system”

\[ g_i^m := \text{comp}(g_i^m, \langle a_i^m \ (f_{m-1}^K) \ , b_{m+1}^K \rangle) \] for \( m = 1 \ldots L, i = 1 \ldots n_i \)

\[ a_i^m = \langle f^L \rangle \circ t^{3D \rightarrow 2D} \circ t_{\text{surface}} \circ t_i^m \ (b_{m+1}^K) \]

“memory” for visual circuit

\[ b^{L+1} = \sum_{i,m} t^{3D \rightarrow 2D} \circ t_{\text{surface}} \circ t_i^m \ (b_{m+1}^K) \]
3D occluding contour

input view

top view

spine

re-projection of 3D occluding contour segments, (x,y exaggerated)
input view

3D occluding contour

re-projection of 3D occluding contour segments, (x,y exaggerated)
re-projection of 3D occluding contour segments,

input view

3D occluding contour
Using two “fixations”

end of first set of segments is new fixation point

re-projection of 3D occluding contour segments,

first

second

input view

3D occluding contour
Completion across interrupting occlusion
Mathematically, what is going on here?
\[ c(x) = \bigcirc_{i=1}^{L} t^{L}_{x_i}(r), w \]

discrete problem space

scale ~ \[ R \left( \prod_{i=1}^{L} n_i \right) \]
\[ Q(G) = \left\langle \sum_{l=1}^{L} \left( \sum_{i} g_i^l \cdot t_i^l \right)(r), w \right\rangle \]

scale \sim R \left( \sum_{l=1}^{L} n_l \right)
Create a continuous embedding of the discrete correspondence space by creating a measure, \( Q \), of correspondence of superpositions of the transformations of \( r \) and \( w \), where the contribution to the superposition(s) is "gated" by the coefficients \( g \)...

\[
Q : \mathbb{R}^{\sum_{l=1}^{L} n_l} \rightarrow \mathbb{R}^1
\]

\[
Q(G) = \left\langle \sum_{l=1}^{L} \left( \sum_{i} g_i^l \cdot t_i^l \right)(r); w \right\rangle
\]

(dot product, or any other \textit{goodness-of-match} quantifying function)

\[
G = \begin{bmatrix} g_{x_m}^m \end{bmatrix} \quad m = 1 \cdots L, \ x_m = 1 \cdots n_m
\]

\( where \ n_m \) is number of \( t \) in \( m \)-th composition

\( g_{x_m}^m \in [0,1] \)
Let
\[ G_x = \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ 1 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \]

such that the 1 in row \( m \) is at the index of \( x_m \) in \( x \), \( x = \arg \max c(j) \) then

\[ Q(G) = \left\langle \bigcirc_{l=1}^{L} \left( \sum_i g^l_i \cdot t^l_i \right)(r), w \right\rangle \]

\[ Q(G_x) = \left\langle \bigcirc_{l=1}^{L} t^l_{x_l}(r), (w) \right\rangle = c(x) \]

recall \( c(j) = \left\langle \bigcirc_{i=1}^{L} t^i_{j_i}(r), w \right\rangle \)
We start agnostically with... and try to end with...

\[ G_1 = \begin{bmatrix} 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \rightarrow \quad G_x = \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ 1 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & a_1 & 0 & \cdots \\ 0 & 0 & a_2 & \cdots \\ a_3 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \]

\[ Q(G) = \left\langle \bigcirc_{l=1}^{L} \left( \sum_{i} g_i^l \cdot t_i^l \right) (\mathbf{r}), \mathbf{w} \right\rangle \]

where

\[ Q(G_x) = \left\langle \bigcirc_{l=1}^{L} t_{xl}^l (\mathbf{r}), (\mathbf{w}) \right\rangle = c(\mathbf{x}) \]

we have arrived at the correct solution by \textbf{superposition culling}...
we do this iteratively by reducing those values of $G$ corresponding to the transformations which contribute less than maximally to the correspondence in superposition space $Q(G)$.

$$\frac{\partial Q(G)}{g_j^m} = \left\langle \bigcirc_{l=m+1}^{L} \left( \sum_i g_i^l \cdot t_i^l \right) \bigcirc t_j^m \bigcirc_{l=\emptyset,1}^{m-1} \left( \sum_i g_i^l \cdot t_i^l \right) \right\rangle (r), w$$

At each step the change in $G$ is computed row-wise, for the $m$-th row

$$\Delta g^m = f \left( \frac{\partial Q(G)}{\partial g_1^m}, \ldots, \frac{\partial Q(G)}{\partial g_{nm}} \right)$$

$f(\ )$ preserves the maximal component(s) and reduces the others proportional to their distance from the maximal component.

In neuronal terms: **competition** or **lateral inhibition**.

$$\Delta g_i^m = -k_1 \cdot \left( 1 - \frac{q_i^m}{\max q^m} \right)^{k_2}$$
To build MSC, using
\[
\langle (A_1 + A_2)(B_1 + B_2)Cr, w \rangle = \langle Cr, (B_1' + B_2')(A_1' + A_2')w \rangle
\]
rewrite
\[
\frac{\partial Q(G)}{g^m_j} = \langle \bigcirc_{l=m+1}^L \left( \sum_i g^l_i \cdot t'_l \right) \circ t^m_j \circ \bigcirc_{l=\emptyset,1}^{m-1} \left( \sum_i g^l_i \cdot t^l_i \right) (r), w \rangle
\]
as
\[
q^m_j = \left( t^m_j \circ \bigcirc_{l=1}^{m-1} \left( \sum_i g^l_i \cdot t^l_i \right) \right) (r) \cdot \bigcirc_{l=L}^{m+1} \left( \sum_i g^l_i \cdot t'^l_i \right) (w)
\]
\(t'_l\) is adjoint of \(t^l\)

On each iteration, update
\[
g^m_i = g^m_i + \Delta g^m_i
\]
\[
= \max\left(0, g^m_i - k_1 \cdot \left(1 - \frac{q^m_i}{\max q^m} \right)^{k_2} \right)
\]
\[
g^m = \kappa(g^m, q^m)
\]
\(\kappa\) is any competition function
(also Lyapunov, \(|g^m|\) must decrease)
For further efficiency, instead of computing

\[ q_j^m = \left\langle t_j^m \circ \left( \sum_{l=1}^{m-1} (g^l_i \cdot t^l_i) \right) (r), \circ_{l=m}^{m+1} \left( \sum_i g^l_i \cdot t^l_i \right) (w) \right\rangle \]

compute

\[ b_j^m = \sum_{j=1}^{n_m} g_j^m \cdot t_j^m (b_j^{m+1}) \quad \text{for } m = L \ldots 1, -1 \quad \text{saving all } b \text{ vectors} \]

then compute (for \( m = 1 \ldots \) as needed)

\[ v = t_j^m (f_j^{m-1}) \quad \text{(Note: transformations of } f \text{ and } b \text{ are thrown away after immediate use.)} \]

\[ f_j^m = \sum_{j=1}^{n_m} g_j^m \cdot v \]

\[ q_j^m = \left\langle v, b_j^{m+1} \right\rangle \]

Then update

\[ g_j^m = \kappa \left( g_j^m, q_j^m \right) \]

**KEEP IN MIND:**

the magnitudes of elements of \( f \) and \( b \) are “plausibility states” for a particular feature... not quantities of a property (e.g. not image intensity)
\[
\begin{align*}
\mathbf{f}^m &= \sum_{j=1}^{n_j} g_j^m \cdot t_j^m (\mathbf{f}^{m-1}) \\
\mathbf{b}^m &= \sum_{j=1}^{n_m} g_j^m \cdot t_j^m (\mathbf{b}^{m+1}) \quad \text{for } m = 1 \ldots L \\
\mathbf{b}^{L+1} &= \begin{cases} 
\sum_k g_k^{L+1} \langle \mathbf{f}^L ; \mathbf{w}_k \rangle \cdot \mathbf{w}_k \\
\mathbf{w}
\end{cases} \\
q_i^m &= \langle t_i^m (\mathbf{f}^{m-1}) ; \mathbf{b}^{m+1} \rangle \quad \text{for } m = 1 \ldots L \\
[q_i^{L+1} &= \langle \mathbf{f}^L ; \mathbf{w}_k \rangle] \\
g_i^m &= \kappa (g_i^m, q^m)
\end{align*}
\]
The **ordering property of superpositions** determines the relationship of
the components of the competition or lateral inhibition.

\[ g_i^m = \kappa(g_j^m, q_j^m) = \max \left( 0, g_i^m - k_1 \cdot \left( 1 - \frac{q_i^m}{\max q_j^m} \right)^{k_2} \right) \]

In this context, as \( \mathbf{G} \) becomes sparser (in the sense of more values < 1),
the greater the probability that

\[ q_j^m \]

is maximal for the transformation \( t_{m_j} \) such that \( x_m = j \). i.e. for the
\( m \)-th transformation in the optimal composition specified by \( \mathbf{x} \).

Seen in terms of surface \( Q \) the resulting path a "**high traverse**"
(versus gradient ascent or descent usual in optimization methods)
What is the failure mode?

Given correspondences \( c(x) > c(y) \)

\[
c(x) = \left< \bigotimes_{i=1}^{L} t_{x_i}^i (r), w \right> \quad \text{and} \quad c(y) = \left< \bigotimes_{i=1}^{L} t_{y_i}^i (r), w \right>
\]

where \( x \) and \( y \) differ in at least one index, say \( m \)

**Collusion** occurs when

\[x_m = j, \ y_m = k\]

\[q^m_j = \left< t^m_j \circ \left( \bigotimes_{l=1}^{m-1} g_{i_l}^l \cdot t_{i_l}^l \right) (r), \bigotimes_{l=L}^{m+1} (\sum g_{i_l}^l \cdot t_{i_l}^l) (w) \right>\]

\[q^m_k = \left< t^m_k \circ \left( \bigotimes_{l=1}^{m-1} g_{i_l}^l \cdot t_{i_l}^l \right) (r), \bigotimes_{l=L}^{m+1} (\sum g_{i_l}^l \cdot t_{i_l}^l) (w) \right>\]

\[q^m_j < q^m_k \quad \text{versus} \quad c(x) > c(y)\]
Observations:

Correspondences $c(x)$ only survive if they support high $q$’s in ALL layers.

$q$’s supported by numerous small correspondences are likely to diminish quickly (unless improbable distribution).

$q$’s supported by few large correspondences are likely to diminish slowly (or not at all), even if also supported temporarily by small correspondences.

Durable collusion supporting states are few, fragile collusion supporting states are many.
Definition: “Peaky” correspondence landscapes have few large correspondences and many small ones.

Conclusion: In a correspondence landscape that is “peaky”, it takes an improbable arrangement of the input pattern to sustain a destructive collusion. Collusions do occur frequently in early iterations, but rarely survive.

Analysis by correspondence set partitioning [A2002]
another viewpoint...

Highest Traverse on Q surface, not Gradient Descent or Ascent

\[ \Delta g_j^m = -k_1 \left( 1 - \frac{q_j^m}{\max(q^m)} \right) \]

where surface gradient is characterized by OPS
Collusion cases

bad corridor

good slope

anywhere along here is success