Computer Vision

James L. Crowley

M2R MoSIG

Fall Semester
21 Nov 2019

Lesson 6

Attention and Cognition for Computer Vision

Lesson Outline:

Bibliography

- N. Cowan, Working memory underpins cognitive development, learning, and education. *Educational Psychology Review*, 26(2), 197–223, 2014
- G. A. Miller, The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2), 81-97, 1956.
- J. R. Anderson, A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, *22*(3), 261-295, 1983.
- J. F. Sowa, *Knowledge representation: logical, philosophical, and computational foundations* (Vol. 13). Pacific Grove, CA: Brooks/Cole, 2000.
- M. Minsky, A Framework for Representing Knowledge, in P. H. Winston (ed.), *The Psychology of Computer Vision*. McGraw-Hill, New York (U.S.A.), 1975.
- P. N. Johnson-Laird, Mental models, *MIT Press Cambridge*, MA, USA, 1989.

1 Cognitive Vision

1.1 The Hour-Glass model in machine learning

An increasing trend in Machine Learning is to construct an "end-to-end" system that maps an input signal (such as an image) directly onto an output signal (such as spoken word or an action). This is used, for example for speech translation and for chat bots. This is done by combining a discriminative network with a generative Network.

Discriminative and Generative Networks

In lecture 2 you saw networks that learn a discriminative model using back-propagation for gradient decent. Discriminative Neural networks take a signal as an input and output the likelihood the one or more target classes are in the signal

$$\vec{X} \rightarrow \boxed{D(\vec{X})} \rightarrow \hat{y}$$

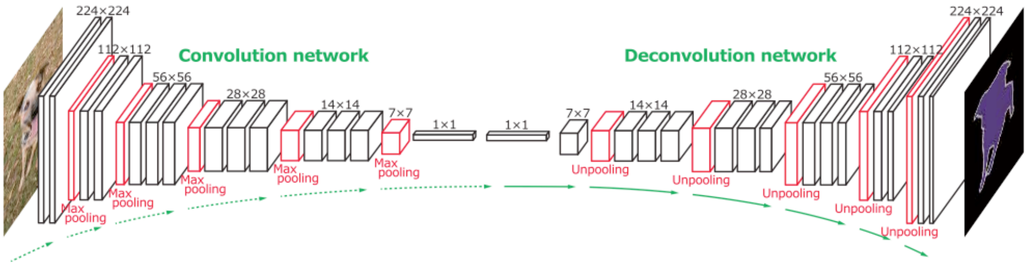
A Generative network runs the other way, generating a realistic signal from a random (or arbitrary) input.

$$y \rightarrow \boxed{G(y)} \rightarrow \vec{X}$$

The generative process can be learned using back-propagation from training data. We can put a discriminative network together with a generative network to generate an output in one domain from an input in another.

$$\vec{X} \rightarrow \boxed{D(\vec{X})} \rightarrow \hat{y} \rightarrow \boxed{G(y)} \rightarrow \vec{X}$$

This called an "Hourglass model" and is used, for example, with Chatbots.



However, this model is simple Stimlus->Response.
There is no "understanding".

What would it take to add "intelligence" to such a model?
We can find inspiration in Cognitive Science.

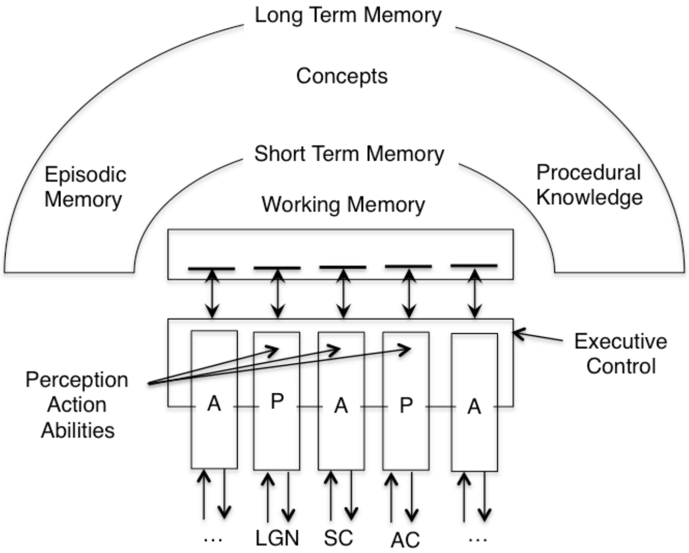
1.2 Long Term Memory and Working Memory

Most models of human cognition share a number of common elements:

- Perception: Transforms and combines sensory stimuli to Phenomena
- Action: Transforms commands in WM to motor commands.
- Perceptual Memory: Temporary buffer holding recent stimuli
- Working Memory: 7+/-2 memory entities (perceived or remembered)
- Long Term Memory: Episodic, Procedural, Spatial and Conceptual

Long-term memory (LTM) refers to memory structures used in several different cognitive abilities:

- Episodic Memories: recordings of significant sensory experiences
- Conceptual Memory: Abstract representations for sensory experiences
- Procedural Memory: Sequences of operations to accomplish goals
- Spatial memory (e.g. network of spatial relations in the hippocampus)



The core element for a cognitive system is Working Memory (WM). Working memory is a limited number of storage units that are used to associate perceptual phenomena with episodic memory, learned concepts, spatial memory, procedural knowledge and reactive skills (actions).

In humans, working memory is thought to reside in the hippocampus with dense associations to the visual cortex, auditory cortex, and many regions of the superior cortex. It is easily demonstrated that Working Memory for humans is limited to an association of 7 +/- 2 entities (Miller 56).

Perception and action are sub-symbolic processes that are activated by working memory. Visual information that passes through an attention filter triggers recognition processes that express visual patterns as concepts.

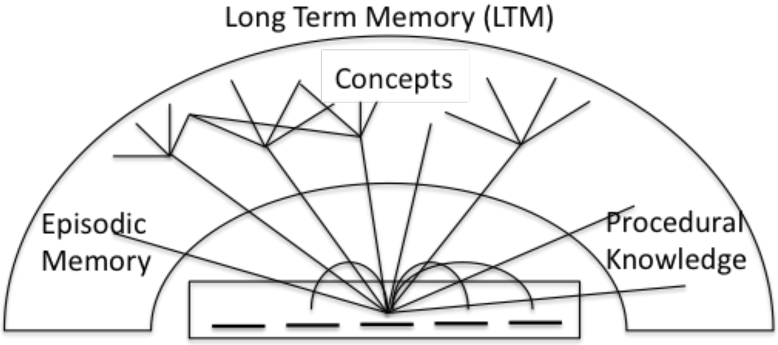
Visual concepts are expressed as associations of entities in working memory. Visual concepts represent perceptual phenomena and are associated with episodic memories, procedures, and other entities in working memory.

Actions are learned processes for locomotion, manipulation, vocal expression etc. Actions include "perceptual actions" that can enhance perception by directing fixation, tuning auditory perception, or focusing attention.

Entities in working memories are instantiations of Concepts from Long-Term memory.

1.3 Spreading Activation

Most theories posit some form of "spreading activation" (Anderson 83) in which activation energy propagates through a network of cognitive "units". preading activation is mechanism for associating cognitive units and controlling the contents of the limited Working memory.



Activation energy spreads from working memory to other elements of working memory and to long-term memory including concept memory, episodic memory, procedural knowledge, etc. Activated units then spread their energy to other units where it can arrive from multiple paths and accumulate. At the same time the energy decays with time, disappearing within 30 seconds.

Units that receive energy from several other units can become "activated" and can replace one of the active units in working memory. Theories differ in describing how activation energy propagates and how this propagation can be controlled by emotions and physiological state.

The limited size of working memory is the primary bottleneck for cognition. This limit is used to explain and predict many phenomena in human Factors (Ergonomy).

This limit is NOT because of the cost of memory. The limitation is the algorithm complexity caused by spreading activation. $O((7b)^d)$ where b is the average branching factor (number of associated units) and d is the average depth.

In Cognitive systems, spreading activation is represented as relations that are encoded as predicates in Rules or Frames.

1.4 Conceptual Knowledge

In cognitive psychology, concepts are studied as the units of human cognition. Concepts are mental constructs that represent abstract or generic ideas generalized from particular instances. Concepts are basic elements of cognition. Concepts provide abstractions for reasoning and communications.

Concepts can represent words, actions, perceived phenomena, experiences, feelings, etc. Concepts arise as abstractions or generalizations from experience in episodic memory or from the transformation of existing concepts. Concepts can also be learned by communication.

A concept is instantiated as an association of memories and other concepts. These memories may be images, sounds, image sequences, feelings, or any other perceived phenomena (e.g. taste, smell, etc).

Chunking is a process of grouping individual cognitive units into larger composed units. Chunking allows multiple cognitive units to be held in working memory at the same time, overcoming the limits to working memory. However, associations to LTM and STM are with the chunk and not its individual elements. Some theories postulate that frequently encountered situations are recorded as "chunks" giving rise to new concepts.

To say more, we need to define what we mean by a cognitive "unit". In computer science, concepts are formalized as Frames using Schema.

1.5 Schema

Schema are declarative structures for representing concepts.

Schema describes a pattern of association that organizes information. A key property of Schema is the association of concepts with procedures for perception, action and reasoning.

Schemas represent concepts as data structures with slots that define the properties of the concept and associate the concept with other concepts.

A typical Schema for a concept has
1) A name.
2) A definition (test for inclusion)
3) Meanings: memories of examples of the concept
4) Roles: Operations or procedures that are enabled or prevented by the concept.
5) Relations to other concepts and other elements in LTM.
(In many schema systems, meaning and roles are part of the list of associations).

Meaning denotes memories that serve as examples. Meanings can be from actual examples or can be imagined. Meanings can be visual, episodic, auditory, olfactory, emotional or examples of feelings.

Roles are operations or procedures (procedural knowledge) that are enabled or prevented by the concept. Roles can also refer to uses that the concept can have.

For Example: Consider the number 5.

The number 5 has
1) a name: (five in english, cinq in french etc).
2) a definition (the name of a set of all sets with 5 elements)

This is an intentional definition that may be implemented either by counting the elements (5 comes after 4) (procedural Knowledge) or by direct recognition (learned perceptual ability).

3) Meanings: Experiences with examples of the concept 5. (visual pattern, sounds).

4) Roles: Operations such as addition, subtraction, division, etc that are made possible. (Example 5 can not be directly divided by 2).

5) Relations: Associations with other concepts, episodic memories, or actions.
Multiple kinds of relations are possible:
ISA and AKO: Identifies the concept as a member of a larger class.
(AKO = A Kind Of). Examples: (5 ISA number) (5 ISA integer) (5 ISA odd)
Part-Of: Identifies the concept as a component of a larger concept
(5 is a part of the number 15, 5 is a part of the formula 15/3)
Order Relations : (5 < 6), (3 < 5), Time relations 5h is before 6h.

Relations are formalized as Predicates. They are implemented as spreading activations and using rules.

1.6 Frames

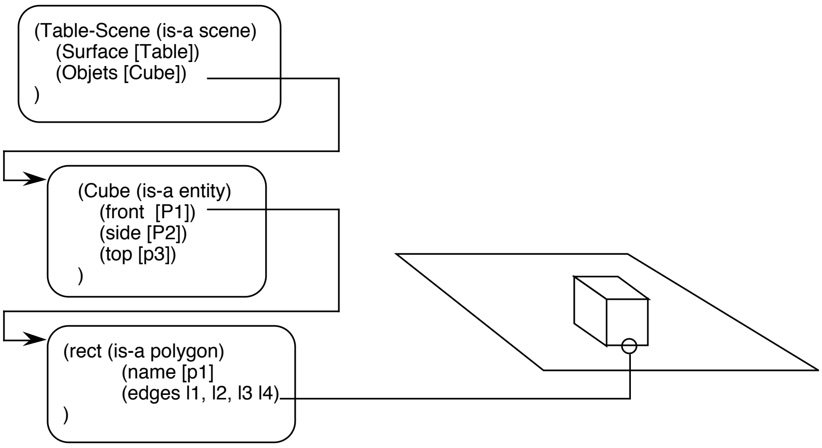
Frames are data structures that guide perception. Frames represent perceived entities as examples of concepts. Frames are used to organize perceptions in Computer Vision, Linguistics and Cognitive Systems.

Frames were made popular by Marvin Minsky (1976) as a control structure to guide visual interpretation in a top down manner. Frames tell a vision system where to look and what to look for.

A frame describes a perceived entity with a set of properties and relations, represented by slots. The frame includes a collection of procedures for perceiving, reasoning and acting with the concept.

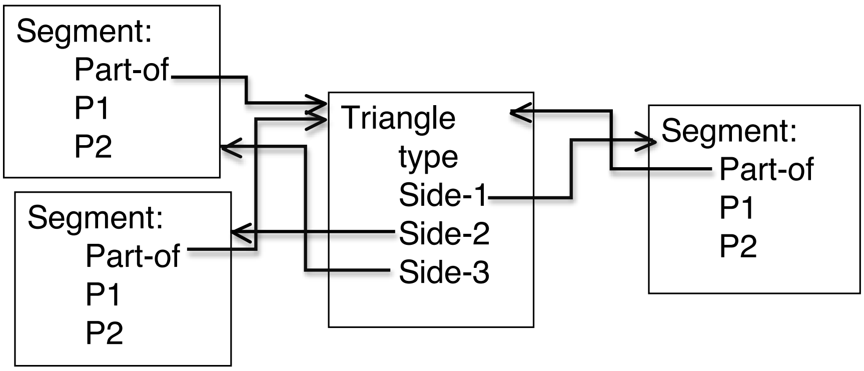
Frames provide procedures or operations to detect entities. Frames also provide default values for properties when perception is not possible or fails.

Frames can be composed hierarchically to describe complex entities.

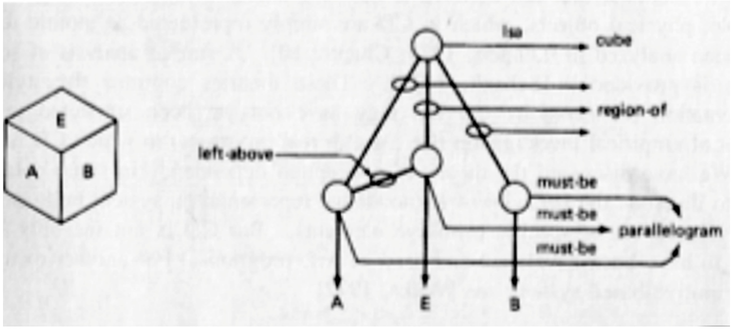


Frames are implemented as a form of Schema. They are composed of relations, represented by slots that contain pointers to other frames. Relations represents information about the object, such as part relations (composed of, part-of), Position relations (above, below, beside, inside, contains), Time relations (before, after, during), as well as specific properties of the entity (size, position, color, orientation).

For example, the concept Triangle has the part relation "composed of" with three segments. The triangle can also have an is-a relation (category membership) with different triangle types such as equilateral, isosceles, right angle, etc.



Ultimately, some slots point to raw perceptions (visual phenomena).



A Frame for a Cube
(from E. Rich "Artificial Intelligence", Fig 7-13, p231

Frames typically come with methods (procedures) for searching for the entities that can plays roles in the frame. Typically a slot-filling procedure will apply a set of acceptance tests to an entity to see if it satisfies the requirements for the slot.

Frames generally include prototypes that can serve as examples in reasoning, and default values that are used if no entity has been found to fill the slot. Thus frames can be used for abstract reasoning or for reasoning when perception is not possible.

1.7 Relations

Relations represent associations of concepts to form structures. Examples include temporal relations, spatial relations, Part-whole relations, family relations, social relations, administrative organizations, military hierarchies, and class hierarchies.

A non-exhaustive list of relations between concepts includes:
1) Class membership (ISA, AKO) relations
2) Structural (Part-of) Relations
3) Ordinal relations (bigger-than, smaller than)
4) Spatial Relations (right-of, left-of, above, below, in-front-of, behind, etc)
5) Temporal relations (Allen's 13 relations between intervals).

Relations can be defined as needed by a domain.

Relations are formalized as Predicates (Truth functions).

A predicate is function that assigns a property to an association of arguments. Predicates are often assumed to be Boolean functions. However, probabilties can be used to define probabilistic predicates to represent relations.

A predicate is a function that tells whether or not a relation is valid for a set of entities. Classically, predicates are treated as Boolean functions that can only return a value of TRUE or FALSE. As we have seen, in probabilistic reasoning, predicates represent the likelihood that the relation holds, with a value between 0 and 1.

Common relations include class hierarchies (ISA, AKO) and part hierarchies (PART-OF and COMPOSED-OF), Spatial relations (left-of, right-of, above, below), termporal relations (Before, after, during, etc).

ISA represents class hierarchy for entities. Is-a enable for reasoning about classes and categories. Has associates a concept with components. (Fish Has eyes).

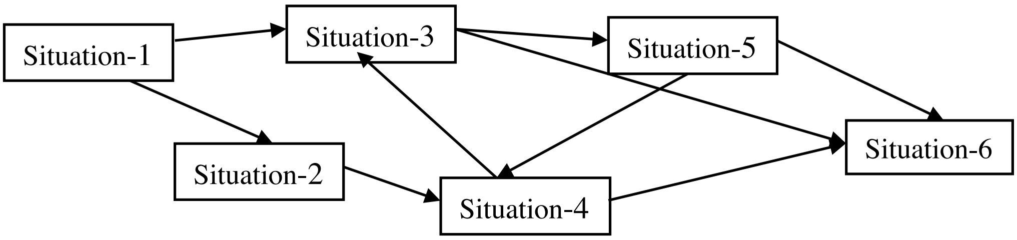
1.8 Situation Models

Situations models are used in cognitive psychology to express the mental models that people use to understand and reason. Situations models describe the contents of Working Memory.

A situate model is composed of entities with properties, and relations. Relations may be with other entities, memory episodes, concepts, actions or procedures.

Entities: Anything that can be named or designated; People, things, etc.
(entitles are defined using schema or frames)
Properties: Descriptions of entities such as position, size, color, etc
Relations: N-ary predicates (N=1,2,3 …) that relate entities.
(relations are defined as tests on the properties of entities).
Situation: A set of relations between entities

Situations can be organized into a state space referred to as a situation network.
Each situation (or state) corresponds to a specific configuration of relations between entities. A change in relation results in a change in situation (or state).



Each situation can prescribe and proscribe behaviors.
1) Behaviors: List of actions and reactions that are allowed or forbidden for each situation. Behaviors are commonly encoded as Condition-Action rules.
2) Attention: entities and relations for the system to observe, with methods to observe the entities
3) Default values: Expectations for entities, relations, and properties
4) Possible situations: Adjacent neighbors in the situation graph.

Each situation indicates:
Transition probabilities for next situations
The appropriateness or inappropriateness of behaviors

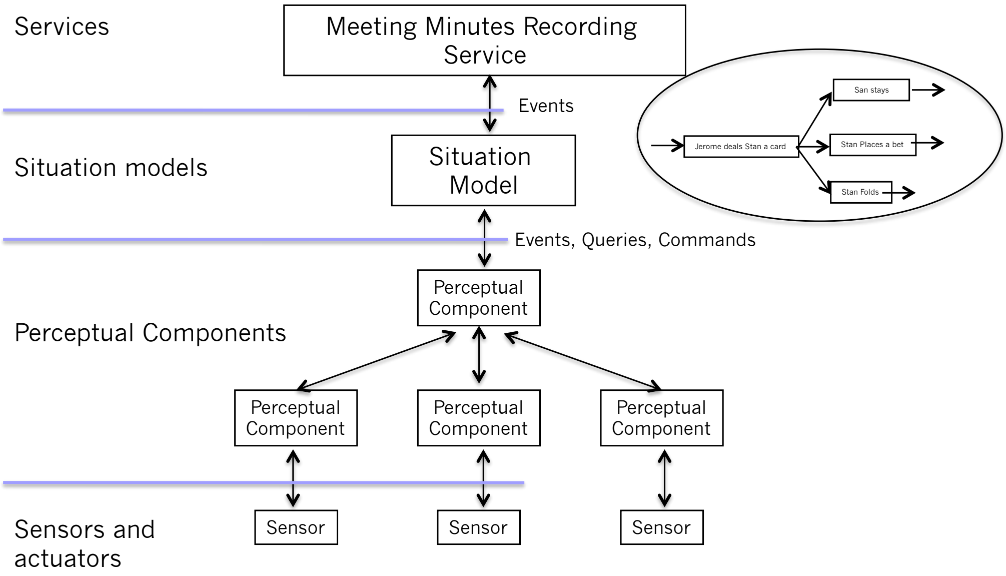
Behaviors include
1) methods for sensing and perception, and
2) appropriateness of actions
3) changes in state in reaction to events.

The sets of entities, relations, behaviors, and situations define a "Context".

Situation models are used to construct context aware systems.

A "Context" is defined as
1) A set of entities, with their properties.
2) A set of relations between entities
3) A network of situations, such that each situation specifies
- A list of adjacent situations, possibly with transition probabilities.
-A list of system behaviors that are allowed or forbidden,
possibly with preferences (appropriateness) for the situation.

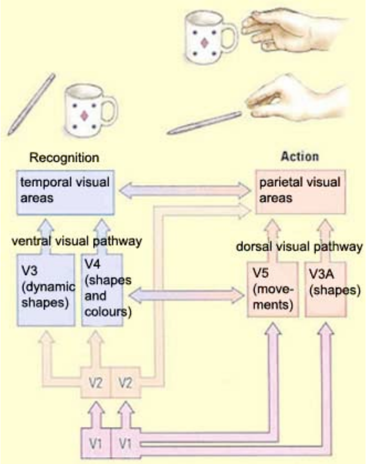
Example: Meeting Recording System



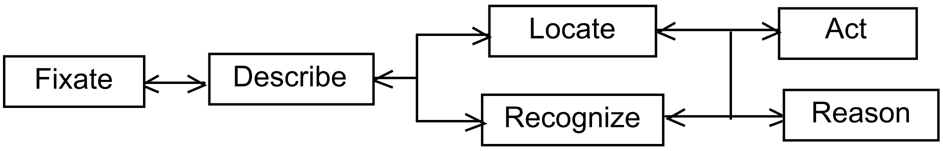
In this example, the situation model tells the system where to look, and what to look for in order to follow the meeting.

2 Vision as Process

As we saw in our first lecture, human vision is performed by a series of visual pathways.



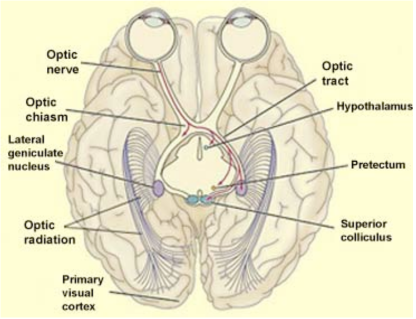
We can approximate this as a process with the following steps.

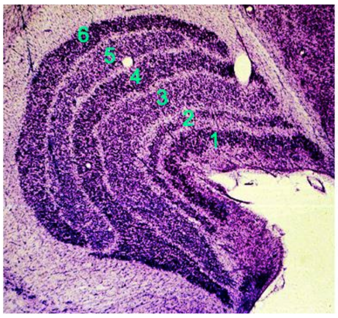


Note that each step is a tightly coupled perception-action loop.

2.1 Fixation in Human Vision

As we saw, fixation is provided by steering the horopter (region of convergence of visual fields) with the Superior Colliculus (SC).





The Superior Colliculus is organized as a set of 7 layers that receives stimulus from different regions of the brain. Each layer expresses possible targets for vergence and version as activations in a "muscle space" map of neurons. The output of the superior colliculus is an activation spike at a particular location in the vergence-version space. This spike steers the eye muscles to place the horopter at the specified angle (version) and distance (vergence).

Fixation serves to reduce computational load. Without fixation, the visual cortex would require a mass of several tens of kilograms.

Fixation is naturally reactive to noise, bright lights and looming objects. However Fixation can also be directed as a learned motor skill. For example, reading is a learned motor skill that automatically drives the horopter along a line of text. Driving a car, or piloting an airplane, require learned motor skills for fixation and attention.

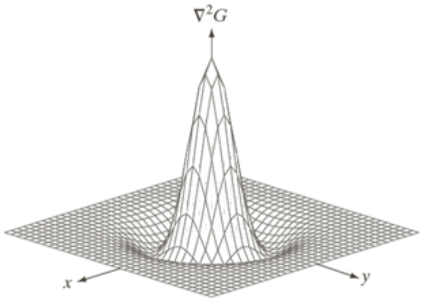
2.2 Attention in Human Vision.

Visual attention is implemented by suppression of visual information in the Lateral Geniculate Nucleus (LGN).

Visual stimulus from both retina are transmitted in the form of retinal maps to the LGN. The LGN filters these retinal maps to suppress non-attended visual stimuli. 80% of the energy in the LGN represents top-down signals from the other regions of the brain to determine what to filter and what to attend.

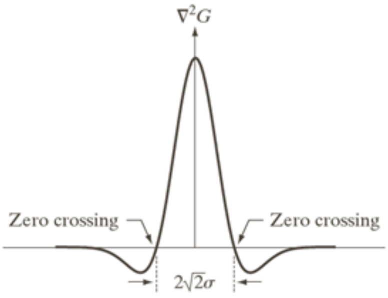
Detection and suppression are provided by Center Surround cells, sometimes referred to as a Mexican Hat and modeled as the Laplacian (2nd derivative) of a Gaussian function.

$$\nabla^2 G(x,y,\sigma) = G_{xx}(x,y,\sigma) + G_{yy}(x,y,\sigma) = \frac{\partial G(x,y,\sigma)}{\partial \sigma}$$



2D Plot of $\nabla^2 G(x,y,\sigma)$

Image of $\nabla^2 G(x,y,\sigma)$



1-D Cross section $\nabla^2 G(x,y,\sigma)$

The center surround cells are a kind of "spot detector" for blobs of a particular size. Laplacians over a range of scales are used to filter the image from the retina.

But how does the brain decide what to attend to? Models for attention are provided by Cognitive Science and widely used in Ergonomics and Human factors.

3 Fixation and Attention in Machine Vision

Fixation is used in computer vision for detection and tracking.

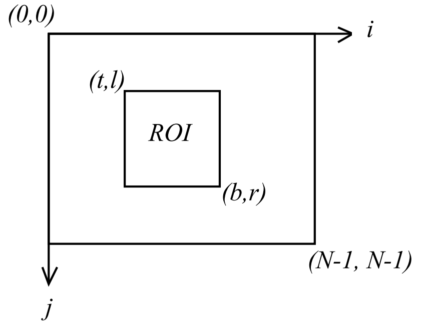
In machine vision, fixation serves to reduce computational load and reduce errors by focusing processing on parts of the image that are most likely to contain information. The fixated region is generally referred to as a “Region of Interests” (ROI). The ROI is determined either by a-prior knowledge or by some form of search procedure. In many cases this is based on tracking of targets in the scene.

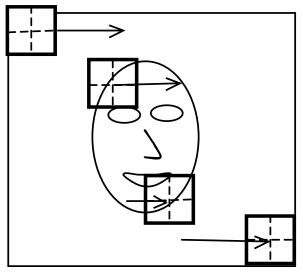
The most common form of ROI is a rectangle represented by four coordinates:
(top, left, bottom, right) or (*t, l, b, r*)

t - "top" - first row of the ROI.
l - "left" - first column of the ROI.
b - "bottom" - last row of the ROI
r - "right" -last column of the ROI.

(t,l,b,r) can be seen as a bounding box, expressed by opposite corners (l,t), (r,b)

In some vision techniques the ROI is simply a sliding window that scans the entire image. This is the technique commonly used with the Viola Jones Face Detector.





In some systems both the size and the position of the ROI are scanned. This can be on a linear scale or a logarithmic scale. Of course, with sufficient computing power, all windows can be processed in parallel.

3.1 Robust Fixation: Gaussian windows.

The problem with a rectangular window is that the window may only partially overlap the target. A more robust technique is to use a Gaussian Window.

Gaussian Fixation Window: $F(i,j;X) = e^{-\frac{1}{2}\left(\begin{pmatrix} i \\ j \end{pmatrix}-\begin{pmatrix} \mu_i \\ \mu_i \end{pmatrix}\right)^T \Sigma^{-1}\left(\begin{pmatrix} i \\ j \end{pmatrix}-\begin{pmatrix} \mu_i \\ \mu_i \end{pmatrix}\right)}$

Where : $\vec{\mu}_t = \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix}$ is the position and $\Sigma = \begin{pmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{pmatrix}$ is the scale of the mask.

This is sometimes referred to as a Gaussian “Blob”. Gaussian blobs express a region in terms of moments. The zeroth moment is sum of detection energy in the target. This is equivalent to the mass in physics and can be used to estimate a confidence factor (CF) that a target has been detected within the ROI.

The first moment gives is the center of gravity. This is the "position" of the target.

The second moment is the covariance. This gives width, height and orientation.

Target pixels *T(i,j)* are detected by multiplying the image *P(i,j)* with the fixation window.

$$T(i,j) \leftarrow P(i,j) \cdot e^{-\frac{1}{2}\left(\begin{pmatrix} i \\ j \end{pmatrix}-\begin{pmatrix} \mu_i \\ \mu_i \end{pmatrix}\right)^T \Sigma^{-1}\left(\begin{pmatrix} i \\ j \end{pmatrix}-\begin{pmatrix} \mu_i \\ \mu_i \end{pmatrix}\right)}$$

Note that that the center of the Gaussian Fixation window the coefficients are 1. The window tapers to 0.01 at 3σ. The covariance matrix is typically chosen to fit 2 or 3 standard deviations within the ROI.

3.2 Moment Calculations for Blobs

Given a target detection image *T(i,j)* within some region of the image *(r,l,t,b)*. This can be the entire image or a smaller window.

Zeroth Moment or Sum: $S = \sum_{i=l}^{r} \sum_{j=t}^{b} T(i,j)$

We can estimate the "confidence" as the average detection probability:

Confidence: $CF = \frac{S}{(b-t)(r-l)}$

First moment or Center of Gravity:

The first moment is the center of gravity of the target

$$\mu_i = \frac{1}{S}\sum_{i=l}^{r}\sum_{j=t}^{b} T(i,j)\cdot i$$
$$\mu_j = \frac{1}{S}\sum_{i=l}^{r}\sum_{j=t}^{b} T(i,j)\cdot j$$

This gives a position vector *(μ_i, μ_j)*
We will use this as the position of the blob.

Scale and orientation or Second Moments:

The second moments tell the spatial extent of the blob.

$$\sigma_i^2 = \frac{1}{S}\sum_{i=l}^{r}\sum_{j=t}^{b} T(i,j)\cdot (i-\mu_i)^2$$
$$\sigma_j^2 = \frac{1}{S}\sum_{i=l}^{r}\sum_{j=t}^{b} T(i,j)\cdot (j-\mu_j)^2$$
$$\sigma_{ij}^2 = \frac{1}{S}\sum_{i=l}^{r}\sum_{j=t}^{b} T(i,j)\cdot (i-\mu_i)\cdot(j-\mu_j)$$

These compose a covariance matrix: $\Sigma = \begin{pmatrix} \sigma_i^2 & \sigma_{ij}^2 \\ \sigma_{ij}^2 & \sigma_j^2 \end{pmatrix}$

This suggests a "feature vector" for the blob: $\vec{X}_n = \begin{pmatrix} \mu_x \\ \mu_y \\ \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix}$

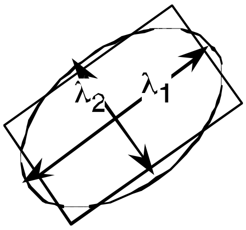
Alternatively we can use the position, length, width and orientation: $\vec{X} = \begin{pmatrix} x \\ y \\ l \\ w \\ \theta \end{pmatrix}$

where length, width and orientation are determined from principle components (λ_1, λ_2) of Σ.

We can discover these by principle components analysis.

Principle Components Analysis

The principle components of the blob are found by rotating the covariance to for a diagonal matrix.



$$R\Sigma R^T = \Lambda = \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix}$$

where

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

The principle components, λ_1^2, λ_2^2, are the eigenvalues or characteristic values of Σ.

The length to width ratio, λ_1/λ_2, is an invariant for shape.

The angle θ is a "Covariant" for orientation.

We can use λ_1 and λ_2, to define the "width and height" of the blob:

Length: $l=\lambda_1$,
Width: $w=\lambda_2$

where $x=\mu_i$, $y=\mu_j$, $l=\lambda_1$, $w=\lambda_2$ and $\theta = \tan^{-1}\left(\frac{r_{21}}{r_{11}}\right) = \tan^{-1}\left(\frac{\sin(\theta)}{\cos(\theta)}\right)$

The sum of the detections tells a confidence of a detection of a target:

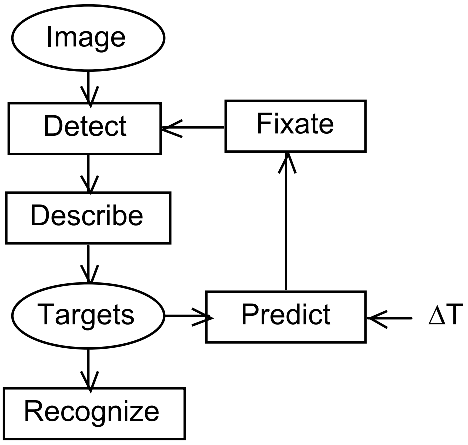
$$CF = \frac{S}{(b-t)(r-l)}$$

The confidence (CF) can be seen as the "Likelihood" that a target has been detected.

Tracking allows us to continually update an estimate for the features of the target, even if the target is temporarily lost to occlusion or noise.

4 Tracking and Object Constancy

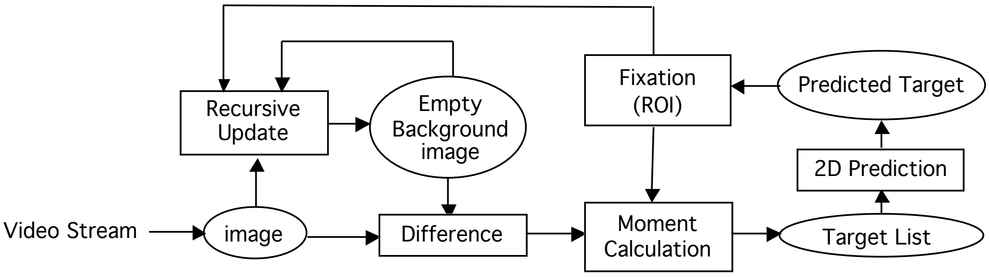
Tracking is a process of recursive estimation of target parameters from observations. Tracking is widely used in machine vision to provide reduce computational load, reduce errors and provide object constancy (preserve identity of entities over time).



We can use many different methods to detect target entities.

4.1 Detection by Background Difference Subtraction

A common example is detection using Background Difference Subtraction.
This technique assumes that the camera is stationary and is a widely used for Video Surveillance.



Tuned Parameter: α update rate. (Typically ~ 0.01)

This algorithm uses a Gaussian Blob to represent targets.

Target parameters for N targets are $\vec{X}_n = \begin{pmatrix} \mu_x \\ \mu_y \\ \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix}$.

The technique detects targets with a Gaussian Fixation window:

$$F(i,j;X) = e^{-\frac{1}{2}\left(\begin{pmatrix} i \\ j \end{pmatrix} - \begin{pmatrix} \mu_i \\ \mu_i \end{pmatrix}\right)^T \Sigma^{-1}\left(\begin{pmatrix} i \\ j \end{pmatrix} - \begin{pmatrix} \mu_i \\ \mu_i \end{pmatrix}\right)}$$

Algorithm:

0) Initialize: Acquire an empty background image

$$B(i,j) \leftarrow P(i,j)$$

Loop:

1) Image Difference: $D(i,j) = P(i,j) - B(i,j)$

Note Difference can be positive or negative!

2) Fixation: For each target N:

Fixated target image: $T_n(i,j) = |D(i,j)| \cdot F(i,j;X_n)$

3) Update moments for each Target, n

$$S = \sum_{i=l}^{r}\sum_{j=t}^{b} T_n(i,j)$$

$$\mu_i = \frac{1}{S}\sum_{i=l}^{r}\ \sum_{j=t}^{b} T_n(i,j)\cdot i$$

$$\mu_j = \frac{1}{S}\sum_{i=l}^{r}\ \sum_{j=t}^{b} T_n(i,j)\cdot j$$

$$\sigma_i^2 = \frac{1}{S}\sum_{i=l}^{r}\ \sum_{j=t}^{b} T_n(i,j)\cdot (i-\mu_i)^2$$

$$\sigma_j^2 = \frac{1}{S}\sum_{i=l}^{r}\ \sum_{j=t}^{b} T_n(i,j)\cdot (j-\mu_j)^2$$

$$\sigma_{ij}^2 = \frac{1}{S}\sum_{i=l}^{r}\ \sum_{j=t}^{b} T_n(i,j)\cdot (i-\mu_i)\cdot (j-\mu_j)$$

$$T_n = (\mu_i, \mu_j, \sigma_i, \sigma_j, \sigma_{ij})^T$$

$$CF = \frac{S}{(t-b)(l-r)}$$

$$CF_n \leftarrow \eta CF + (1-\eta)CF_n$$

4) Suppress target from difference image:

$$D(i,j) \leftarrow D(i,j)\big(1 - F(i,j;X_n)\big)$$

5) Detect any new targets (algorithms vary).

6) Update background image with remaining difference image

$$B(i,j) \leftarrow \alpha \cdot D(i,j) + (1-\alpha)\cdot B(i,j)$$

(typical value is $\alpha = 0.01$)

Demonstration Video

(This is a video from a tracker demonstrated at the first PETS 2000 workshop (Performance Evaluation for Tracking and Surveillance). The code was later comercialised by a start up BlueEye Video.)

4.2 Detecting New targets

Methods to detect new targets generally depend on the application domain.

Popular methods include

1) Entry regions. Place an ROI window of approximate target size on areas where new targets can appear. (edges of the image, doors, roadways, etc).
Detect new targets from residue image, by computing the moments in each entry ROI window.

2) Stochastic detection: Place a ROI window of approximate target size at random places in the image at the end of each cycle. Detect new targets by computing the moments in each entry ROI window.

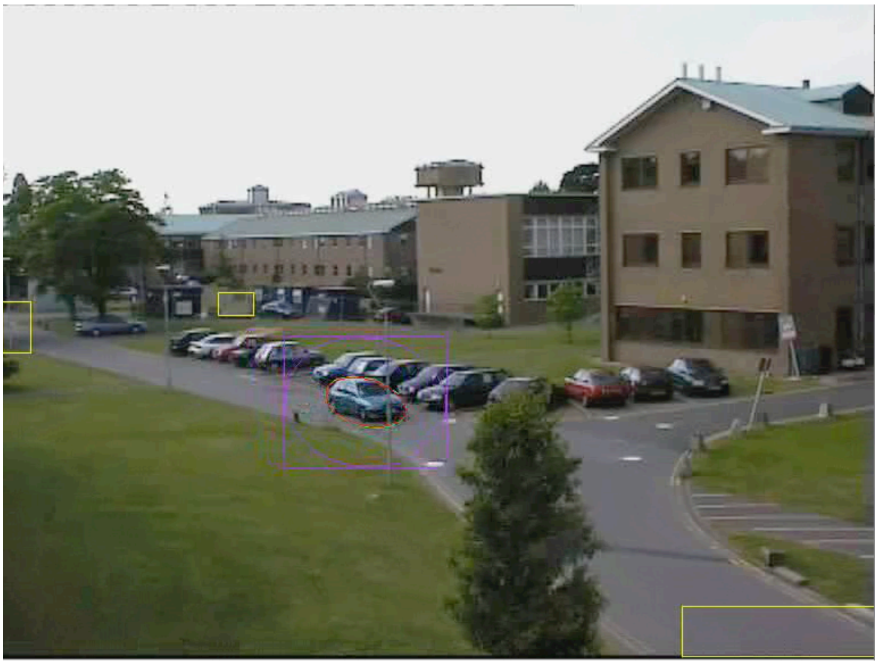


Image from PETS 2000 Data Set. Yellow boxes are Detection Regions.