

# Associative data structures

Gerrit Lebbink

Aris b.v.

Eindhoven

12-5-2011

# Associative data-structures.

---

## Introduction.

This paper is a short description of the structure and the use of Mams.

Mam is an abbreviation of “Machine Associative Memory”. The term is adopted from “Human Associative Memory”, which is the title of a book written by the psychologists Bower and Anderson. The development of Mams is inspired by the work of those psychologists who create and test models in order to investigate the functioning of human brains.

Mams are designed to represent something that is being observed. They provide a “memory” for the interpretation of incoming data-streams. Throughout this paper representations are datasets in a computer-memory and operations are performed by a processor.

In the example used here, input-data come from a video-camera, producing a sequence of images. Each image is processed, resulting in a “Plant-skeleton”. Plant-skeletons of a sequence of images are stored in one Mam.

## Plant-skeletons.

Plant-skeletons are derived from a Distance Transform. In a Distance Transform the value of each foreground-pixel represents the shortest distance to the border between foreground and background. All pixels of the background have value 0.

A Plant-skeleton is a representation of an image, consisting of Lines, Splits and Fences.



A Distance Transform and its Plant-skeleton: green lines are Lines, white points are Splits and gray contours are Fences.

Each point of a Line has 3 parameters: (X, Y, Width). X and Y are coordinates in the Image. Width is the value of the Distance Transform at (X, Y). Width represents the local width, perpendicular to the tangent of the Line. A Split has one such point (X, Y, Width). Width of a Split represents the diameter of a circle. The contour of a Fence consists of points with X and Y only.

A Line has two ends. Each end is connected with a Split or with a Fence, or it is a “free” end. In an image connections may exist between Lines and Splits and between Lines and Fences. Two Splits can also be connected.

A nice feature of Plant-skeletons is that they facilitate distinction between stalk shaped parts and leaf or flower shaped parts. It is not true that a Line always represents a stalk or twig, or that a Split always indicates a branching point. The advantage of Plant-skeletons is that Lines, Splits and Fences enable tracking of parts of images in a sequence of images, even before it is clearly known what these parts represent. This opens the opportunity to make decisions based on sequences of image parts. Examples are sequences of Splits or Lines, or even sequences of partial structures, e.g. a Split and the Lines connected with it. Decisions based on sequences are generally more reliable than decisions based on single image parts.

The need to postpone decisions has led to the development of Mams.

## Mams.

Mams can represent anything consisting of discernable parts (Nodes), which may or may not be linked with one another.

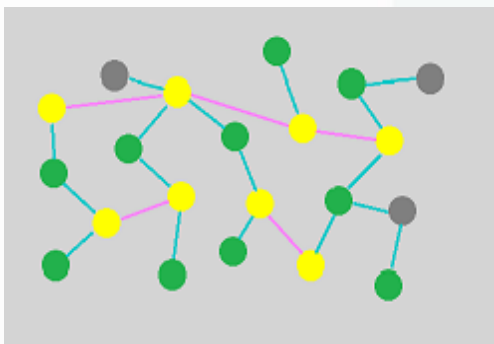
Inheritance as defined in Object Oriented Programming has the ability to let objects inherit features from other, simpler objects. Inheritance is used in the definition of “tMam” and “tNode” in order to create a structure for data-management without any knowledge of the data.

In its basic form a Mam is scarcely more than an array of pointers to the Nodes belonging to the Mam and a few processes to support creation of Nodes and to remove Nodes from the Mam. The array of pointers to Nodes is called “NodeArray”.

The way Mams function is embodied in the definition of the object “tNode”. Each Node has a type and a list (or dynamic array) of Links. A Link consists of a pointer to another Node, a type and a direction. Links between two Nodes in a Mam are “double-links”: if Node A has a Link to Node B, then Node B has a Link to Node A. When a Link between Node A and Node B is directed, then the two parts of the double-link have opposite directions.

Remark: there is no objection to define tMam as a descendant of tNode. Problems may arise if a Mam is a Node within itself, but this can be avoided very easy.

Nodes as well as Links may be of different type. The data-content of a Node depends on the type of the Node and of course on the application at hand. The type of a Link reflects the reason why the two Nodes are Linked.



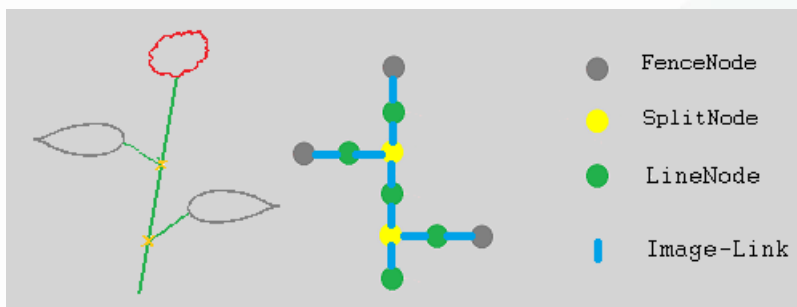
A (mini) Mam: balls are Nodes and lines are Links. Colors reflect the type of Nodes c.q. Links. This Mam has 3 types of Nodes and 2 types of Links.

The structure of a Mam is similar to the structure of a Graph, the difference is that Nodes as well as Links in one Mam usually have different meaning.

Linking pairs of Nodes results in other (indirect) connections between Nodes. This enables processes like detection of disjunct parts and traversing a Graph or Mam, guided by information in the Nodes and the Links.

In Mams a group of connected Nodes may be replaced by one Node (of another type). Nodes may be accessed in two ways: through NodeArray or through Links with other Nodes.

A plant-skeleton is stored in a Mam by creating a LineNode for each Line, a SplitNode for each Split and a FenceNode for each Fence. These Nodes are descendants of tNode. They take the data from the corresponding part of the Plant-skeleton and share the capability to have Links to other Nodes. Pairs of Nodes belonging to connected parts in the Plant-skeleton are linked with a double-link of type "Image-Link".



A simple Plant-skeleton and its representation in a Mam.

A "RingMam" can contain a sequence of Plant-skeletons from successive images. tRingMam is a descendant of tMam. The NodeArray of a RingMam is a ringbuffer. The number of Nodes a RingMam can hold is determined by a parameter at initialisation.

Processing input is iterative. For each incoming image do:

[Image]->{Preprocessing}->[Plant-skeleton]->{Update}->+[RingMam]

Instead of creating a new RingMam, the Update process adds information to an existing RingMam. Update consists of 3 parts:

1. AddSkeleton adds a Plant-skeleton to the RingMam. This new information leads to changes in the RingMam. (see 2. and 3.)
2. UpdateTraces extends "Traces" and creates new ones. A Trace is a sequence of Nodes, all of the same type, coming from a sequence of images. (see Tracking in Mams.)
3. Application dependant Update processes.

AddSkeleton creates Nodes in the RingMam and Links between pairs of Nodes, starting with all FenceNodes. Next are the SplitNodes. If a Split in the Plant-skeleton is connected with a Fence or with a Split for which a SplitNode already exists, an Image-Link is created between the belonging Nodes in the Mam. AddSkeleton ends with creating LineNodes for all Lines of the Plant-skeleton. Both ends of each Line are tested on connection with a Split or a Fence in order to create the appropriate Image-Links between the corresponding Nodes.

The ringbuffer of a RingMam is paged. All Nodes resulting from one Plant-skeleton reside within a Page, consisting of pointers in consecutive positions (indices) in NodeArray. The ringbuffer holds a number of pages. If there is no room for a new Node the eldest Page is removed from the RingMam.

This enables continuous watching.

## Tracking in Mams.

Tracking connects representations of successive images by linking Nodes of successive Pages. Tracking results in Traces. A Trace is a sequence of Nodes, representing different views of the same thing, e.g. a sequence of views of the same branching point in a moving or rotating plant.

Remark: the vague expression “same thing” has some relationship with a general problem encountered when humans make descriptions of reality. Within a formal description everything can be defined logical, but the step from reality to description involves choices of which the correctness can only be proven by experiments.

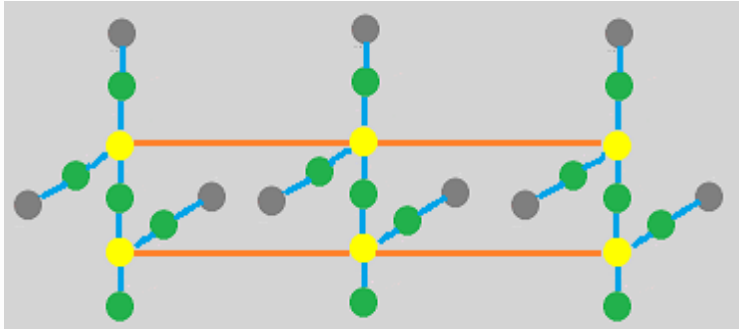
Tracking uses two tests:

1. A “first order” test on each pair of Nodes, both of the same (selected) type, one from the last Page and the other from the previous Page. The test determines whether the Nodes will be linked or not. This kind of Link has type “TimeLink”. A TimeLink means that the linked Nodes may represent different views of the same thing in successive Images. TimeLinks are directed.
2. A “second order” test on Nodes of the previous Page, having one or more “back” TimeLinks and one or more “forward” TimeLinks. Each pair of a back Link and a forward Link is tested on how good this Node fits between the Node at the other side of the back Link and the one at the other side of the forward Link.

The first test serves as a filter for the second. The second test often results in a number of alternative ways to “bridge” the middle Node, each with a value indicating how good this alternative fits. Many different features, like color, form, movement, etc. are useful for tracking. For plants moving in a way, known at beforehand, a model of this movement can be used.

If junctions in Traces are excluded, the alternative that fits best is chosen to extend a Trace or to create a new one. If junctions exist Traces can be linked in order to represent the structure of splitting and merging Traces. Each Trace in a RingMam is a Node in a (sub)Mam of Traces. So a Trace may be linked with other Traces.





Detail of a RingMam with two Traces of SplitNodes (yellow balls). The orange lines indicate TimeLinks.

Everytime a Trace is extended, more information about the observed object or detail is gathered. Easy acces to all Nodes in a Trace facilitates proper interpretation of this information.

Traces provide information about changes, e.g. movement. On the other hand knowledge about the way observed objects move (or change) can support Tracking.

Tracking adds information to a RingMam which is not directly present in the sequence of images, namely information about the continuity in the sequence of images.

Remark: If we watch a running animal, we do not see a number of instances of that animal. We see the animal continuously even if we watch a running animal in a movie. This feature of human perception is obvious to us. The fact that observations, done at different times, are related, indicates that “memory” must be involved.

Watching an object continuously enables cumulation of information about the observed object, which raises the accuracy of judgements.

## Conclusion.

Aris has used Graphs for image-processing since 1995. It was observed that Graphs offer great flexibility when used for representations of images of plants.

Most of Aris-applications use more than one view of each object. As processors became faster, more views could be processed. Around 2000 all images of a videostream could be processed in many applications. Mams and Tracking extend the flexibility of Graphs to the interpretation of video as it is: a film.

There is an interesting difference between human perception and the way a vision-system with a fixed camera operates. If we focus on a moving object we can keep the object in the center of our view. The result is that we do not see much detail of other objects in the neighbourhood. A vision-system equipped with Mams and Tracking can follow all objects in a stream of objects at the same time in order to count the objects and/or inspect each object.

Since 2008 Mams are implemented in a number of Aris-applications. This has led to the development of many application dependant update processes. My personal favorite is a self-propagating proces that uses redundancy in succeeding images to change the probability of a Split, being a real branching point in a plant.

Description of application dependant update processes is beyond the scope of this paper. The purpose of this paper is to put on stage a challenging and effective approach to data-processing. My personal objective is to share knowledge that has offered me joy and succes.

Gerrit Lebbink,

Aris b.v.

Eindhoven.