# TriangleVision: a toy visual system

Thomas Bangert

15 Bence House, Rainsborough Avenue, SE8 5RU, London, UK
txbangert@hotmail.com
+44 (20) 8691 6039

**Abstract.** This paper presents a simple but fully functioning and complete artificial visual system. The triangle is the simplest object of perception and therefore the simplest visual system is one which *sees* only triangles. The system presented is *complete* in the sense that it will *see* any triangle presented to it as visual input in bitmap form, even triangles with *illusory contours* that can only be detected by inference.

**Key words:** computer vision, image analysis, Kanizsa figures

## 1 Introduction

The human visual system is extraordinarily complex[1]. It is so complex that little progress has been made in understanding its function. Furthermore, very little progress has been made in artificially (computationally) replicating functionality of visual systems found in humans and other organisms, even organisms with much simpler visual systems.

The most advanced artificial visual systems have recently been put to the test in the DARPA Grand Challenge[2]. The Grand Challenge is a competition sponsored by the United States Defense Advanced Research Projects Agency in which the entrants are autonomous vehicles that must navigate a route across difficult desert terrain. This is a task that is primarily a visual one, and one which many animals are able to perform with ease, using visual systems that are varied in design but generally much less sophisticated than the human visual system. In 2004 when the Grand Challenge was first held none of the vehicles completed the race and when it was held in the following year only 5 out of the 23 finalists were able to complete the race. This is despite the fact that all the vehicles were required to do was to follow an unpaved desert road from the start position to the destination; something a human driver is able to do with ease. The 5 vehicles that completed the race did not complete the race because of their ability to *see* the road, but because entrants were given a detailed map of the route in advance and allowed the use of the Global Positioning System to determine in real time their precise location on the map. The vehicles therefore did not need sophisticated visual systems to navigate the route. The only visual information the vehicles needed was detection of obstacles (such as rocks or potholes). To detect obstacles the entrants employed an array of technically sophisticated and expensive sensors such as laser rangefinders and radar. These sensors performed dedicated tasks,

primarily *obstacle detection*. None of the entrants employed what could be called a more general visual system, which would require sophisticated processing but allow the use of simple and inexpensive sensors.

Visual systems vary in complexity. Some organisms such as humans have a very sophisticated visual system whereas other organisms have much simpler visual systems[1]. Even the simplest of the visual systems are not well understood. One way to classify the complexity of a visual system is by the complexity of the geometric objects the visual system is capable of representing. Circles may be said to be more complex than polygons and therefore a visual system that is capable of processing line arcs as well as straight lines may be said to be more complex. Rather than attempting to computationally reproduce the function of the most complex natural visual systems, the approach taken here is to construct a visual system of minimal complexity.

It is assumed that the basis of any visual system is geometry, and the simplest of geometric objects is the triangle. Therefore the simplest of all possible visual systems is one which *sees* only triangles. This paper presents such a visual system. The visual system presented is limited to representing triangles but it is capable of detecting all triangles from a given visual input, even meta-triangles (triangles whose edges are themselves composed of triangles) and *illusory* triangles, which can only be detected by inference.

### 1.1 Completeness

We may say that a visual system is complete when it is able to detect from a given visual input all that can effectively be represented by that visual system. That is, that there is no other visual system with an identical imaging model that is able to extract additional objects from the same visual input.

### 1.2 PostScript

PostScript is a system whose purpose is to represent text and graphical images for reproduction on printed pages. To do this it employs a set of geometric primitives and rules in respect to how they are used. This set of rules is called the *imaging model* of the system.[3] The creators of the PostScript imaging model claim it to be "very similar to the model we instinctively adopt when drawing by hand"[4]. It is suggested that when we *instinctively* draw by hand that the visual system of our brain is employed to carry out the processing necessary for the task. Therefore when seeking to develop an artificial visual system it may well be useful to take inspiration from existing models that in some way reflect the visual system of the human brain.

When we draw by hand we must draw on a piece of paper or canvas. It is always the case that we draw *on* something, and this something may be seen as the framework (or space) of that activity. The canvas does not form a part of the drawing itself but establishes the space on which the drawing is set out. In this same sense, PostScript posits an abstract drawing space, which is referred to as the *current page*. It is a two dimensional space (or canvas) upon which

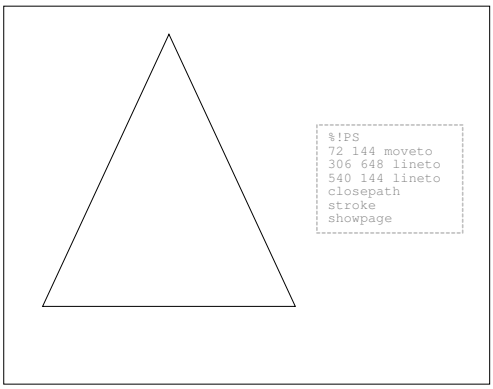


**Fig. 1.** A triangle, and its PostScript code. The line width is set automatically by PostScript to the default. It may also be set to any arbitrary width with the `setlinewidth` command.

PostScript *painting operators* draw. This space is a Cartesian plane where any arbitrary point can be uniquely addressed by the Cartesian coordinate system. Postscript output is confined to the first quadrant of this plane and therefore the bottom left corner of the page is defined as the origin for a PostScript page.

The basic PostScript geometric operators are coordinate points and lines. A line is drawn by setting two points (by giving the X-Y coordinate) and giving the line draw command. More complex geometric objects may be drawn by repeating this process. Figure 1 gives an example of how a triangle is drawn using PostScript.

PostScript represents objects such as triangles in a rigorous formal way. Lines are drawn between set points and then the area enclosed by the path set out by the lines is used to establish a shape. PostScript can represent very complex shapes, but we will restrict ourselves to the simplest of its geometric shapes, the triangle. Under the PostScript imaging model the exterior of a shape can have its lines stroked (that is drawn) and the shape can be filled with a texture. Figure 1 gives an example of a conventional triangle with stroked lines. Figure 2 gives an example of a triangle with a simple gray texture and no stroked lines. Lines can be can also be more complex by being dashed. Shapes can also be defined that have no stroked lines and no texture as shown in figures 6 and 7.

## 2 Prolog

The principal function of an artificial visual system is to induce detailed and precise geometry from a visual input that is incomplete, inconsistent and poor in quality. Prolog (PROgramming in LOGic) is the only programming language that is non-deterministic and allows the representation of entities which do not have to be instantiated at the outset[5]. It allows for the establishment of relations between as yet unestablished entities and without having to provide further

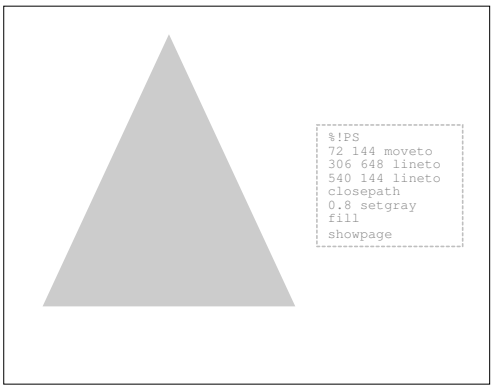


**Fig. 2.** A PostScript triangle with grey texture and no line.

detail will work through all the possibilities and determine the values that allow an entity to be instantiated once sufficient grounds for it are found[6]. These are the kinds of processes inherent to visual systems and therefore Prolog is a natural choice for implementing the processing framework of an artificial visual system.

Most Prolog implementations allow what might be called a logical space or framework into which new statements of fact may be asserted or from which previously asserted statements may be retracted. It is therefore possible in Prolog to directly implement a logical visual space set out purely by the geometrical entities of that visual space and the logical relationships between them. Once seeded with the facts that represent raw sensor data such a system would operate on the raw facts by way of recursive transformation. This system would transform the initial perception-objects into entities of increasing abstractness until the visual space reaches a steady state that is populated entirely by the geometrical entities of the visual space. Recursion is inherent to Prolog and it is a process that operates from a set of rules by which entities are replaced by new entities and where these new entities in turn become subject to the same process and may in turn are replaced.

## 3 Retinal Processing

The retina is the only part of the human visual system that is to some degree understood[1]. While it may well be possible to design visual systems that are superior to the visual systems found in natural organisms the fact that visual systems have so far proven to be in practice intractable suggests that artificial visual systems should take inspiration from natural visual systems.

The first layer of the retina (see figure 8) is a simple sensor array which responds to light. Assuming rate coding, each sensor produces a value representing light intensity. This function is well understood and has been replicated with considerable success for a variety of functions, notably in digital cameras. A

digital camera simply stores the sensor light intensity values at a certain point in time in a two dimensional array, which can then be read back and viewed from the camera's memory store.

While photography as a whole may be said to have been inspired by the study of the human visual system, specifically the retina, the function replicated by the camera is merely the first layer of the retina, this being a simple sensor array. However, the retina (unlike the digital camera) does not send the output of the sensor array directly to the brain. The retina incorporates an additional layer that carries out pre-processing function(s) and the neural circuitry for this function is arranged immediately behind the sensor array (actually in the mammalian retina the physiology is inverted). This pre-processing circuitry is not complex, has been studied in considerable detail and its function is to some degree understood.

In the mammalian retina what the pre-processing circuitry does is to arrange the individual light sensors into groups. Each group operates as receptor unit, has a set function and produces a unitary output. These groups are generally referred to as receptive fields. Receptive fields do not produce a value for light intensity but rather a value for a change in the intensity between two areas of the field. Receptive fields are roughly circular in organization and have two distinct areas; a circular centre and a a concentric surround (see figure 3). A receptive field measures the difference in light intensity between these two areas and gives as output the value of the contrast. There are two distinct types of receptive fields: on-centre and off-centre. The former gives a contrast value when the light intensity is greater in the centre than the surround (the center is *on* and the surround is *off*)and the latter gives a contrast value when the light intensity in the surround is greater than in the centre (the center is *off* and the surround is *on*). It is these contrast values that are sent from the retina to the brain. It is important to note that these contrast values are the only visual information that the brain receives.

While the function of receptive fields within the visual system is not fully understood, it is generally accepted that receptive fields are a first step in the process of edge detection. How edges are detected or elicited from images has been a matter of study for approximately half a century and has elicited many rather complex models[7]. This complexity has in part been driven by the empirical observation that receptive fields have a concentric surround. This concentricity is paradoxical to the otherwise straight forward function of edge contrast detection.

For our simplified visual system we have adopted a simple computational model of receptive fields, a model that is directly useful for edge contrast detection. It is proposed that the empirical findings in respect of receptive fields are correct but that there are alternatives to the concentric surround hypothesis that are consistent with the empirical findings. It is proposed that the concentric surround is not a whole unit but is divided into segments. It is assumed that the surround is composed of 8 units, each being equal in size to the the center. Each surround unit calculates a contrast value in comparison to the centre.

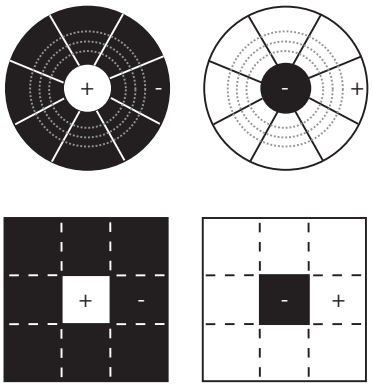


**Fig. 3.** Receptive Fields. The receptive fields of natural visual systems are circular and vary in size. This has been simplified to receptive fields with square zones of a fixed size of 3 by 3 pixels.

This is then integrated to a single output. A receptive field therefore calculates 8 separate contrast values internally and takes the maximum of these values as its output. A receptive field of this type when presented with the conventional experimental stimuli would give an output identical to the conventional model of a receptive field.

## 4 the Visual System

The system is designed to take any PostScript produced raster (bitmap) image and produce from it geometric representation that may be used to accurately recreate the original PostScript code. The original PostScript code may be arbitrarily complex, but limited to the only geometric form allowed by our system, the triangle.

### 4.1 Receptive Fields and Points

Receptive fields of the human visual system have a number of different sizes. For the sake of simplicity the visual system presented here employs receptive fields of a single fixed size. It is impractical to perform edge detection pixel by pixel and therefore the receptive field size was set to 3 pixels. The total size of a receptive field is therefore 9 pixels (square). To achieve the best resolution the only the center of the receptive fields are mapped to the bitmap input rather than both the center and surround. This requires surrounds to overlap onto the centers of other receptive fields. This effectively sets the receptive fields to 3 by 3 pixels. If the sensor array is (for example) 1350 by 900 then the number of receptive fields would be 450 by 300. The center of each receptive field receives a fixed input of 9 pixels and the surround is a set of 8 zones also of 9 pixels each. The receptive fields are in a fixed location but to allow for some overlap and flexibility

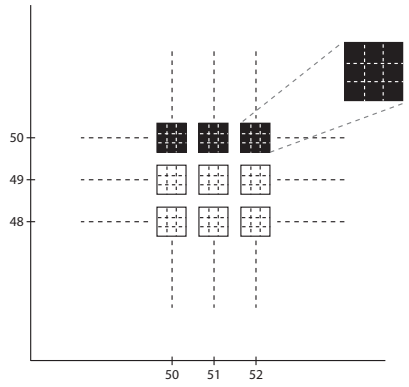


**Fig. 4.** A bitmap image divided into receptive field zones of 3 by 3 pixels. The pixels shown illustrate a horizontal black/white contrast zone.

each receptive field may shift one pixel in any direction to maximize available contrast. Receptive fields come in two types: center-on and one center-off. This means that for every group of 3 by 3 pixels there are two receptive fields. All receptive fields are autonomous of each other.

The function of a receptive fields under this system is to assert a contrast value fact when its contrast threshold (a global value) is breached. Facts are asserted as Prolog statements. For example, in figure 4 the center-off receptive fields at addresses (50,50), (51,50), (52,50) will activate and find a contrast of 255. As a result they will assert the following Prolog statements:

```
point((50,50),255).
point((51,50),255).
point((52,50),255).
```

The center-on receptive fields at (50,49),(51,49), (52,49) will also activate and assert the corresponding points, but with inverse contrast.

```
point((50,49),-255).
point((51,49),-255).
point((52,49),-255).
```

### 4.2 Edges

Edges are defined as 3 or more adjacent points, and this allows direction to be extracted from points that are individually directionless. Points are considered adjacent in the directions allowed by the receptive field. They may be adjacent horizontally, vertically or diagonally (two directions of diagonality). For the points asserted from figure 4 the following edge would be asserted.

```
edge((50,50), (52,50), 255,
  [[(50,50),(52,50),[(51,50)]],[(50,49),(52,49),[(51,49)]]]).
```

The edge is from the (50,50) to (52,50) and has a contrast of 255. All the points associated with the edge are retracted when the edge is asserted. The retracted points are stored as a list with the asserted edge. The edge may at some point be retracted, and if so the edge points stored in the list would be re-asserted.

The fragment of the code that asserts edges is given below, in simplified form. Ignoring some of the minor issues such as handedness, an edge is asserted if the edge detector finds a set of edge points adjacent to a set of edge points parallel to it (linear equations must match) but with inverse contrast . When an edge has been found, the edge points are retracted and the edge is asserted. The simplified code fragment presented below is given as representative of the general approach used by the system.

```
edge :-
  edge_detector(PointA,PointB,Contrast,EdgePointsAB),
  edge_detector(PointC,PointD,-Contrast,EdgePointsCD),
  linear_eqn(PointA,PointB,Slope,ConstantAB),
  linear_eqn(PointC,PointD,Slope,ConstantCD),
  ConstantAB>ConstantCD, nextTo(ConstantAB,ConstantCD),
  ContrastEdge is Contrast * sign(Slope),
  retractor(EdgePointsAB), retractor(EdgePointCD),
  assertz(edge(PointA,PointB,ContrastEdge,
          [EdgePointsAB,EdgePointsCD])).
```

### 4.3 Lines

A line is defined as two edges that are parallel and *reasonably* close to each other. For the sake of simplicity we ignore that lines have a thickness. A simplified code fragment is given below.

```
line :-
  edge(PointA,PointB,Contrast,EdgeAB),
  edge(PointC,PointD,-Contrast,EdgeCD),
  linear_eqn(PointA,PointB,Slope,ConstantAB),
  linear_eqn(PointC,PointD,Slope,ConstantCD),
  reasonably_close(ConstantAB,ConstantCD),
  voidcheck(PointA,PointB,PointC,PointD,[EdgeAB,EdgeCD]),
  retractor(EdgeAB), retractor(EdgeCD),
  assertz(line(PointA,PointB,Contrast,[EdgeA,EdgeB])).
```

### 4.4 Verteces

A vertex is defined as the point where two non-parallel lines meet. Vertexes are pseudo-objects that are computed from lines or edges at the point where they meet. They may be used to assert objects but they will not prevent objects from being asserted and if they fall within their bounds. As they do not contradict assertion they are retracted if they fall within the boundaries of an object that is asserted.

### 4.5 Triangles

A triangle is a set of 3 matching lines or vertexes. Again for the simplified sample code, we ignore that lines have thickness and that triangles may have texture.

```
triangle :-
  line(PointA,PointB,Contrast,LineAB),
  line(PointB,PointC,Contrast,LineBC),
  line(PointC,PointA,Contrast,LineAC),
  trianglegeometry(PointA,PointB,PointC),
  voidcheck(PointA,PointB,PointC,[LineAB,LineBC,LineAC]),
  retractor(LineAB), retractor(LineBC), retractor(LineAC),
  assertz(triangle(PointA,PointB,PointC,
          Contrast,[LineAB,LineBC,LineAC])).
```

### 4.6 Processing Flow

As shown by the simplified Prolog code the visual system sets out the space into which objects may be asserted and the geometric nature of that space. This object-space is then seeded by the receptive fields to which the visual input is presented. This introduces low-level facts into the visual system. The processing flow of the visual system is to recursively replace these fact statements with ever more high level replacements. Points are replaced with edges, edges with lines and finally lines with the end product objects, in this case triangles. The system may be said to complete when it reaches a steady state, where no further objects are replaced or where there is circular object replacement.

## 5 Colour

The system defines colour as a simple luminance value. As with natural visual systems, this may be extended by the addition of two further colour contrast values, but at its simplest colour may be said to be a simple luminance value. As visual input is restricted to the output of receptive fields, colour must be computed purely from relative contrast values. These contrast values feed through edges, lines and finally to the final objects of the visual system. Contrast values may be negative as well as positive. Once the visual system completes (that is, reaches a steady state from a given initial sensory input) then normalized colour values can be calculated.

The simplest contrast colour model is derived simply by normalizing negative contrast values. This is done by calculating the baseline colour of white from the lowest negative contrast value. The colour of the canvas is then simply set to the absolute value of the lowest contrast (as final colour values cannot be negative, the lowest contrast value plus the canvas value must equal zero). Figure 5 gives an example how colours are calculated. It is assumed (with respect to the system) that colour is represented by values from 0 to 255. The colour white

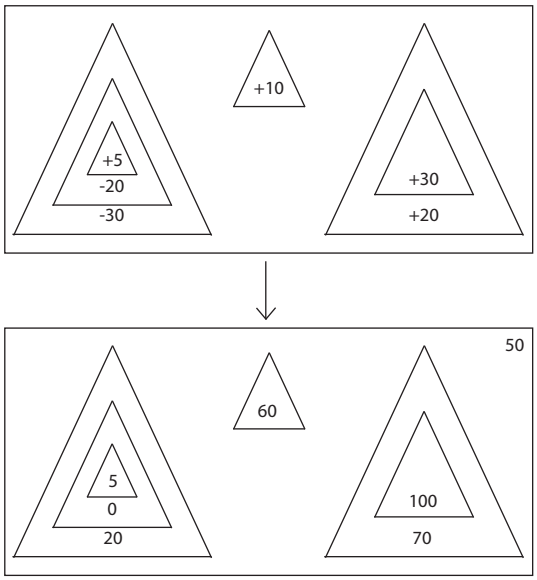


**Fig. 5.** Contrast colour normalization. The top panel shows objects with the raw contrast values as derived from edges and lines. The bottom panel shows the same objects with normalized colour values. The bottom panel has a colour value for the canvas as well as the object, and this is determined by negating the negative raw contrast.

(full luminance) is defined to have the value of 0 and 255 is defined as black (no luminance).

Colours are recursively computed from the contrasts of objects in relation to their background. The default background is the canvas (current page). The colour of the canvas is set to the difference between positive and negative contrast values. For example, if the canvas has a single black triangle, then the canvas would have a colour of 0 (white) and the triangle would have a colour of 255 (black). For a more illustrative example, consider a canvas with two seperate triangles, one with a contrast value of -100 and another with a contrast value of +155. The lowest contrast value is -100 and therefore the normalization value would be +100. The final colours for the two triangles would as a result be 0 (white) and 255 (black), and the canvas would be 100 (light gray). Figure 5 give a slightly more complex example which involves recursion. The nested triangles give a combined minimum contrast value of -50, and therefore the normalization value would be +50.

## 6 Kanizsa Figures and Illusory Contours: an extended example

One of the strengths of the visual system is that it provides the theoretical basis for perception of illusory contours and implicit objects (Kanizsa figures)[8][9].

When presented with figure 6 human observers generally see three notched black triangles. However, when the black triangles are rotated so as to obscure their tips a human observer will generally now see a single large white triangle in place of the small triangular notches. Such figures are called subjective figures or illusory figures because they are visible only on the basis of how they obscure other figures. Moreover, such figures have an interesting aside in that despite there being no empirical difference in luminance a human observer will report that the subjective figure is brighter than its surroundings. The area where the brightness changes (there being no change in the original image itself) is called an illusory contour (or edge).

Figure 6 shows three black on white triangles with white on black triangular notches on one side. Under the visual system presented here an object will not complete (be *recognized*) if any of the objects that fall within its borders are unaccounted for (that is, contradict it). The white on black notches lead to the assertion of two white on black edges and a white on black vertex, and these will prevent the larger black on white edges from completing a triangle. There is also a break in the line, which also prevents the triangle from completing. It will fail because a bridging line cannot be asserted to complete one of the lines for the triangle (lines cannot simply be asserted as needed). Therefore the black on white triangle will fail. However, the two white on black edges and the vertex are sufficient to allow the small white on black triangle to be asserted. It has no line and has a contrast colour of white on black (the contrast colour of the two edges). Assertion of the small white triangle now removes the objects that initially blocked completion of the larger black on white triangle. A bridging line can now be asserted because the space is a void (covered by the small white on black triangle) where anything can be asserted as needed. This allows line completion, and as there are no other objects that remain to contradict it the black on white triangle completes and is asserted.

Figure 7 differs from figure 6 only in that the black triangles have been rotated so that the tip of the triangle is obscured rather than the base. In this case the small white on black triangles will complete as before. However, completion of the 3 small white on black triangles does in this instance not allow the larger black on white triangles to complete. This is because the vertexes lie outside of the void left by the asserted triangles. The entire image therefore fails to complete. Failure leads to backtracking and eventually to the retraction of the 3 small white on black triangles. Examining alternatives, the three vertexes formed by the white on black edges line up correctly to form a large triangle which completes without complications. The void left by the assertion of a large triangle now allows double bridging lines to be asserted to complete the black on white triangles.

It is important to note that colours are relative contrast values rather than absolute. These values are calculated from the contrast values that are passed down from edges and then adjusted globally when the image completes. The large triangle in Figure 7 is asserted on the basis of 3 vertexes (which have no contrast value) and without any associated edges or lines. This means there is

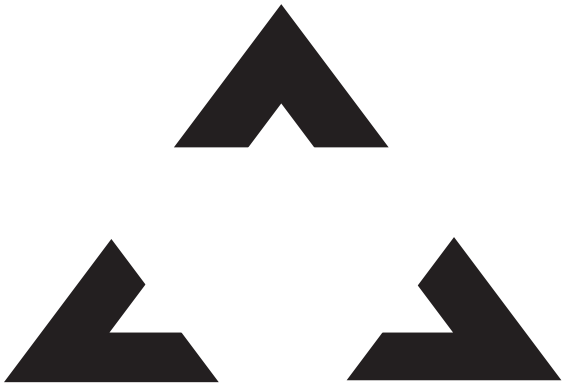

**Fig. 6.** Three black on white triangles with triangle notches.

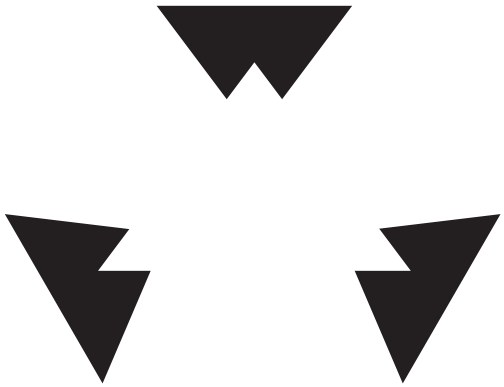

**Fig. 7.** The same notched triangles as in figure 6, but rotated so that the triangular notch obscures the tip of the triangle. Most human observers will now perceive an additional white triangle, whose 3 vertexes cover the tips of the 3 black triangles.

no contrast value to assign to the triangle. The solution to this is to simply set the contrast value to the default(-1). Without the induced triangle the colour normalization would result in a white canvas with 3 black triangles. However, the induced white triangle has a contrast colour of -1, which makes it the lowest colour contrast value on the canvas, and is therefore assigned as reference white. As a result the background canvas is set to a colour of 1. In this way the colour of the inferred Kanizsa triangle may be seen to be *whiter than white*[8].

## 7 Related Work

One of the few researchers who has made an attempt to develop the framework of a visual system is Marr[10]. Marr attempts to give an overview of the entire human visual system. While Marr's work is ambitious in scope it fails in provid-

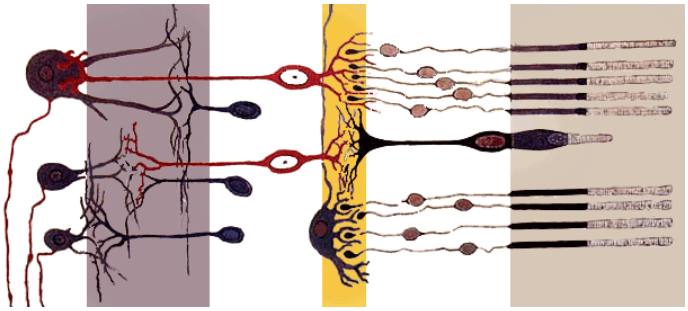

**Fig. 8.** Structure of the human retina.

ing a detailed and functional model of visual systems. Marr's work has not led to the development of effective artificial visual systems.

## 8 Discussion

Conventional edge detection algorithms take a bitmap image as an input and produce a bitmap image of the same dimensions as output. While the visual input for the human visual system is equivalent to a bitmap image, the human visual system does not do pixel based global transforms. All retinal sensors are directly organized into receptive fields in the retinal itself via a small set of specialized neurons, which lie directly behind the visual sensory neurons. As can be seen from Figure 8 the neural structure of the retina is not overly complex, with sensory neurons organized by a small number of specialist intermediary neurons, whose output is sent to the brain[11]. While it is not universally agreed what the function of the specialist neurons involved is, it is generally accepted that they organize the individual sensor neurons into receptive fields, and that the output of receptive fields is then used to carry out some form of edge detection function. This is made more precise in this paper by setting out the dedicated function of calculating segment contrast values and integrating them into individual receptive fields.

One of the corollaries of the segmented contrast hypothesis is an explanation for why two types of receptive field are found natural visual systems. As the the two types of receptive field are inversely related, they will operate as a pair – that is, an edge will always attract both types of receptive field, one for each side (or hand) of the edge. It might appear from this that only a single type of receptive field is necessary to carry out edge detection. Evolution rarely indulges in redundancy which serves no useful function, and therefore it follows that the second inverse receptive field is likely to have some necessary function for the visual system. Indeed this function becomes apparent when we seek to compute edges from points and lines from edges. Edges are derived from sets of adjacent points. Edges have direction (or angle), which is information that individual receptive field do not generate. It can, however, be derived from the

direction in which a set of two or more points is adjacent. An edge is a contrast difference boundary, and this boundary can be of two types, positive contrast or negative contrast. This handedness (or chirality) is also information that is not generated by individual receptive fields. It is also not derivable from a set of adjacent points. Indeed it cannot be resolved from the information provided by a single type of receptive field. A second type of receptive field, one which provides inverse contrast values, allows resolution of whether an edge contrast is positive or negative. This is therefore sets out the reason why two types of receptive field are necessary and as a result commonly found in natural visual system.

Using the precise imaging model set out by PostScript has allowed the rigorous definition of the representational space of the visual system. One of the primary features of the visual system is to formally describe how the abstract representational space of the visual system is linked to the space of the visual input. The space set out by the receptive fields is equivalent to a bitmap and is a two dimensional integer space. The PostScript imaging model allows this two dimensional integer space of the visual input to be transformed into a space for abstract objects by simply converting integers to real numbers. This space of real numbers is then represented by the use of Prolog, which effectively transforms what was intially a two dimensional bitmap into a set of logic statements. The coordinate system of the initial visual input is maintained but objects are represented by abstract logic statements. These logic statements exist within a logical space, in which geometric relationships recursively transform known facts into hypothetical solutions. This is a logical search space, and the visual processing that may be said to be done in this space has very little relation to the pixels and bitmap that compose the original input. Visual processing activity consists solely of the inference mechanisms of mathematical logic and the rules of geometry. Nevertheless, the final objects are set out using the coordinates that match the coordinates of the initial bitmap input and the abstract objects may be accurately rasterized by translation into PostScript.

## 9 Future Work

The visual system is not yet fully implemented in Prolog. However, the logic for the system is complete and simply needs to be implemented.

PostScript has a very comprehensive imaging model that is designed to be able to represent a significant subset of what the human visual system is able to represent. PostScript is of course limited to two dimensional static images whereas the human visual system functions in four dimensions; 3 dimensional space as well as time. Nevertheless, a visual system that would incorporate all of the PostScript imaging model (Bézier curves in particular) would be a significant step in producing a more general visual system that would approach the functionality of the human visual system.

The system is at present designed to process bitmap images that are of good quality, without noise or distortion. At present minor noise such as random

pixels are simply ignored because they are not detected by the receptive fields. Minor distortion is also corrected by the geometry margin of error set into the geometry rules. It is unclear at present how significant distortion or noise will be dealt with. It is expected that this is to be a significant area of research.

## 10 Conclusion

TriangleVision is a toy visual system, but one which within its limited parameters is a complete visual system. Its visual field is restricted to triangles, but in this limited visual environment it may be said to genuinely *see* what is presented to it. As the visual environment is further restricted to triangles produced by the PostScript imaging model, we are able to formally define what it means to *see* as the ability to reconstruct PostScript code from a bitmap image. This ability can be accurately measured by simply comparing the original PostScript code that produced the raster output with the PostScript code produced by the system from the bitmap input.

The ability of the visual system presented here to provide a theoretical model for illusory contour and Kanizsa figure perception is a strong indication that this artificial visual system operates in a way that is directly related to the way the human visual system functions. The contrast model of colour employed is able to fully account for the phenomenon of illusory contours.

## References


1. Hubel, D.H.: Eye, Brain, and Vision. Scientific American Library, New York (1988)
2. Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P.: Winning the darpa grand challenge. Journal of Field Robotics (2006) accepted for publication.
3. Adobe: PostScript Language Reference Manual (2nd Edition). Addison-Wesley Professional, Boston, MA, USA
4. Adobe: PostScript Language Tutorial and Cookbook. Addison-Wesley Longman Publishing, Boston, MA, USA (1986)
5. Clocksin, W., Mellish, C.: Programming in Prolog. Springer-Verlag, Berlin (1984)
6. Giannesini, F., Kanoui, H., Pasero, R., van Caneghem M: Prolog. Addison-Esley, Wokingham, England (1986)
7. Grossberg, S., Mingolla, E.: Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading. Psychological Review **92**(2) (1985) 173–211
8. Kanizsa, G.: Subjective contours. Scientific American **234**(4) (1976) 48–52
9. Kanizsa, G.: Organization in Vision: Essays on Gestalt Perception. Praeger Publishers, New York (1970)
10. Marr, D.: Vision. W.H. Freeman and Company, New York (1982)
11. Cajal, S.R.y.: Histologie du Système Nerveux de l'Homme et des Vertébrés. Maloine, Paris (1911)