

1  
2 Gen



# Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

MODELO TEORICO DE PERCEPCION VISUAL DE  
PROFUNDIDAD. UN ENFOQUE COMPUTACIONAL

## Tesis Profesional

Que para obtener el título de  
INGENIERO EN COMPUTACION

Presentan

GABRIEL MAURICIO ALVAREZ MEDINA  
SERGIO RAMON CARDENAS GARCIA  
CARLOS PALOMINO HADDAS



Director :

DR. CARLOS ROLANDO LARA Y ZAVALA

México, D. F.

1985



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## Indice

<b>1. Introducción</b>	<b>0</b>
1.1 Descripción del Problema	0
1.2 Objetivos del Proyecto	2
1.3 Organización de la Tesis	4
<b>2. Antecedentes Generales</b>	<b>6</b>
2.1 Introducción	6
2.2 Sistema Visual en Vertebrados	7
2.3 Mecanismos de Visión en Anfibios	11
2.4 Modelos Teóricos de la Estereopsis	19
2.5 Visión en Robots	25
<b>3. Modelo Teórico Propuesto</b>	<b>31</b>
3.1 Introducción	31
3.2 Descripción del Modelo	31
3.2.1 Elementos del Modelo	32
3.2.2 Descripción Matemática	34
3.3 Implementación del Modelo en Computadora Digital	35
3.3.1 Elección de Herramientas de Cómputo	35
3.4 Características del Modelo Implementado	36
3.4.1 Modelo para Dos Dimensiones	36
3.4.2 Una implementación para tres dimensiones	42
3.5 Simulación en Computadora Digital	50
3.5.1 Formas de Despliegue de Resultados	50
3.6 Experimentos Realizados	61
3.7 Discusión de Resultados	89
<b>4. Análisis de sensibilidad</b>	<b>94</b>
4.1 Introducción	94
4.2 Metodología	95
4.3 Simulación en Computadora Digital	96
4.4 Discusión de los Resultados	115
<b>5. Conclusiones</b>	<b>122</b>
5.1 Evaluación del Modelo	122
5.2 Comparación y Contraste del Modelo con Otros Modelos	123
5.3 Etapas Futuras	125
<b>I. Manual de Usuarios del Sistema</b>	<b>127</b>
<b>II. Listado del Programa de Simulación</b>	<b>130</b>
<b>III. Mecanismos de Graficación</b>	<b>157</b>

## 1. Introduccion

### 1.1 Descripcion del Problema

Mediante la visión (sentido de la vista) le es posible a la mayoría de los seres vivos determinar características de los objetos que les rodean tales como forma, tamaño, color, distancia de ellos al objeto y distancia entre objetos; además de elementos como textura, brillantéz y otras características físicas propias de los objetos, específicas de cada cuerpo.

En el presente trabajo estudiamos un aspecto del fenómeno de visión llamado percepción de profundidad.

La percepción de profundidad es la cualidad que permite al observador estimar la distancia que guarda un objeto con respecto a su posición en el espacio. A través de ella se puede determinar, además de la distancia, el espesor de un objeto.

La importancia específica de esta cualidad es el aportar al observador una idea clara del espacio tridimensional en que se encuentra. Esta idea le proporciona la capacidad de movimiento coherente en el espacio y es un elemento importante en la determinación de su comportamiento.

Las imágenes que el sujeto recibe del medio son captadas a través de los elementos sensores que son los ojos. Estas imágenes, son recibidas, principalmente, por los fotorreceptores (detectores de luz) en la retina de ambos ojos; sin embargo la información que contiene cada ojo por separado, está muy limitada para lograr una determinación correcta de la profundidad.

De lo anterior surge la pregunta de cómo, a partir de información binocular captada en las retinas oculares, obtiene el

## Introducción

sistema nervioso una buena imagen en tercera dimensión, con la que logra determinar en la forma más precisa posible los parámetros de profundidad y espesor.

Para ello, se han propuesto una serie de teorías sobre percepción de profundidad que pretenden explicar este fenómeno, éstas se fundamentan en experimentos empíricos realizados en diversas preparaciones biológicas (animales). Muchas de estas teorías tienen como base el fenómeno de la estereopsis binocular (percepción de la profundidad por el efecto de la disparidad de la imagen captada por las dos retinas) y como elemento cooperativo, la acción monocular que recae en la acomodación de lentes (cristalino). La utilización de éstos mecanismos como medios para determinar la profundidad también está apoyada por experimentos en animales.

Ahora bien, se sabe que los animales hacen uso de éstos dos mecanismos para la percepción de profundidad y en base a ello se han creado modelos que tratan de explicar la naturaleza global del fenómeno.

Si bien los modelos que se han desarrollado han logrado reproducir aproximadamente los resultados experimentales, no se ha encontrado uno que se ajuste completamente a lo que sería la realidad, siendo la causa más importante de esto, el que no se conozca a fondo el fenómeno de la determinación de profundidad.

La existencia de un modelo que se aproxime lo más posible a la realidad, permitiría predecir conductas y ayudaría al entendimiento general del sistema visual de los animales.

El modelo presentado en este trabajo está basado en los

## Introduccion

estudios realizados sobre anuros (sapos y ranas). Estos animales presentan ventajas para el estudio de su sistema visual, ya que una buena parte de su conducta está regida por el medio que ven, y ésto ha ocasionado genéticamente que específicamente en la retina se preprocese mucha más información que en las de otros animales superiores (por ejemplo el gato o el hombre). Además, dado que son animales básicamente estáticos, puede estudiárseles sin prácticamente alterar su medio ambiente.

Otra ventaja que simplifica su estudio está en que los fotoreceptores se encuentran distribuidos prácticamente sobre toda la superficie de la retina, pues carecen de una zona con más concentración de éstos llamada fóvea, que permite a animales superiores tener mucha mayor resolución visual.

Los modelos existentes presentan problemas en cuanto a su implementación física [7] o problemas de velocidad de resolución [4] (son demasiado lentos pues necesitan iteraciones de filtrado).

Lo anterior sugiere la necesidad de un modelo que sea lo suficientemente completo, pero realizable, con una velocidad de convergencia rápida al estado final, existiendo entonces la posibilidad de implementarse en software y hardware. Esto permitiría entonces a algún dispositivo electrónico (como por ejemplo un robot) moverse libremente en medios variantes, gracias a la información proporcionada por su sistema visual en tiempo real.

### 1.2 Objetivos del Proyecto

Los objetivos concretos del proyecto desarrollado en ésta tesis pueden listarse de la siguiente manera:

- 1) El primer objetivo del trabajo es el desarrollo e

implementación en computadora digital de un modelo teórico de percepción de profundidad basado en estereopsis binocular y acomodación.

Se pretende que el modelo contenga los elementos fundamentales del sistema visual conocidos hasta ahora, además de presentar características que permitan su implementación. Se intenta que el modelo, al contener los elementos básicos de la visión permita simular posibles comportamientos. Se desea, además, vencer las dificultades encontradas en modelos anteriores (dificultad para manejar el modelo debido a su complejidad -como el modelo de House [4]-, o modelos que proponen la existencia estructuras biológicas muy complejas -modelo de Marr y Poggio [7]-).

2) Una vez identificados los elementos que constituyen al modelo, analizar las relaciones entre ellos a través de pesos dados a cada uno mediante las constantes involucradas, y de esta forma aproximar al modelo a las condiciones reales que indican los resultados experimentales.

3) Realizar simulaciones con diferentes conjuntos de constantes en el modelo que permitan realizar experimentos de índole predictiva haciendo variaciones en sus valores, para de esta manera ver cual es el resultado de modificar alguna o algunas variables en el proceso de la percepción de profundidad (por ejemplo el factor de acomodación)

Que sucedería si el animal (o el dispositivo electrónico) se quedase con un sólo sensor óptico, o si su mecanismo de acomodación no funcionara.

4) Por último se debe evaluar el modelo propuesto con respecto

a otros modelos existentes. Deben analizarse las ventajas y desventajas, así como las aportaciones dadas y posibles mejoras.

### 1.3 Organizacion de la Tesis

El presente trabajo se encuentra organizado de la siguiente manera:

En el capítulo 2 se presentan algunos conceptos y términos necesarios para comprender mejor el modelo propuesto. Estos conceptos se refieren a aspectos biológicos y fisiológicos de los mecanismos de visión tomados desde un punto de vista general, y después particularizando con el caso de los anfibios. Después se describen otros modelos teóricos ya realizados sobre estereópsis, que tienen vinculación con éste trabajo. Finalmente se revisa alguna literatura sobre visión en robots y la necesidad que existe de métodos eficientes de percepción de profundidad y de espesor para autómatas.

En el capítulo 3 se describen los elementos funcionales que conforman al modelo y después se hace una descripción matemática del proceso de percepción de profundidad por estereópsis utilizando éstos elementos. Una vez explicado el modelo matemático se muestra su implementación hecha en computadora digital, con la que se realizaron simulaciones bajo diversas condiciones de visión, y finalmente se analizan los resultados obtenidos.

Debido a que con el modelo propuesto se pretende, además de reproducir lo obtenido por la experimentación, realizar algunas predicciones, en el capítulo 4 se describe el análisis de sensibilidad de los parámetros que conforman el modelo, y se muestran las conclusiones a las que se llegó.

## Introduccion

En el último capítulo (5) se analizan los resultados obtenidos con el modelo, se comparan sus logros y carencias con los de otros modelos alternativos y finalmente se contempla su utilidad a futuro.

## 2. Antecedentes Generales

### 2.1 Introduccion

El trabajo actual, aunque compete a la Ingeniería en Computación, se desarrolló en el campo de la Bioingeniería, área que interrelaciona a la ingeniería con otras disciplinas, como lo es la biología en sus diferentes ramas (e.g. anatomía, fisiología, etc.). Debido que aún en lectores familiarizados con el tema pueden existir lagunas sobre algunos de los conceptos y elementos biológicos inherentes al sistema visual que se manejan en el presente trabajo, en ésta sección se describen en forma breve las estructuras más importantes que participan en la visión y se explican sus funciones.

La descripción del sistema visual se hace primero para los vertebrados en general, mostrando los elementos que les son comunes a la mayoría. En éste primer enfoque, no se hace énfasis sobre alguna estructura en especial dentro del sistema visual, aunque más adelante el trabajo se enfoca al área de la retina y a las células que la conforman.

En la segunda sección se hace una descripción más o menos detallada de la visión en los anfibios, que nos presenta las particularidades de la retina y de las células ganglionares, las cuales explican la forma detallada de las estructuras que en forma de bloques más generales se manejan dentro del modelo propuesto de estereópsis.

Creemos que otro antecedente de peso para este trabajo lo es la visión en robots, ya que se pretende que lo aquí desarrollado pueda servir como una base para la implementación de sistemas visuales aplicables a la robótica. Por ello se hace un análisis de

## Antecedentes Generales

las necesidades que en la actualidad existen en materia de visión por computadora y sus posibles aplicaciones.

### 2.2 Sistema Visual en Vertebrados

En el ojo del vertebrado, la estructura fundamental (fig. 2-1) es un globo ocular más o menos esférico, situado en una cavidad llamada órbita a ambos lados de la caja craneana, y unido al cerebro por un nervio óptico, situado en un tallo que nace de la superficie interna del globo ocular. Este tiene simetría radial bastante aceptable; su eje principal va de dentro a afuera. En su interior se encuentra una serie de cámaras llenas de líquidos acuosos o gelatinosos. Entre éstas cámaras, en situación anterior, encontramos una lente llamada cristalino. Las paredes del globo ocular tienen fundamentalmente tres capas, que son, de fuera hacia adentro: la esclerótica, la coroides, y la retina. La esclerótica es una esfera completa; la coroides y la retina son incompletas hacia afuera. Las dos primeras capas son de origen mesenquimatoso, y desempeñan fundamentalmente funciones de sosten y nutrición; la retina (que en realidad es doble) es la zona sensitiva del sistema ocular. En el polo externo del globo ocular, la esclerótica se modifica y forma junto con la piel que la cubre una córnea transparente. Hacia afuera también, la coroides y la retina se unen y modifican. Frente a los bordes del cristalino, estas dos capas suelen ensancharse para formar un cuerpo ciliar donde se fija el cristalino. Hacia adelante, a partir de ese punto, las dos capas ya unidas se incurvan paralelamente al cristalino y forman el iris, con una abertura central: la pupila.

**RETINA.** Todas las demás partes del ojo son las "ayudantes" de la retina; su tarea consiste en llevar como es debido los rayos luminosos y enfocarlos sobre la retina, que a su vez los recibe y

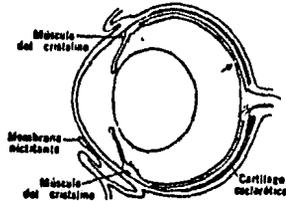


Figura 2-1: Ojo de una rana  
Romer (1983)

lleva al cerebro los estímulos visuales. El aspecto general de la retina se observa en la figura 2-2. Por dentro de la capa pigmentada (que toca a la coroides) existe una zona con estriaciones perpendiculares; luego vienen (yendo hacia adentro) tres zonas separadas que contienen objetos circulares que pueden identificarse como núcleos celulares. La zona estriada contiene los extremos alargados de las células receptoras de la luz, llamadas bastones y conos; la zona granulosa externa contiene los cuerpos celulares y los núcleos de éstas estructuras. La zona granulosa media corresponde a células bipolares que transmiten hacia adentro impulsos procedentes de bastones y conos, y a tipos accesorios de células nerviosas retinianas, como las amacrinas y las horizontales, las cuales transmiten los impulsos en otras direcciones (por ejemplo en forma lateral). Los núcleos más internos pertenecen a las células ganglionares, que reciben los estímulos de los elementos bipolares, células horizontales y amacrinas, y cuyas fibras van al cerebro a través del nervio

Óptico.

Los bastones y conos, cuyo nombre se refiere a su forma habitual, son los fotoreceptores; cada célula tiene un extremo sensitivo dirigido a la coroides, una porción ensanchada, y una pieza basal que contiene el núcleo. Estos tienen funciones muy distintas, ya que los conos se concentran en el centro del campo visual, y los bastones, en general, en la periferia. Los bastones desempeñan su función en la luz tenue; los conos sólo pueden intervenir cuando la iluminación es buena. Los conos, como grupo, dan imágenes muy detalladas, y los bastones dan una imagen más borrosa. Los conos dan la sensación de color, y los bastones imágenes en blanco y negro.

Los conos y los bastones pueden presentarse en cualquier región de la retina, pero cuando hay conos suelen concentrarse en la fovea central (pequeña depresión) en la parte posterior de la retina. Por dentro de los conos y bastones se encuentra una capa de células bipolares con prolongaciones que recogen, por un lado, impulsos de conos y bastones, y por otro los transmiten a las células amacrinas, a las horizontales y hacia dentro a la tercera capa formada por células ganglionares, las cuales dan origen al nervio óptico. A menudo un cono aislado hace contacto con una cierta célula bipolar, y esta célula bipolar única se une también con una sola célula ganglionar; con esta disposición, cada cono puede tener una "línea privada" en el nervio óptico hasta el cerebro. En cambio para los bastones la regla es que un gran número de ellos converjan sobre una sola célula bipolar; por tanto, el cerebro no puede saber cuál es el bastón de este grupo que ha sido estimulado; esto explica la falta de precisión de la visión de los bastones en comparación con la que procede de los conos.

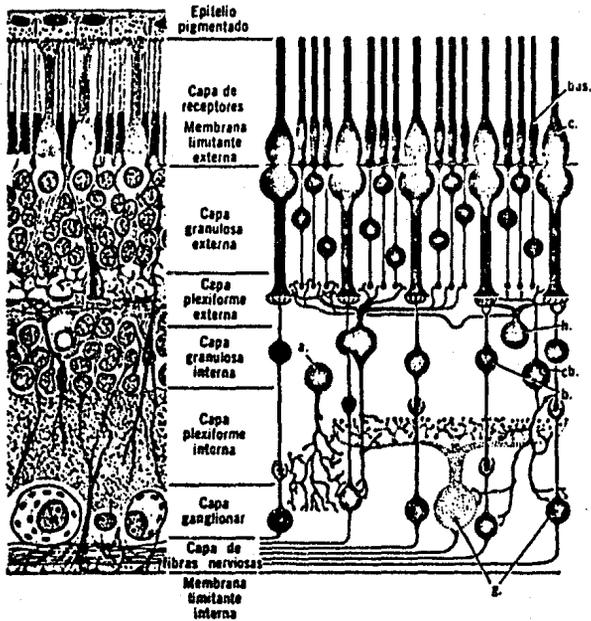


Figura 2-2: Conexiones dentro de la retina: b, células bipolares unidas a conos únicos; bas, bastones; c, conos; cb, células bipolares unidas a varios bastones; a y h, células horizontales y amacrinas; g, células ganglionares  
Romer (1983)

## Antecedentes Generales

**NERVIO OPTICO.** Las células ganglionares, situadas en la superficie interna de la retina, producen largas fibras que convergen en un punto situado cerca del centro de la retina, atraviesan esta capa (habiendo ahí un punto ciego por falta de conos y bastones) y se dirigen al cerebro a lo largo del tallo del ojo, formando el nervio óptico. Cuando llegan al piso del cerebro anterior, los nervios ópticos dan lugar al quiasma óptico, en forma de X (fig. 2-3); la mayoría de las fibras del nervio óptico derecho pasan al lado izquierdo del cerebro, y viceversa; este cruzamiento de fibras se llama decusación. En la mayor parte de los vertebrados los dos conjuntos de fibras siguen hacia arriba y atrás hasta llegar a los centros cerebrales en el techo del cerebro medio (técum).

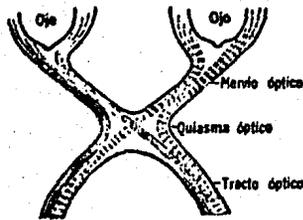


Figura 2-3: Esquema del quiasma óptico  
Romer (1983)

### 2.3 Mecanismos de Vision en Anfibios

El ojo de los anfibios se diferencia por algunas particularidades de los ojos de otros vertebrados. El tamaño de los ojos en estos animales es considerable. Lo mismo sucede con el

### Antecedentes Generales

tamaño de la parte del cerebro que se refiere a la vision en relación con sus otras partes. De ésto se concluye que una parte muy importante en la vida de estos animales son los ojos y la conducta guiada visualmente. En la figura 2-3 se ve el diagrama de un ojo de una rana y el de un humano. El radio de la superficie exterior de la cornea del ojo de la rana mide de 4-5 mm en la region de la pupila. El lente es relativamente grande, la distancia entre la córnea y la retina sin embargo es pequeña en relacion al grosor de la retina. La calidad del medio ocular en el ojo de la rana es pobre comparada a la del ojo humano; consecuentemente, para pequeños estímulos (2.15 grados) el contraste de la imagen en la retina depende no sólo del contraste físico de la imagen y el estímulo, sino también del tamaño del objeto. El lente en el ojo de los anfibios enfoca la imagen en la retina.

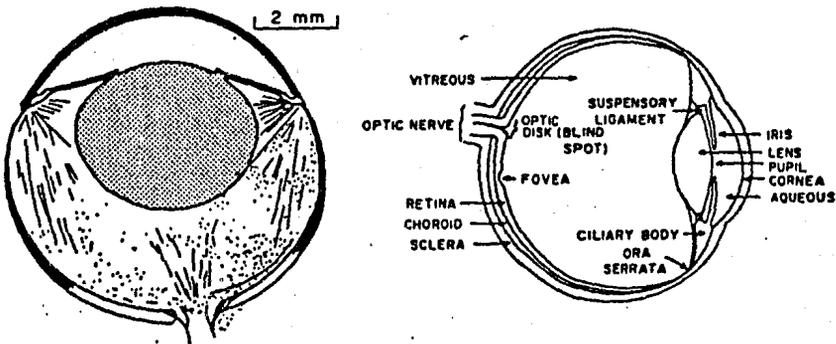


Figura 2-4: Ojo de rana (a) y de humano (b)  
(Lara, 1981)

Varias técnicas ópticas y neurofisiológicas han sido usadas para medir el campo visual del anuro. Schneider (1954) y Fite (1973) usaron métodos ópticos para ver la imagen proyectada en la

## Antecedentes Generales

pupila y determinaron el tamaño del campo visual monocular y binocular de anfibios de diferentes especies. Gaze y colaboradores (1958) usaron técnicas electrofisiológicas para medir el campo monocular visual encontrado entre los vertebrados; campos que, en algunas especies de sapos alcanzan casi 360 grados de ángulo visual (fig. 2-5). El gran tamaño del campo visual se debe a la posición periscópica de los ojos y al gran ángulo de divergencia entre los ejes visuales de los mismos. Estudios de mapeos paramétricos han mostrado que los anuros tienen un gran campo de visión binocular, el cual se extiende en los sapos cerca de 40 o 55 grados debajo del meridiano horizontal y se expande hacia arriba para incluir un área de 160 a 170 grados directamente sobre la cabeza.

Debido a que los anfibios no tienen movimientos oculares significantes, existe una relación relativamente invariante entre el lugar geométrico de la retina y la porción del campo visual, relativa al eje del cuerpo de la cual surge la información visual. Los mayores cambios en la fijación ocurren solo como resultado de cambios en la orientación del cuerpo entero relativo al estímulo visual, dado que los movimientos discretos de la cabeza son relativamente también poco frecuentes en las ranas. El lugar geométrico del estímulo, así pues, puede llevar información relativamente específica referente al lugar geométrico espacial del objeto físico relativo al eje del cuerpo.

La retina recibe la luz que entra a través de sus fotorreceptores, y sus fibras eferentes son axones de diferentes tipos de células ganglionares. En las células retinales se encuentra una compleja organización interna constituida principalmente de células bipolares, amacrinas, horizontales y ganglionares (Fig. 2-6).

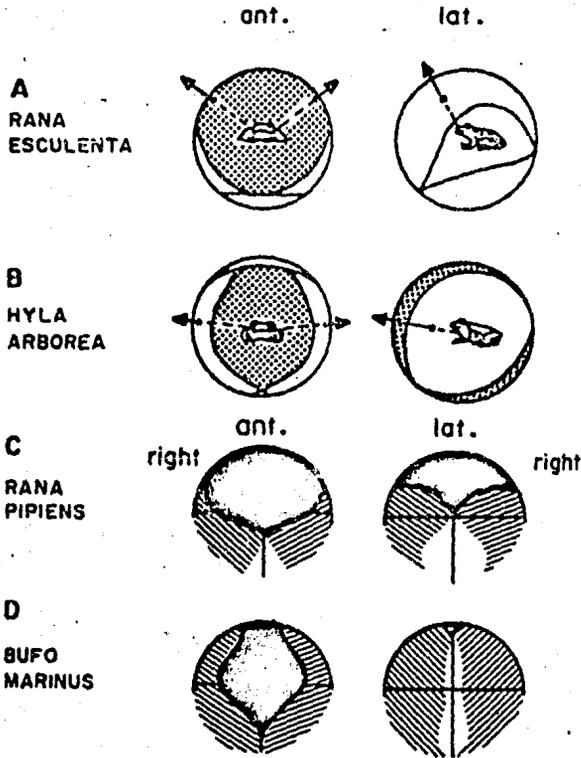


Figura 2-5: (a) y (b). Campos visuales monocular (blanco) y binocular (punteado) de una rana. Pipiens (c) y B. Marinus (d). (partes binoculares oscuras, parte fuera del campo visual blanca). (De Fite, 1973)

CONOS Y BASTONES

CELULAS HORIZONTALES  
Y VERTICALES

CELULAS GANGLIONARES

FIBRAS DEL NERVIO  
OPTICO

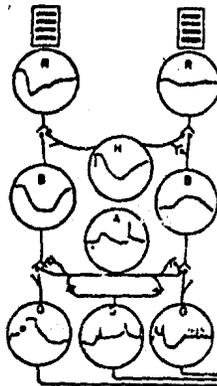
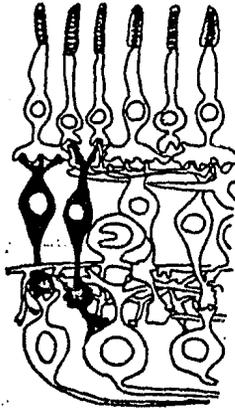


Figura 2-6: Anatomía y fisiología de las células retinales (De Shepherd 1970)

## Antecedentes Generales

Hartline (1940) definió tres tipos de células ganglionares: neuronas "on", neuronas "on"- "off", y neuronas "off". El definió al campo receptivo de una célula ganglionar dada como la región de la retina que debe ser iluminada para obtener una respuesta a dicha neurona. Mástarde Maturana y colaboradores (1959, 1960) definieron fisiológicamente 4 tipos de células ganglionares: 1) Detectores de bordes sostenidos, 2) detectores de bordes convexos, 3) detectores de contraste cambiante, y 4) detectores de luz baja. Una clasificación más reciente, y de alguna manera mayor de Grusser y Grusser-Cornehls (1976) definen 6 tipos de células ganglionares. Brevemente, las propiedades de estas células son las siguientes:

**Neuronas de clase 0 (neuronas "on").** Estas células tienen un campo receptivo excitatorio (CRE) que va de 0.4-1 mm (15 grados) en la retina. Pequeños puntos de luz proyectados en el CRE generan una corta respuesta a la activación con una fuerte componente transitoria, la cual es seguida por un aumento más o menos pronunciado de la tasa de impulso neural. La iluminación combinada del CRE y el campo receptivo inhibitorio (CRI) da una respuesta más débil comparada a aquella cuando el CRE está excitado. Los estímulos de luz roja proyectados en el CRE generan una fuerte activación transitoria, mientras que la luz azul lleva a una respuesta sostenida la cual puede durar hasta dos minutos.

**Neuronas de clase 1.** Estas son equivalentes a los detectores de bordes sostenido de Maturana. Las neuronas de éste tipo no muestran ninguna respuesta a la iluminación difusa u obscurecimiento del campo receptivo completo en el ojo adaptado a la luz. Un pequeño punto de luz proyectado al CRE genera predominantemente una activación sostenida. El CRE es global a menudo, su tamaño se extiende de 1.5 hasta máximo 4 grados. Un CRI

## Antecedentes Generales

rodea al CRE pero la inhibición generada por el contraste brillante u obscuro de un estímulo moviéndose hacia el CRI es significativamente menor que en las neuronas de clase 2. Las neuronas de clase 1 se proyectan al tectum óptico contralateral en las ranas, donde las terminales sinápticas terminan en las dendritas de células tectales dentro de la capa más superficial del tectum óptico, estas fibras no se proyectan en el tectum en los sapos.

Neuronas de clase 2. Estos son los detectores de bordes convexos de Maturana. El CRE de éstas células tiene 2.5-5 grados de diámetro, a menudo son ovals o de formas irregulares y rodeados por un gran CRI. El diámetro del CRI se extiende, dependiendo del tipo del estímulo, de 20 a 45 grados. La mayoría de las unidades de clase 2 no responde a la estimulación de "on-off" del campo receptivo completo de la retina adaptada a la luz. Algunas unidades, sin embargo, muestran claramente una respuesta de "off" retardada a una tasa baja de retardo y una gran latencia entre 2-15 segundos. En la mayoría de las neuronas la respuesta a un estímulo moviéndose del centro del CRI es más fuerte que el movimiento centrífugo del centro del CRE hacia el CRI. La sensibilidad máxima se encuentra en el centro del CRE. Cuando un estímulo pequeño contrastante se detiene dentro del CRE, muchas neuronas de clase 2 continúan descargándose por varios segundos. Esta fuerte inhibición explica porqué la respuesta a grandes bordes contrastantes moviéndose simultáneamente através del CRE y CRI es muy baja o completamente ausente. La forma del estímulo móvil se hace relevante para la respuesta neuronal si el estímulo se extiende dentro del CRI.

Neuronas de clase 3. Estas neuronas son las neuronas de

#### Antecedentes Generales

"on"- "off" de Hartline (1949) y Barlow (1953). Son llamadas detectores de contraste cambiante por Maturana et. al. (1960). Su CRE se extiende a cerca de 6-10 grados y tiene en la mayoría de los casos una forma oval regular. Se encuentra rodeada por un CRI de 12-20 grados de diámetro. El tamaño pero no la forma del estímulo móvil influencia al proceso neural en tanto que el estímulo sea menor que el CR. Cuando se introduce un objeto en el CR de éstas neuronas, dan una respuesta "on"- "off". Los estímulos pequeños contrastantes que se mueven por pasos através del CRE facilitan una pequeña cantidad de actividad neuronal la cual es máxima en el centro del CRE. La estimulación del CRI produce una inhibición más débil que en las células de la clase 2. Estas células presentan propiedades adaptativas muy lentas. Por esta razón, éstas neuronas en combinación con las células de clase 0 y 1 son buenos candidatos para sentir el mundo estático. Los axones de las neuronas de clase 3 probablemente forman terminaciones en todas las partes centrales del sistema visual del cerebro de la rana.

**Neuronas de clase 4.** Estas son las neuronas de "off" de Hartline y Granit a las que Lettvin et. al. (1959) llamaron detectores de obscurecimiento. Este nombre se escogió porque hay una activación más o menos pronunciada de éstas neuronas ocasionada por la reducción de la iluminación retinal. El CRE es redondo u oval y se extiende hasta 10-15 grados de diámetro. Las neuronas de clase 4 responden a cualquier objeto móvil mayor de 2-5 grados que lleva a un obscurecimiento del CRE donde el obscurecimiento total es el factor esencial y no el tamaño, forma o contraste del estímulo. Estas células se proyectan al tectum contralateral y al pretectum con componentes ipsi y contralaterales.

**Neuronas de clase 5.** Estas células se encuentran raramente en

## Antecedentes Generales

el tétum y la retina. Tienen un grán CR, cuya frontera es difícil de definir. Su activación es proporcional al valor recíproco de la intensidad de luz promedio que cae dentro del CR. Sólo responden a cambios en la iluminación y no del todo a pequeños estímulos oscuros en movimiento. Maturana et. al. (1960) llamaron a éstas neuronas, detectores de oscuridad.

### 2.4 Modelos Teóricos de la Estereopsis

Se han propuesto diversos modelos de percepción de profundidad considerando elementos biológicos que probablemente utilizan los animales y que han podido ser determinados a través de la experimentación en laboratorio. Hasta ahora, los modelos más completos recurren a los mecanismos binoculares, utilizando estereópsis y los monoculares, utilizando acomodación del cristalino como elementos fundamentales. Estos elementos no son los únicos ya que se ha propuesto también la convergencia en animales foveados como auxiliar para determinar profundidad de los objetos.

En el caso particular de los anuros, los cuales presentan en un grado mínimo el fenómeno de vergencia, los modelos se han orientado hacia los mecanismos monoculares y binoculares.

Ingle (1976) demostró experimentalmente la importancia de la monocularidad en anuros, encontrando que en la región ipsilateral existe un grado muy pequeño de error (  $\pm 5\text{mm}$  ) en la determinación de profundidad de un objeto. Por otra parte, Collett demostró que la información binocular juega un papel importante realizando experimentos con prismas en los anuros.

Por lo anterior, los modelos que presentaremos son los más significativos considerando monocularidad y binocularidad como

elementos básicos.

#### Modelo Amari-Arbib

Este modelo se caracteriza por presentar un esquema de cooperatividad y competencia. En él, existen dos campos, uno inhibitorio y otro excitatorio que se encuentran en continua interacción.

Para el caso simplificado de una retina unidireccional, el modelo gráfico está representado por la figura 2-7:

La profundidad en el campo excitatorio es proporcionada muy probablemente por la información del sistema de acomodación para cada región de la retina.

Las regiones en el campo excitatorio que presentan información de profundidad y posición retinal semejante o cercana presentan un fenómeno de excitación o cooperatividad mutua.

El campo excitatorio muestra la existencia de una serie de posibles profundidades para una posición retinal. alguna de estas profundidades tendrá un peso superior a las otras posibles, siendo éste función de los estímulos exteriores. Además, el campo inhibitorio realiza una suma de todas las profundidades existiendo una sola entrada retinal en el inhibitorio para cada entrada en el excitatorio.

Todos los pesos de las profundidades son sumados y entonces se realiza una realimentación negativa al campo excitatorio por el campo inhibitorio. Lo anterior ocasiona que el punto de más peso sea el que sobreviva y posiblemente será la profundidad más cercana a lo real, la cual se decodifica para la posición retinal

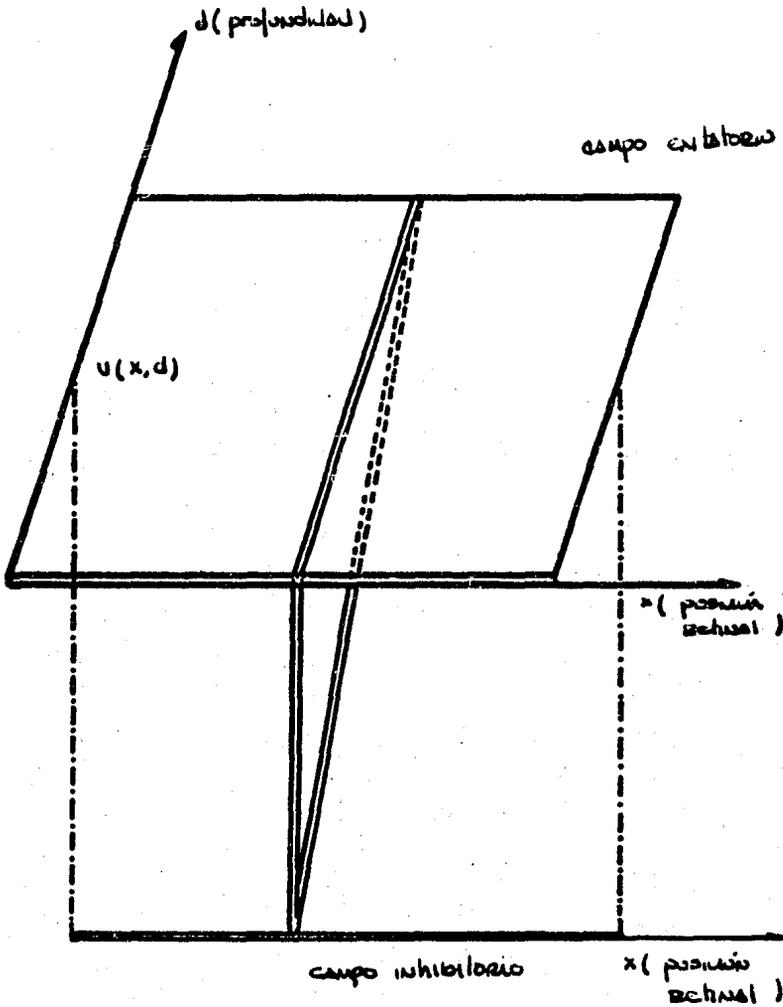


Figura 2-7: Modelo Simplificado de Amari-Arbib

especificada.

El siguiente diagrama 2-8 representa el resultado del proceso de cooperatividad y competencia (dado por la realimentación) cuando se tiene una estimulación fuerte de la estimulación en cierta posición retinal.

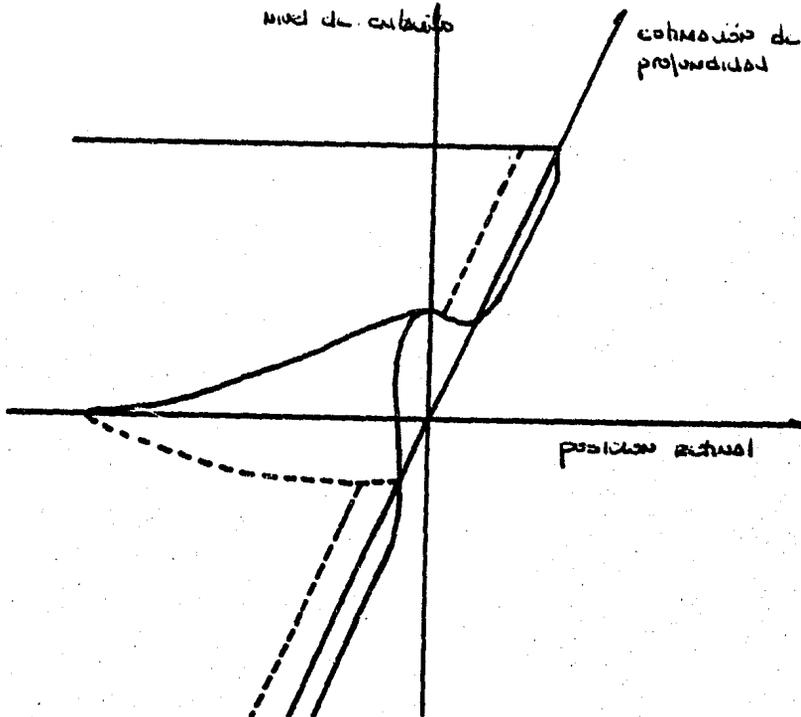


Figura 2-8: Efecto neto de un estímulo exterior en el modelo de Amari-Arbib

Si el problema de percibir se plantea desde un inicio, y se toman en cuenta los resultados experimentales, se llega a la conclusión de que existen dos métodos de solución: la monocularidad y la binocularidad. Con esta idea, se desarrolla un método más completo que toma las bases presentadas en el modelo anterior. El modelo de Donald H. House realiza estas mezclas, pretendiendo con

#### Antecedentes Generales

ello generar un método que sea más representativo de la realidad. Se considera la utilización de 2 planos del modelo de Amari-Arbib, interactuando entre sí, cada uno proporcionando información que permita converger a información real de profundidad. Un mapa representa la información que se ha obtenido por un mecanismo monocular (acomodación del cristalino) El otro mapa representa la información que se ha obtenido por mecanismos binoculares.

Se podría considerar la existencia de un mapeo 1 a 1 entre las regiones del plano monocular y las regiones del plano binocular.

Es posible configurar el modelo de tal forma que la falla de algún mecanismo (monocular o binocular) no impida la determinación de la profundidad de los objetos, obteniendo el mecanismo que sobreviva un peso sobresaliente. En forma esquemática el modelo tendría la apariencia de la figura 2-9

Una característica de éste modelo es el fenómeno de segmentación del espacio visual del animal en regiones angulares, donde un punto dentro de una región determina la excitación de toda la región. Lo anterior impide lograr un mapa de profundidad que sea totalmente acorde con la realidad, teniendo siempre márgenes de error.

Existen modelos que permite lograr mapas de profundidad con una grán exactitud hacia la escena que se presenta en la realidad. El modelo de Marr y Poggio propone la interacción entre las fibras retinales de ambos ojos, las fibras interactúan en un campo de proyección donde se codifican la dirección y profundidad de los estímulos por medio de la disparidad entre la información que proporciona cada retina. Lo anterior es el esquema que sigue cualquiera de los métodos de campo de proyección. El algoritmo de

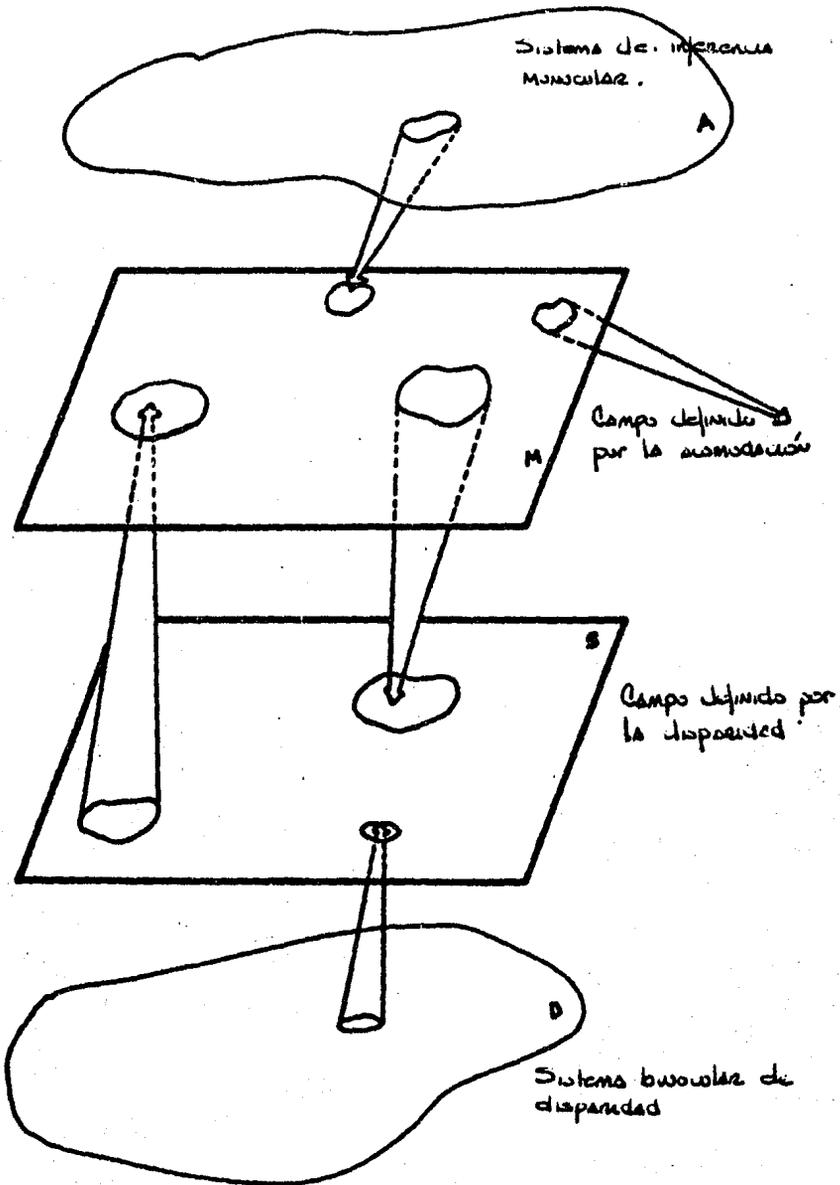


Figura 2-9: Modelo de Percepción de Profundidad de Donald H. House

## Antecedentes Generales

estos autores propone que cada punto en el espacio pueda ser asignado a un conjunto particular de fibras retinales lo cual implica que se tendría una gran exactitud en la determinación de la profundidad, inclusive se lograría determinación puntual, pero se requeriría tener un gran número de elementos receptores en cada retina, situación que parece alejarse de la realidad.

El modelo de Marr y Poggio tiene también el problema de la generación de imágenes o puntos fantasma, siendo un caso crítico en su situación por el número de componentes que considera. Estos puntos fantasma surgen al existir más puntos de intersección entre las fibras retinales que objetos reales. Su modelo propone que los puntos pueden ser eliminados a través de una acción cooperativa entre los puntos que codifican las mismas profundidades y una acción competitiva entre puntos de intersección que codifican profundidades diferentes. Gráficamente su modelo sería como se muestra en la figura 2-10

Si bien el sistema es exacto en cuanto a la determinación de profundidad, es inaplicable a la realidad y presenta problemas (fantasmas) como los demás modelos. Su modelo no considera la ayuda de los mecanismos monoculares siendo por tanto incompleto, además de generar los posibles errores.

### 2.5 Vision en Robots

Un objeto importante de estudio de la inteligencia artificial, lo constituyen los robots, los cuales se pretende que puedan realizar tareas que emulen las hechas por los seres humanos. Entre otras cosas se ha hecho necesario que éstos posean la capacidad de percibir su entorno para poder actuar de acuerdo a lo captado. Dependiendo de las necesidades de información que se tengan, será

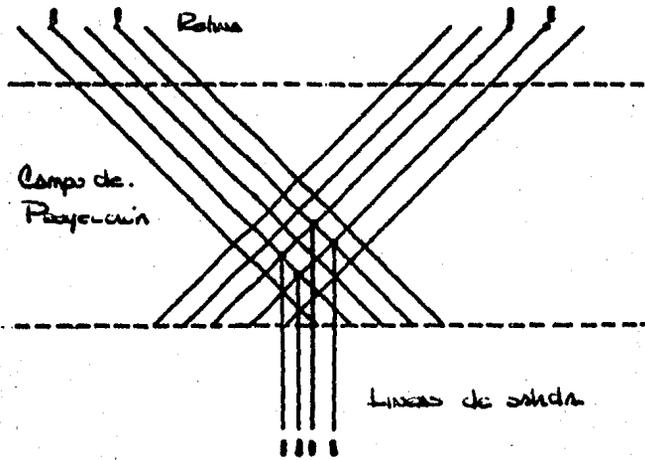
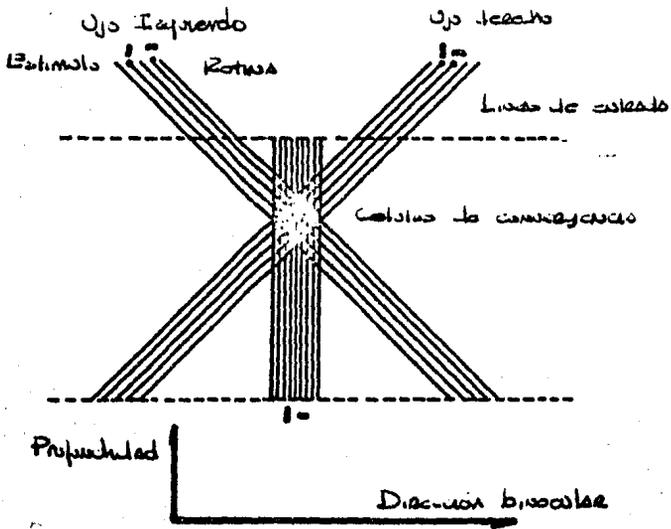


Figura 2-10: Modelo de Percepción de Profundidad de Marr y Poggio

## Antecedentes Generales

la complejidad del tipo de sensores utilizado. Así entonces, existen robots, que se bastan de un simple interruptor para sentir el medio, mientras que los hay que requieren una gran cantidad de información sobre la tarea a ejecutar, y entonces se requiere de sensores que les permitan actuar en ambientes desconocidos o variantes. En general, se pueden distinguir cuatro tipos significativos de sensores: de proximidad, tacto, esfuerzo y visión, siendo de estos últimos de los que hablaremos.

Sin un sistema de visión un robot puede encontrar un objeto a "tintas"; y el tacto puede ser usado si el objeto está aproximadamente en buena posición y orientación, la visión es necesaria si el objeto o los objetos están colocados aleatoriamente en el plano. La visión ofrece también ventajas en cuanto a rapidez y precisión con respecto al tacto.

El sentido de visión en los robots es el elemento esencial que permite a éstos presentar características de adaptabilidad, flexibilidad y capacidad de organización. Los mecanismos que tienen visión, en general son considerados como más "inteligentes", ya que tienen una retroalimentación sensorial fina que hace ampliar sus capacidades. Sin embargo, en la actualidad existen mecanismos de visión para robots bastante limitados, y por ello son necesarios sistemas de propósito general.

En la actualidad los sensores de visión más utilizados son: cámaras de televisión en base a tubos Vidicon, matrices de dispositivos acoplados por carga CCD (Charge Coupled Devices) y sensores ultrasonoros [8].

Los sistemas de visión artificial constan en general de los siguientes elementos:

- . Sensor de las señales de la escena. Formador de la imagen primaria (óptica o ultrasonora) del objeto.
- . Convertidor de la imagen primaria en señales adecuadas a los aparatos que harán su tratamiento.
- . Representación de la imagen. Transformación que pone de relieve la información significativa (de acuerdo a la tarea).
- . Extracción de características pertinentes por medio de algoritmos o criterios heurísticos.
- . Reconocimiento de la imagen caracterizada mediante comparación con características "aprendidas" (registradas en memoria)
- . Decisión de acción. Una vez reconocida total o parcialmente la imagen debe ser tomada una decisión de acción a efectuar por el robot.
- . Control de robot, es decir, ejecución de la decisión tomada en función del reconocimiento de la imagen.
- . Si la tarea no ha sido completamente ejecutada, deberá ser analizada otra imagen primaria producida por el sensor.

Analizando los elementos de un sistema de visión artificial puede notarse que su estructura esta íntimamente ligada a las características morfológicas y topológicas que son relevantes en las imágenes de una aplicación dada. De ahí la dificultad de concebir sistemas generales de visión artificial de aplicación a robótica.

El objetivo de los sistemas de visión por computadora aplicados en robótica es el de transformar el conjunto de datos contenidos en una imagen digitalizada, en una descripción concerniendo los objetos que se encuentran dentro del universo de acción del robot. Dicha descripción será utilizada por los programas que controlan el movimiento del manipulador. Existen

## Antecedentes Generales

varias aplicaciones potenciales de la visión en robótica que incluyen el reconocimiento y localización de objetos, control de las operaciones del robot y la inspección.

La visión en robótica es especialmente recomendable en aplicaciones industriales, exploración de medio ambiente desconocido ó en aplicaciones médicas. La mayor parte de trabajos actuales en visión se encaminan hacia las aplicaciones industriales.

Los postulados de la visión en robótica son los siguientes :

1. Hay objetos en el mundo real.
2. Las imágenes son un medio de obtener informaciones sobre estos objetos.
3. Las informaciones sobre los objetos se tienen con un objetivo.

La extracción de contornos de una imagen es una operación preliminar fundamental para la realización de varias tareas de visión artificial, entre otras podemos dar como ejemplo el reconocimiento de formas visuales, la inspección automática, la localización de un objeto en un plano para la eventual prensión y manipulación de dicho objeto por un robot dotado de visión, etc. Por medio de la extracción del contorno exterior de la imagen de un objeto, se realiza en efecto una segmentación de la imagen, con el fin de compactar la información. Por otro lado, toda la información de forma y dimensión, que nos interesa obtener de la imagen de un objeto, esta contenida en su contorno.

Dentro de la visión artificial, un campo importante es el de

## Antecedentes Generales

la percepción de profundidad, para el cual sin embargo, no se han hecho muchos avances, ya que los robots actuales generalmente se usan en la industria donde tienen un campo de acción limitado. Pero si se requiere que el robot funcione de tal manera que sea lo más parecido a un ser viviente, ya sea por el ambiente en el que se desenvuelve ó por lo delicado de sus funciones, entonces su forma de visión se debe tornar más elaborada, y por esto la investigación de métodos para lograrlo debe realizarse en forma conjunta con la investigación biológica. Este trabajo precisamente, pretende proporcionar algunas bases para lograr tal objetivo, aunque éstas solamente son desde un punto de vista teórico, dado que entre sus metas está el abarcar tanto el campo computacional, como el biológico.

### 3. Modelo Teorico Propuesto

#### 3.1 Introduccion

Tomando como base los elementos y conceptos abarcados en el capítulo anterior, podemos centrarnos en los objetivos de nuestro trabajo. En este capítulo abordamos aspectos fundamentales de la tesis como son: la descripción del modelo que proponemos, su implementación en computadora, la realización de simulaciones y el análisis de los resultados obtenidos.

Dentro de la descripción del modelo se incluye una visión esquemática del sistema, mostrando los elementos que lo conforman así como los procesos que éstos realizan. Mas tarde se dá una descripción matemática de estos procesos.

La parte de implementación cubre aspectos de ingeniería tales como la elección de las herramientas de cómputo empleadas y las características del modelo implantado, primero para un ambiente visual de dos dimensiones y luego para su generalización a tres.

En la sección de simulación en computadora digital, definimos la forma en la cual el modelo implantado despliega sus resultados, así como las características y variaciones de los experimentos de simulación realizados.

Se concluye el capítulo con una sección de análisis e interpretación de los resultados obtenidos a través de los diversos experimentos.

#### 3.2 Descripción del Modelo

### 3.2.1 Elementos del Modelo

En el presente trabajo, proponemos un modelo de percepción de profundidad por estereópsis binocular que postula que la profundidad de los objetos está codificada por la frecuencia de disparo de un grupo de células determinada por la disparidad entre la proyección del punto en la retina derecha con el de la izquierda.

La estructura básica del modelo se puede ver en la figura 3-1 donde se observa que cada punto del espacio se proyecta en un punto en la retina y ese punto a su vez se proyecta al tectum del animal. Cada célula del tectum recibe a su vez la proyección de fibras del otro ojo que pueden codificar una profundidad específica. Nótese que mientras más a la derecha está el punto de proyección, un número menor de profundidades pueden codificarse y, por tanto, un número menor de fibras del otro ojo se proyectan. El postulado básico de este modelo es que la actividad conjunta de la proyección del ojo derecho con el izquierdo codifica una profundidad especificada por la disparidad entre las dos. Sin embargo, es necesario que de varias fibras activas, se defina cual de ellas es la que realmente esta codificando la distancia del objeto. Para ello, nosotros hemos propuesto que además del sistema de disparidad los anfibios pueden utilizar mecanismos monoculares como la acomodación para acelerar la convergencia de la definición de la profundidad de los objetos. Asimismo, postulamos que existe un proceso de cooperatividad entre células vecinas para definir la profundidad del objeto en cuestión y que también ayuda a la convergencia del proceso así como a su fiel representación de los objetos externos.

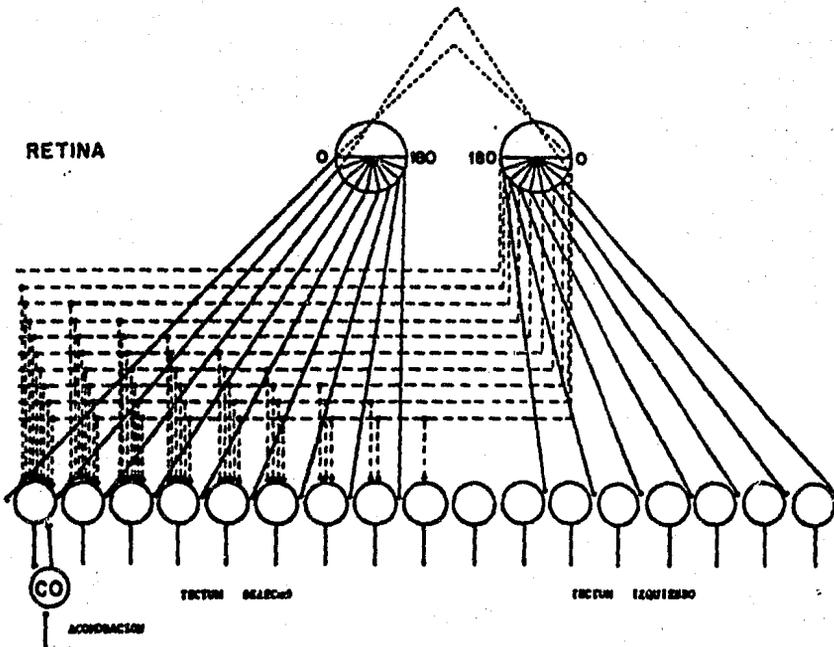


Figura 3-1: Modelo de percepción de profundidad por codificación de la frecuencia neuronal

### 3.2.2 Descripcion Matematica

La ecuación matemática que define estos procesos se puede definir de la siguiente forma:

$$\frac{dw(i,j)}{dt} = \text{TRL}(i) \times \text{TRL}(j) - k3 | w(i,j) - w(i+1,j) | +$$

$$- k3 | w(i,j) - w(i-1,j) | +$$

$$- k1 | \text{disp}(i,j) - \text{ac}(i) | +$$

$$- k2 \left[ \sum_{k=1}^n \text{TRL}(k) \right] - \text{TRL}(i)$$

donde  $w$  es el peso de la sinápsis que combina a las fibras del ojo izquierdo  $i$  con las fibras del ojo derecho  $j$ ;  $\text{TRL}$  es el transmisor liberado dado por:

$$\text{TRL}(i) = I(i) \cdot w(i)$$

donde  $I$  es el potencial de acción de la neurona;  $\text{disp}$  es la profundidad que codifica la disparidad entre la célula  $i$  del ojo izquierdo con la célula  $j$  del ojo derecho;  $\text{ac}$  es el factor de acomodación monocular del punto  $i$ . La acomodación es un factor que depende de la profundidad del objeto y la podemos definir de la siguiente forma:

$$\frac{d\text{ac}}{dt} = f(d) \times \left[ 1 - e^{-|f(d) - \text{ac}|} \right]$$

donde  $f(d)$  es una función cuyo máximo valor esta dado cuando el punto está en foco; en esta forma, la acomodación tomará éste valor y codificará la profundidad del objeto. El término de la sumatoria representa cómo las fibras no activas del ojo son decrementadas en un factor mayor que el de las fibras activas.

lograndose con ello una distinción mayor entre ellas. Los pesos  $w(i,j)$  que sobrepasen cierto valor umbral son los que serán tomados en cuenta para definir la profundidad del punto; mientras que los que sean menores del promedio serán eliminados. Finalmente se suman los resultados de los pesos del ojo derecho y del izquierdo para dar como resultado la codificación final.

Más adelante, en este mismo capítulo y en el siguiente se ve la función que desempeñan las constantes  $k_1$ ,  $k_2$  y  $k_3$ .

### 3.3 Implementacion del Modelo en Computadora Digital

#### 3.3.1 Eleccion de Herramientas de Computo

Para el manejo adecuado del modelo teórico basado en elementos biológicos se hace necesaria una herramienta tal, que permita estudiarlo y analizar fácilmente sus resultados así como poder modificar sus parámetros y realizar predicciones. Dicha herramienta deberá además permitir a personas no involucradas directamente con el tema, el comprender el modelo y poder hacer uso de él en una forma sencilla.

La computadora digital se muestra como el instrumento idóneo para cumplir con estos objetivos, además se cubre automáticamente otro aspecto relevante: el de la contribución del modelo a la visión por computadora (Robótica)

El propósito básico de la mayor parte de los estudios de simulación es comparar alternativas. Por tanto, el programa de simulación debe ser lo suficientemente flexible como para acomodarse con facilidad a las alternativas que se considerarán. Como a menudo es imposible predecir con exactitud cuales alternativas interesantes no serán cubiertas durante el curso del

estudio, es esencial que se introduzcan al programa la flexibilidad y las disposiciones necesarias para contemplar modificaciones rápidas y simples. Lo anterior debe reflejarse en la elección del equipo de computación y el lenguaje de programación que ha de usarse.

Tomando en cuenta que se requería un lenguaje transportable, disponible en la generalidad de los equipos -ya que esto permitiría implementar el modelo en casi cualquier instalación-, se optó por utilizar el lenguaje PASCAL, que facilita la programación de un modelo que utiliza tanto cantidades numéricas, como estructuras de datos, permitiendo debido a su forma estructurada, el diseñar y probar en forma independiente a los componentes de todo el sistema.

Respecto a las facilidades de equipo, se considero importante que las instalaciones contaran con dispositivos de graficación adecuados y con un software eficiente, ya que había la necesidad de desplegar resultados gráficos que ayudarían a comprender el modelo.

Afortunadamente se tuvo libre acceso a instalaciones de este tipo en el I.I.H.A.S. - U.N.A.M., donde se encuentran el sistema Foonly F2, que permitió implementar el software, y la computadora graficadora Chromatics de alta resolución que permitió obtener los despliegues gráficos.

### **3.4 Características del Modelo Implementado**

#### **3.4.1 Modelo para Dos Dimensiones**

La simulación en computadora se realiza en la siguiente forma:

Se tienen doce puntos en cada retina y se consideran coordenadas cartesianas cuyo origen está en el ojo izquierdo.

## Modelo Teorico Propuesto

Se definen entonces  $n$  puntos localizados en el campo visual del animal; el presente modelo sólo considera a los puntos que se encuentran en el campo binocular del animal. Una vez definidos los puntos, el algoritmo presentado en la sección anterior se activa determinando así la serie de sinápsis que van a codificar la profundidad de los objetos. A continuación se da una explicación de los elementos más importantes que componen al programa que implementa al modelo:

1. Captura de datos. Se aceptan los diferentes datos de entrada al programa, necesarios para la simulación, los cuales son:

**DISTOJO:** Distancia Interocular. Esta distancia entre los ojos estará representada en unidades arbitrarias proporcionadas por el usuario. Este valor es importante, ya que en base a él se realizan los cálculos de distancias, y además proporciona la escala para los valores de las coordenadas de los puntos que existan en el espacio visual.

**K1, K2 y K3:** Constantes que influyen en el funcionamiento de la simulación (ver análisis de sensibilidad). Estas constantes corresponden a "acción de otras", "acción de acomodación" y "acción de vecinas" respectivamente.

**(x1, y1), (x2, y2), ..., (xn, yn):** Son las coordenadas cartesianas de los puntos que el usuario define para el espacio visual. El eje  $x$  está definido por la recta que pasa por los puntos focales de los ojos. El origen está en el ojo izquierdo por el cual pasa también el eje  $y$ . La escala está dada por el valor de DISTOJO.

**Uso de lentes:** Se da la opción de simulación de lentes, los cuales pueden ser de acercamiento o alejamiento aparentes. Se debe

especificar el ángulo de desviación de la luz producido por los lentes. éste ángulo puede tomar cualquier valor real, pero sólo tiene un valor significativo si es menor de 30 grados.

ii. Valores Iniciales. En base a los puntos leídos (usando solo aquellos que caen dentro de la región binocular) se marcan las neuronas que deberán estar activas. Se tiene en cuenta para ésto la acción de los lentes (si es que los hay). Ver figura 3-2. La información acerca de las neuronas activas se encuentra en los vectores VTRLI y VTRLD para los ojos izquierdo y derecho respectivamente. Un valor de cero en la posición  $i$ -ésima de cualquiera de estos vectores nos indica que la neurona  $i$ -ésima del ojo en cuestión está inactiva. Un valor de uno nos indica actividad en dicha neurona.

Para cada neurona activa existe un valor de acomodación o enfoque que es proporcional a la distancia que existe de la neurona al punto que hizo que se activara. Este valor se almacena en el vector VFAC.

Un punto localizado en el espacio binocular activa solamente a una neurona del ojo izquierdo y a otra del derecho. El modelo supone la existencia de cierta información genética que relaciona a un par de neuronas -una de cada ojo- con una distancia. Esta información esta almacenada en la matriz MDISTANCIAS. El elemento MDISTANCIAS[ $i,j$ ] contiene un valor aproximadamente igual a la distancia a un punto que active simultáneamente a la neurona  $i$ -ésima del ojo izquierdo y a la  $j$ -ésima del ojo derecho. Sólo se toma en cuenta el triángulo superior de esta matriz donde  $i+j$  es menor que el número de neuronas de un ojo; ya que el inferior contiene combinaciones de neuronas que no pueden activarse

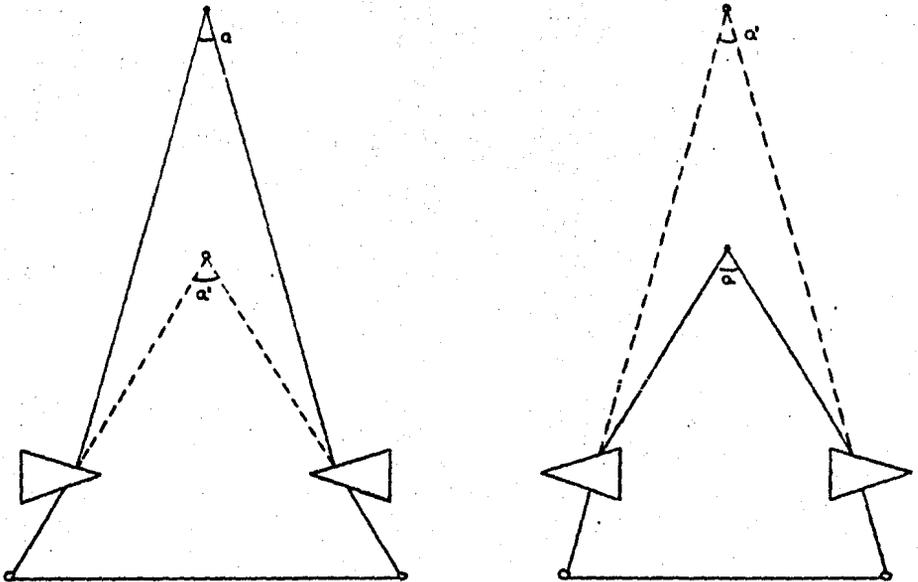


Figura 3-2: Efecto de los lentes sobre la imágen percibida

simultáneamente con un punto.

iii. Parejas Activas. El algoritmo se inicia con la determinación de todas las combinaciones posibles de los pares de neuronas activas en ambos ojos.

Lo anterior se realiza por medio del producto:

$$MDER[i, j] = DISTOJO * VTRLI[i] * VTRLD[j]$$

donde MDER es la matriz que irá variando durante todo el algoritmo. Existen dos matrices de este tipo, una para cada ojo (MDERI para el ojo izquierdo y MDERD para el derecho). En estas matrices tampoco se toma en cuenta el triángulo inferior (donde  $i+j$  es mayor que la dimensión de la matriz)

iv. Acción de Otras. El siguiente paso consiste en darle un peso mayor a las neuronas activas de forma que exista un mayor contraste con los de las no activas. En el ojo izquierdo ésto se lleva a cabo de la siguiente forma

$$MDERInueva[i, j] = MDERI[i, j] - SUMA1$$

donde  $SUMA1 = SUMA - K1 * VTRLI[i]$

donde  $SUMA =$  Suma de todas los valores de VTRLI

Como se observa en las expresiones anteriores, en realidad se resta un término a todos los elementos de MDERI, pero si la neurona  $i$ -ésima del ojo izquierdo está activa, el término restado es menor. Para el ojo derecho ésto se realiza en forma similar.

v. Acción de Vecinas. La fase siguiente consiste en comparar

## Modelo Teorico Propuesto

los valores ya asignados a cada neurona con los de sus vecinas a la izquierda y a la derecha ,de manera que una neurona cuyo peso sea muy diferente al de sus vecinas se verá más afectada que otra con vecinas que tengan pesos similares. Lo anterior se logra :

Para el ojo izquierdo:

$$\begin{aligned} \text{MDERInueva}[i, j] &= \text{MDERI}[i, j] - \\ &K2 * |\text{MDERI}[i, j] - \text{MDERI}[i+1, j]| \\ &|\text{MDERI}[i, j] - \text{MDERI}[i-1, j]| \end{aligned}$$

Para el ojo derecho:

$$\begin{aligned} \text{MDERDnueva}[i, j] &= \text{MDERD}[i, j] - \\ &K2 * |\text{MDERD}[i, j] - \text{MDERD}[i, j+1]| \\ &|\text{MDERD}[i, j] - \text{MDERD}[i, j-1]| \end{aligned}$$

vi. Acomodación. Como se recordará, a cada neurona de cada ojo se le asigna un valor de acomodación o enfoque. A las neuronas no activas se les asigna un valor arbitrario muy grande. En esta etapa del proceso se hace una comparación entre el valor de acomodación y los valores de distancias definidos por una neurona de un ojo con las del otro. Las combinaciones que tengan un valor más parecido al de acomodación serán más favorecidas. Para el ojo izquierdo:

$$\begin{aligned} \text{MDERInueva}[i, j] &= \text{MDERI}[i, j] - \\ &K3 * |\text{MDISTANCIAS}[i, j] - \text{VFAC}[i]| \end{aligned}$$

vii. Visibles. En esta etapa, para el ojo izq., se obtiene el máximo de cada renglón de MDERI. Si el máximo de algún renglón no es mayor que un umbral, entonces se hace igual a cero todo el renglón. Para el ojo derecho se hace lo mismo pero con las columnas de MDERD.

viii. Prueba. En esta etapa iterativa se obtiene el promedio de cada renglón de MDERI y se resta a cada elemento del renglón. Los valores que no pasen de un umbral se hacen cero. Lo mismo se hace con MDERD pero con las columnas.

ix. Suma. Al sumar finalmente las matrices MDERI y MDERD, se obtiene una matriz que representa la contribución de los ojos para la percepción de profundidad.

### 3.4.2 Una implementación para tres dimensiones

La implementación del modelo limitada al plano es suficiente para mostrar el grado con el que el modelo se apega al comportamiento del sistema real y delimitar parámetros e iteraciones necesarias. Es por esto que la mayoría de las simulaciones se llevo a cabo con el modelo limitado a dos dimensiones para simplificar el trabajo y la interpretación de resultados. No obstante para completar la simulación y poder trabajar con ambientes más ricos y variados, se procedio a realizar la generalización para tres dimensiones en base a la implementación para dos.

Para realizar esta generalización se nos presentaron dos alternativas principales. Una consiste en modificar los elementos del modelo matemático, aumentando algunos terminos que tomarian en cuenta la existencia de interacciones entre celulas vecinas localizadas ya no en una fila sino en una matriz. Cada ojo tendría asociada una matriz bidimensional de células (por ejemplo en lugar de un vector de doce células, ahora tendrá una matriz de doce por doce). La otra alternativa -que fue la que seguimos- consistio en tomar la implementación ya existente para dos dimensiones, sin modificar el modelo, con la diferencia de que los puntos sobre los

### Modelo Teorico Propuesto

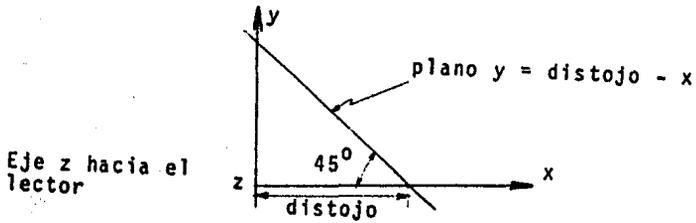
que actua ya no estan limitados a un solo plano, sino que se localizan en un volumen en el espacio aunque casi plano. De esta manera, el modelo para dos dimensiones se aplica repetidamente, cambiando de región, de forma que los volúmenes casi planos se suman formando la región espacial binocular total.

El espacio binocular esta definido de la siguiente manera (utilizando coordenadas cartesianas  $x,y,z$ ):

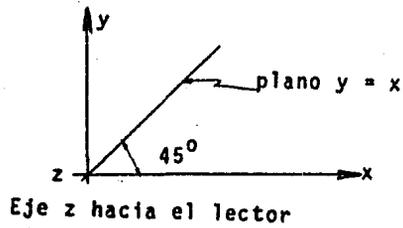
Los ojos estan alineados sobre el eje  $x$ , estando el punto focal del ojo izquierdo en el origen, y el derecho sobre la parte positiva del eje  $x$  con su punto focal a una distancia  $DISTOJO$  del origen. El plano  $xy$  representa el plano frontal de visión (plano horizontal) y el eje  $z$  sirve para representar alturas de objetos. El espacio binocular esta definido solo para valores positivos de  $y$ . Como en el modelo bidimensional se tomo un ángulo horizontal de 90 grados de visión para cada ojo, se considero conveniente el tener un ángulo también de 90 grados de visión vertical para cada ojo. Asi pues el espacio binocular esta limitado por los siguientes planos :

Modelo Teorico Propuesto

$y = \text{distojo} - x$  (plano paralelo al eje  $z$  con traza en el plano  $xy$  como se indica en la siguiente figura)

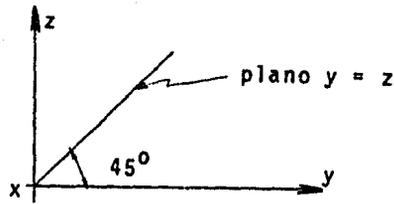


$y = x$  (plano paralelo al eje  $z$  con traza en el plano  $xy$  como se indica en la figura)



## Modelo Teorico Propuesto

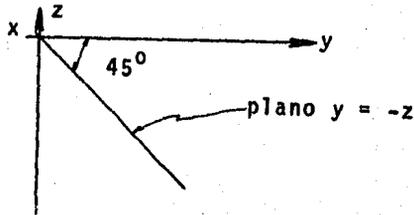
$y = z$  (plano que contiene al eje  $x$  con traza en el plano  $yz$  como se indica en la figura)



Eje  $x$  hacia el lector

$y = -z$  (plano perpendicular al anterior que lo corta en el eje  $x$ )

Eje  $x$  hacia el lector



Para mantener una misma proporción con el modelo bidimensional se dividió el espacio tridimensional en doce regiones (tal como se dividió el plano visual abarcado por un ojo en doce secciones, una para cada fotoreceptor). Las regiones están delimitadas por planos que contienen al eje  $x$ . Cada plano guarda un ángulo de  $90/12$  grados =  $7.5$  grados con el de la siguiente región, las trazas de los planos mencionados en el plano  $xy$  aparecen en la siguiente figura 3-3:

Se puede apreciar que las regiones definidas son rebanadas delgadas del espacio binocular. Como se mencionó, la

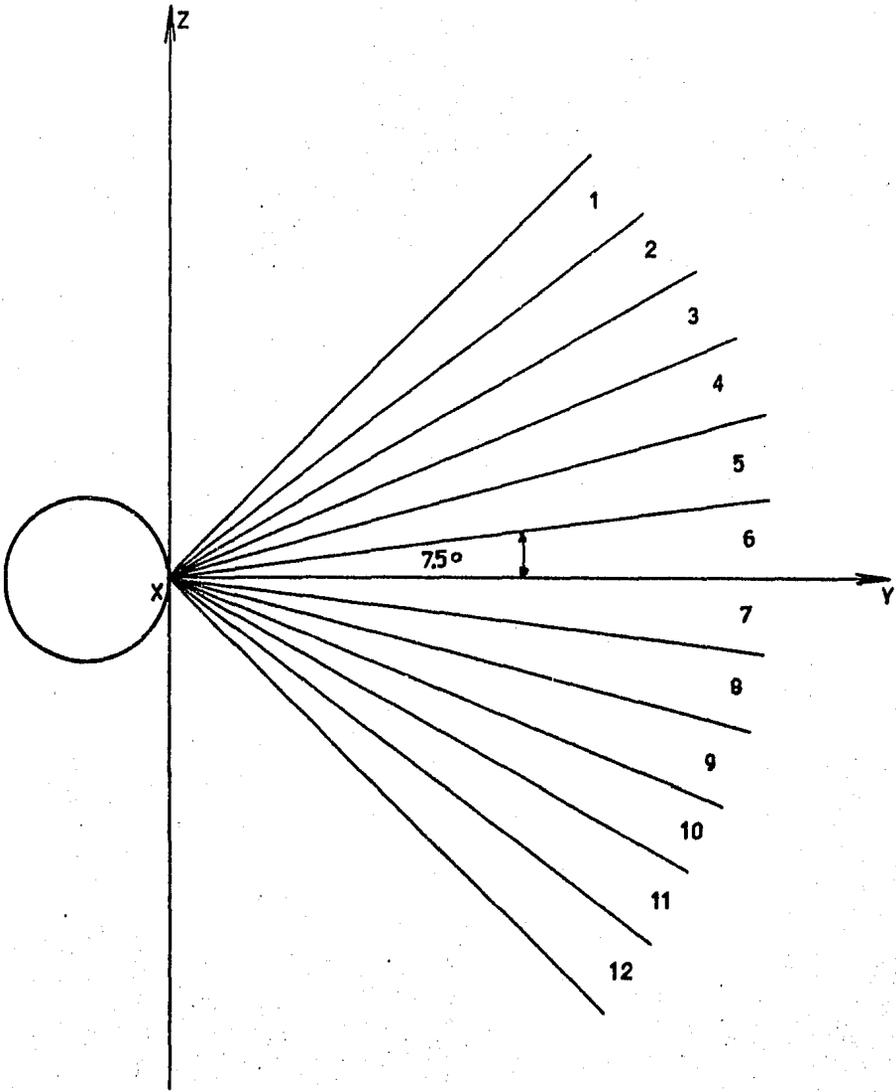


Figura 3-3: Esquema de la división en regiones del espacio tridimensional

## Modelo Teorico Propuesto

implementación binocular se aplica sucesivamente sobre las regiones de la uno a la doce. En la figura 3-4 se observa con más detalle una de las regiones mencionadas.

La figura anterior representa a la región número seis. En ella aparecen activados los fotoreceptores 1 del ojo izquierdo y 6 del derecho del anillo correspondiente a la región. Se supone que existen 12 anillo como los de la figura en cada ojo, uno para cada región.

Los resultados para cada región se almacenan en un arreglo. Al finalizar el barrido de regiones se suman los resultados del ojo izquierdo con los del derecho región por región para obtener así el resultado final.

Como un ejemplo, las 2 figuras siguientes (3-5) muestran el conjunto de puntos en el espacio real que se consideraron para el experimento. Las dos están a la misma escala, y la referencia es la distancia entre los ojos. Cada división corresponde a la cuarta parte entre los ojos.

A continuación se mostrarán fotografías de las gráficas obtenidas en la computadora de un experimento que tuvo como entradas los puntos reales representados en las figuras anteriores. Las fotografías representan el espacio tridimensional tal como lo recupera el sistema. En las primeras cuatro (figs. 3-6 y 3-7) aparece el cuerpo número 1 visto desde diferentes ángulos. Este cuerpo aparece inicialmente ya que corresponde a una de las primeras secciones del espacio que revisa el algoritmo. En las siguientes dos (fig. 3-8) aparece ahora además del cuerpo 1, el cuerpo 2. En el siguiente grupo de dos fotografías (fig. 3-9) se integra además de los cuerpos 1 y 2, el cuerpo 3.

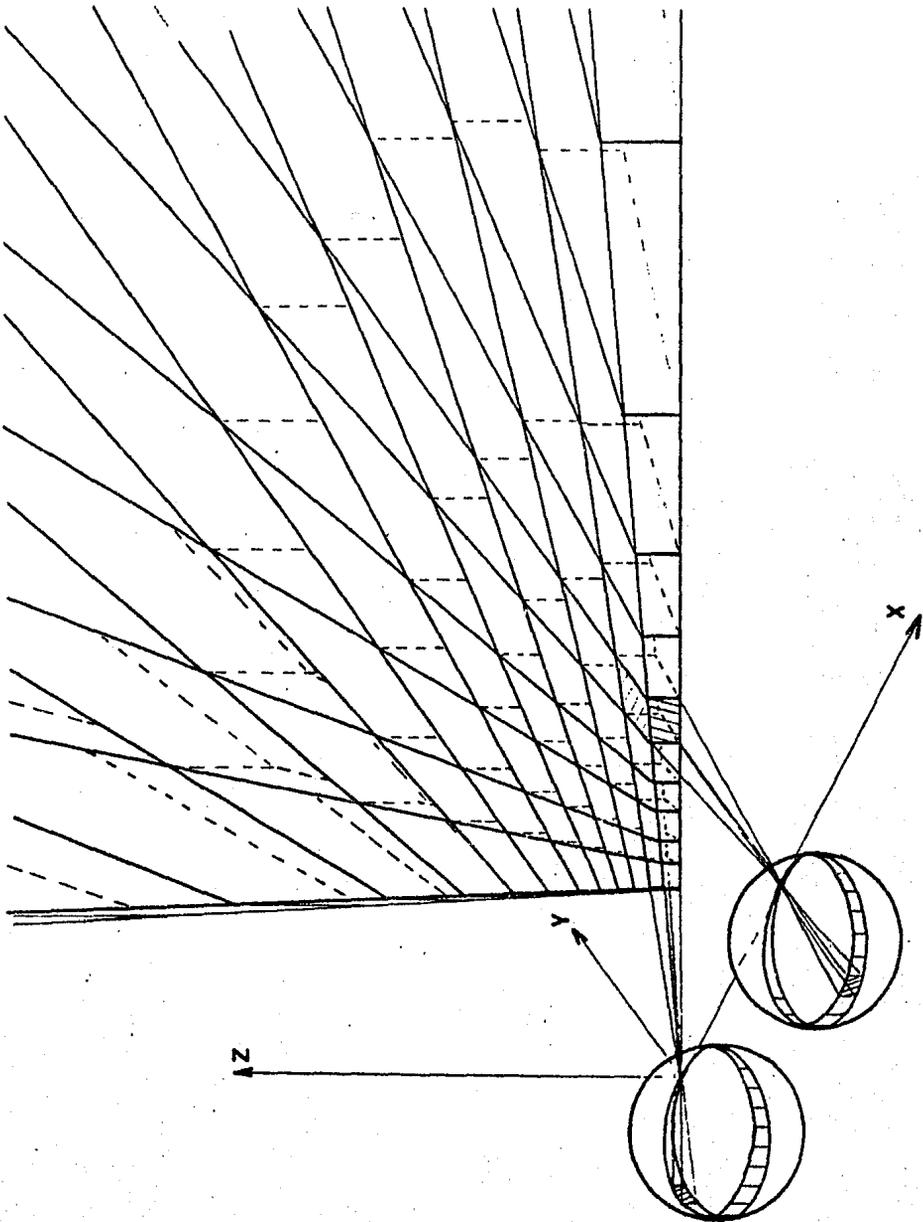


Figura 3-4: Región numero 6 en el espacio tridimensional.  
Fotoreceptores 1 del ojo izquierdo y 6 del derecho  
activos

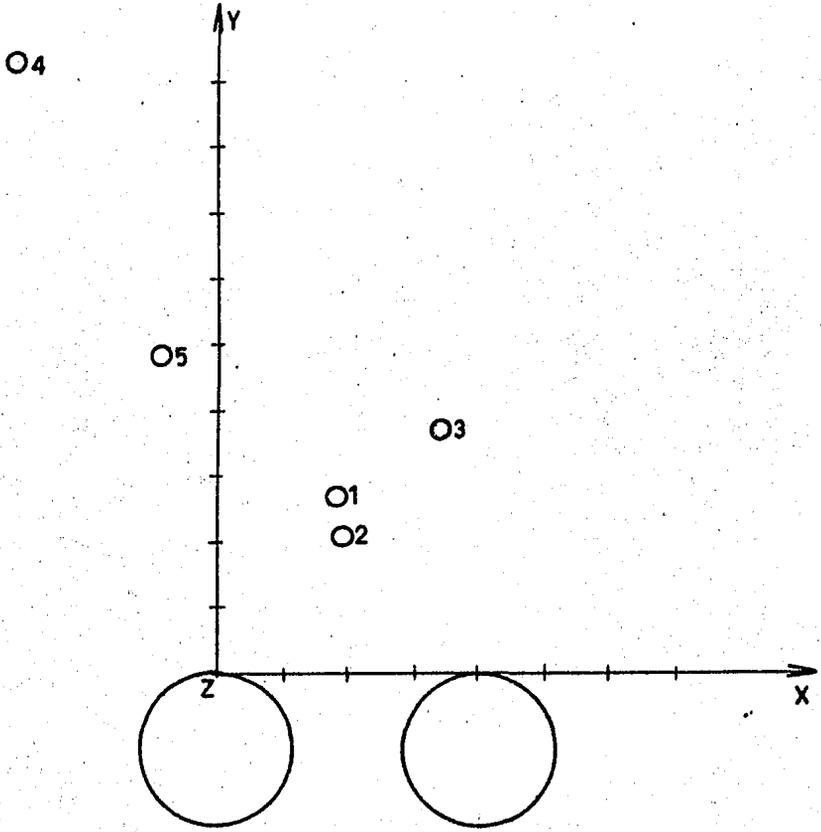
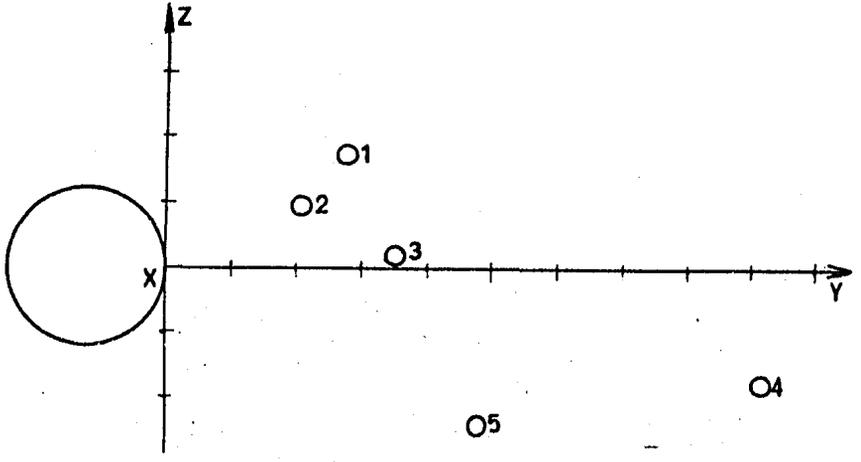


Figura 3-5: Puntos en el espacio real

Puede observarse cómo el algoritmo va recorriendo las diversas secciones del espacio visual. En el último grupo de fotografías (figs. 3-10 y 3-11 se observa un conjunto de 5 cuerpos en el espacio.

### **3.5 Simulación en Computadora Digital**

#### **3.5.1 Formas de Despliegue de Resultados**

La simulación de un modelo por computadora digital presenta la ventaja de permitir diversos tipos de despliegue de resultados. Entre éstos se encuentran los despliegues numéricos, las gráficas y en ocasiones es posible contar con salidas de señales que pueden obtenerse a través de las líneas de un puerto de comunicación.

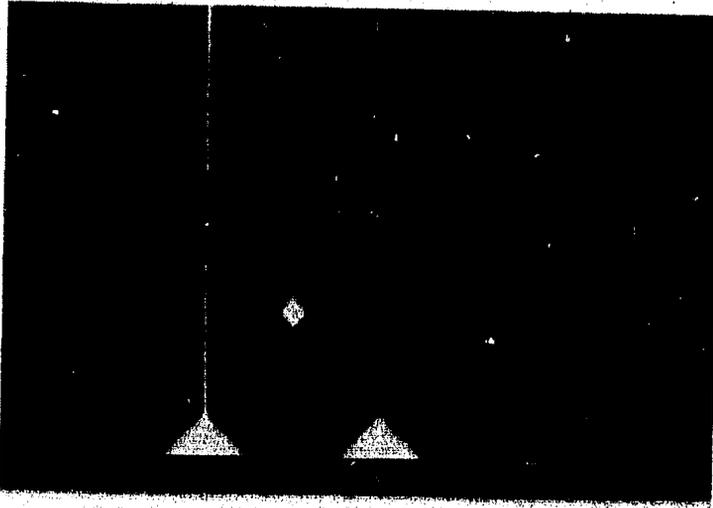
De los mecanismos anteriores utilizamos los primeros, siendo el último la posibilidad más viable cuando se implante el método en un dispositivo físico (robot con visión)

Utilizando los despliegues numérico y gráfico, los resultados de la simulación pueden ser visualizados en las siguientes formas:

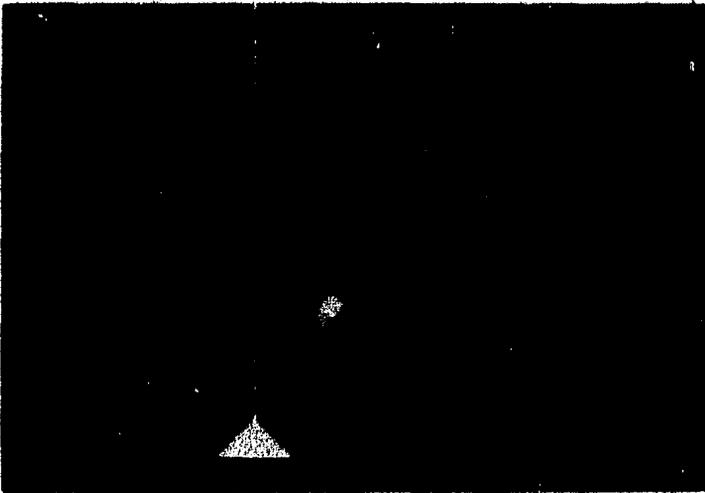
- Malla numérica
- Oróptero bidimensional gráfico
- Malla gráfica
- Oróptero tridimensional gráfico

La primera forma de despliegue, malla numérica, permite seguir la secuencia de simulación con despliegues numéricos que relacionan a las posiciones retinales.

La malla representa a un sólo plano de visión del anfibio y sólo considera las intersecciones posibles de las fibras de ambos

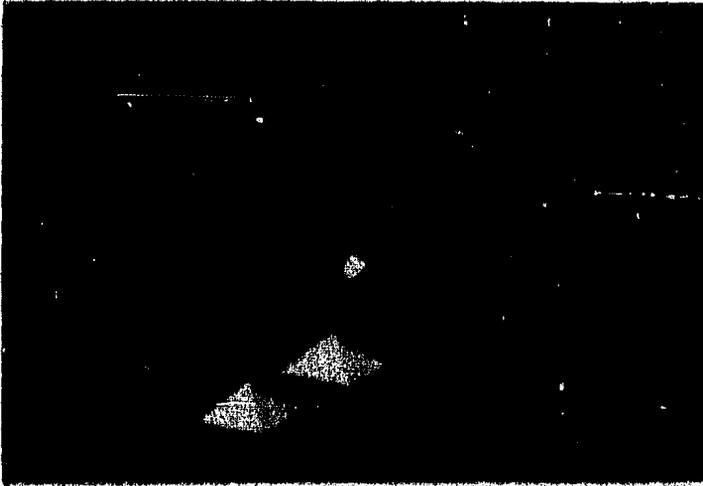


(a)



(b)

Figura 3-6: Aspecto de un sólo cuerpo desde varios puntos de referencia

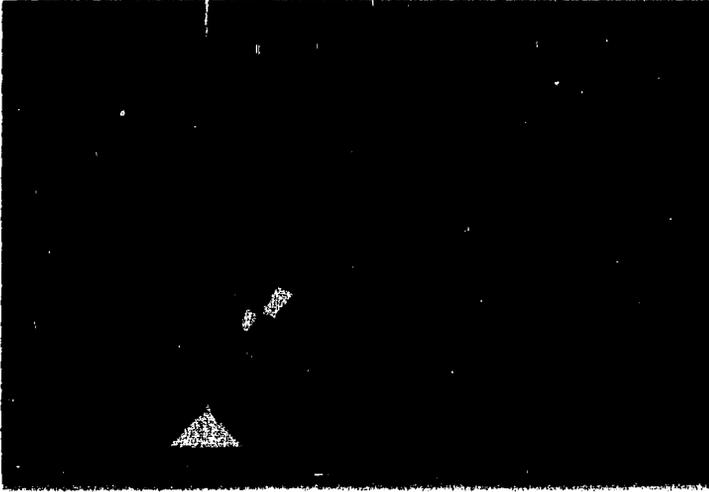


(a)

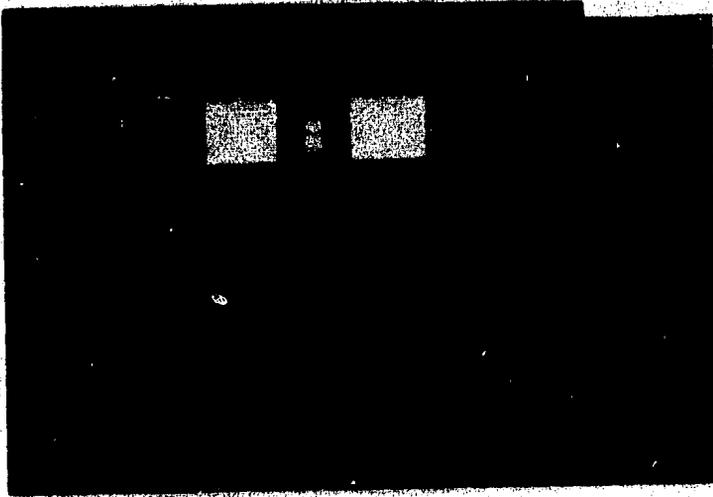


(b)

Figura 3-7: Aspecto de un sólo cuerpo desde varios puntos de referencia (...continuación)

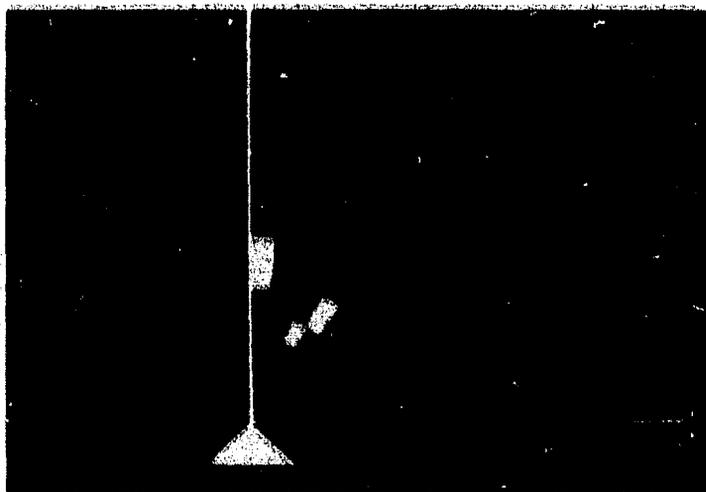


(a)

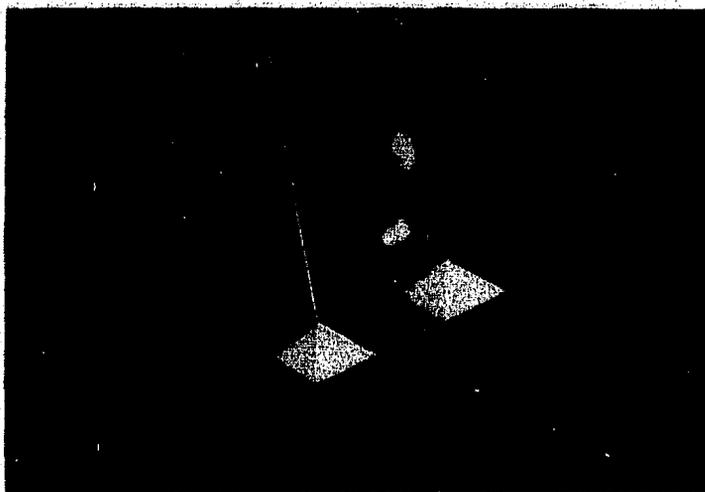


(b)

**Figura 3-8: Aspecto de los dos cuerpos desde  
varios puntos de referencia**

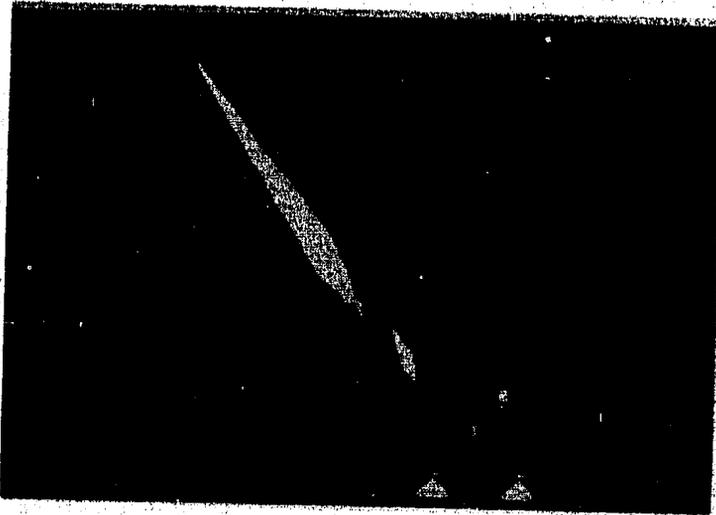


(a)

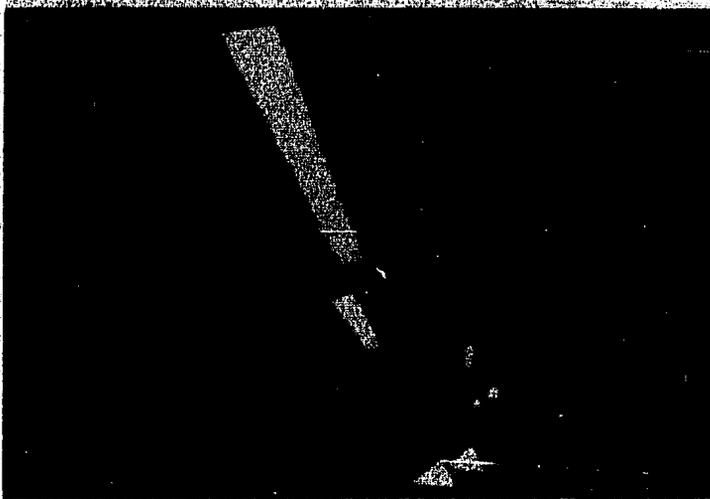


(b)

**Figura 3-9: Aspecto de tres cuerpos desde varios puntos de referencia**

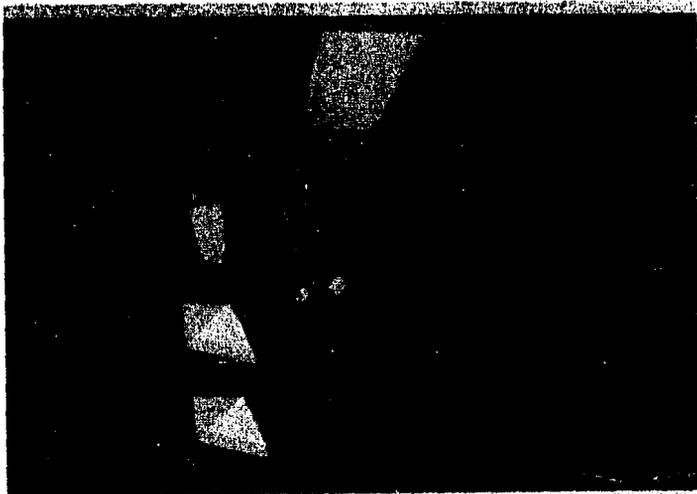


(a)



(b)

**Figura 3-10: Aspecto de cinco cuerpos desde  
varios puntos de referencia**



(a)



(b)

Figura 3-11: Aspecto de cinco cuerpos desde varios puntos de referencia (...continuación)

ojos.

La organización de una malla típica sería la siguiente:

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	*	*	*	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*	*	*	*	*	*
3	*	*	*	*	*	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*	*	*
5	*	*	*	*	*	*	*	*	*	*	*	*
6	*	*	*	*	*	*	*	*	*	*	*	*
7	*	*	*	*	*	*	*	*	*	*	*	*
8	*	*	*	*	*	*	*	*	*	*	*	*
9	*	*	*	*	*	*	*	*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*	*	*
11	*	*	*	*	*	*	*	*	*	*	*	*
12	*	*	*	*	*	*	*	*	*	*	*	*

Se observa que se trata de una matriz cuadrada de NUMNEUR x NUMNEUR (NUMNEUR es igual al número de fotoreceptores en un corte diametral de la retina) y además en los lugares donde no hay asteriscos los elementos son cero. Esta última condición existe debido a que no se consideran las interacciones no existentes entre fibras. Las entradas de la matriz  $MDER[i,j]$  representan el peso que tiene en ese momento la intersección de la fibra  $i$  del ojo izquierdo con la  $j$  del derecho.

Si bien ésta forma permite visualizar el proceso, es posible permitir un despliegue más sencillo de entender. El oróptero bidimensional gráfico (representación del espacio visual donde se distinguen las regiones de visión monocular, binocular, y la región ciega) es una buena opción. Al ser bidimensional, el observador del proceso nuevamente está restringido a percibir la simulación de un sólo plano de visión del animal, aunque éste tiene la ventaja de la simplicidad en comparación con lo complejo de la representación gráfica del modelo tridimensional.

El despliegue gráfico se realiza en una computadora Chromatics

de alta resolución. Se utilizan 15 diferentes tonalidades (negro a blanco) para indicar el peso de cada intersección de fibras, dejando uno para propósitos especiales. De ésta manera estos colores deberán codificar la información que se presenta en la malla numérica.

Un oróptero está representado en la siguiente figura 3-12:

Siempre existe un rótulo en la parte superior del oróptero indicando en qué etapa de simulación se encuentra el proceso.

El sistema en en su estado actual muestra siempre dos orópteros simultáneamente, uno donde se observa el espacio externo, y otro donde se tiene la llamada representación interna.

Cuando la simulación termina, el oróptero de representación interna mostrará en alguna tonalidad la regiones de intersección donde se localizan los estímulos, ésto es, no se mostrará un resultado puntual, sino un resultado regional, ésto se debe a que un estímulo excita toda un área especificada por intersecciones.

Es claro que al codificar un intervalo muy grande de pesos con solo 15 colores, se pierda resolución, y no se podrá indicar con precisión qué región tiene mayor o menor peso si las intensidades resultantes de la codificación resultan iguales para diferentes magnitudes numéricas. Este punto se vuelve importante cuando surgen estímulos que están ocultos parcialmente por otros y sólo son visibles por uno de los ojos.

La malla gráfica logra una combinación de sensación espacial, lo que permite una interpretación más fácil de las magnitudes lo que hace que se noten hasta las diferencias más sutiles en los pesos de las intersecciones.

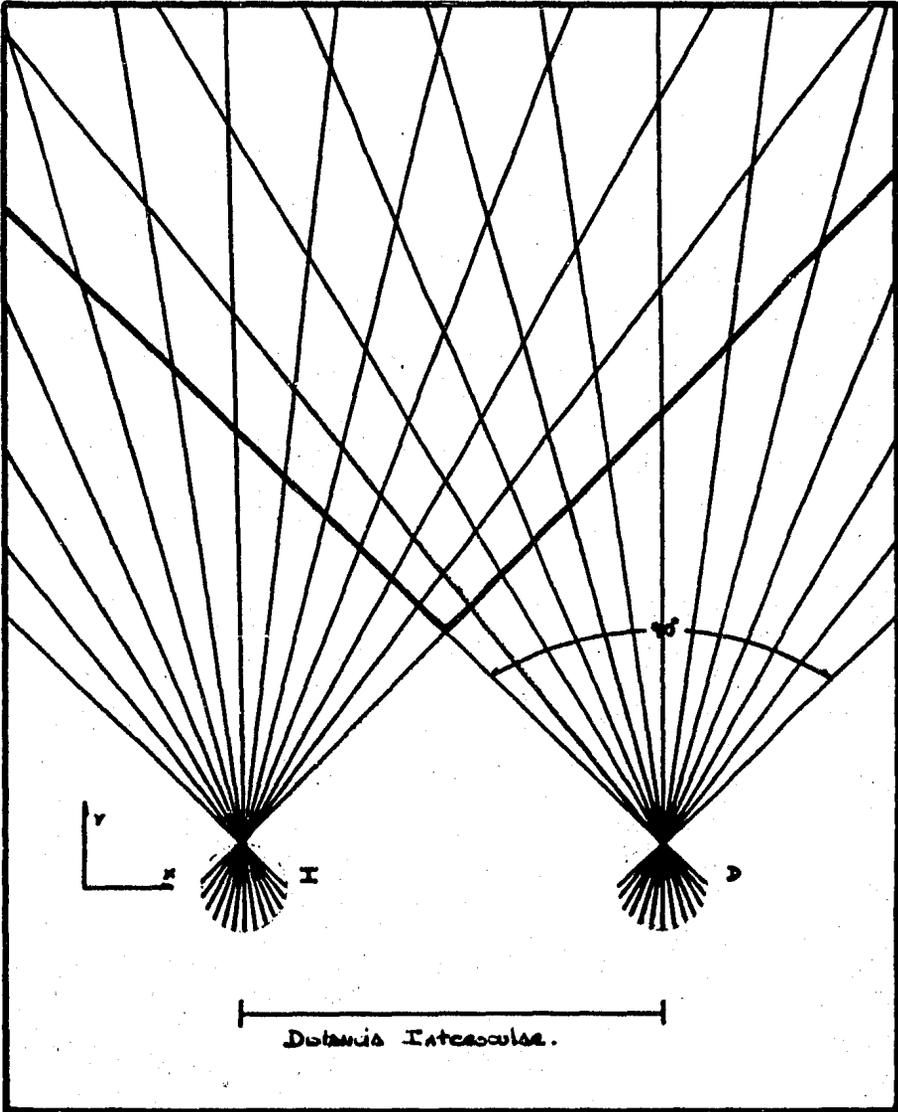


Figura 3-12: Representación usando un oróptero

## Modelo Teorico Propuesto

Nuevamente se tiene una malla, con un eje para las fibras de un ojo en un corte diametral de retina, y otro eje para las fibras del otro ojo. Las entradas son los pesos de las intesecciones, pero en este caso son mostradas como alturas proporcionales.

El estado actual del sistema ha establecido una posición fija de la malla en el video, siendo ésto resultado de la experiencia en cuanto a los resultados, pero existe la posibilidad de mover la posición de ésta para lograr diferentes puntos de visión. Si bien este mecanismo de despliegue reúne buenas características, presenta la desventaja de ser lento, siendo la causa de ésto el control de visibilidad de caras el número de interpolaciones (ver apéndice 3)

Si deseásemos un despliegue para una simulación del modelo en tres dimensiones, las técnicas anteriores no serían las adecuadas. Por esto para el despliegue de información en espacio tridimensional utilizamos software orientado a despliegue de cuerpos. El modelo tridimensional considera una serie de regiones (casi planas) de visión en cada retina cubriendo un espacio de 90 grados (ver figura 3-4), correspondientes al espacio binocular del animal.

El paquete DESPL75 toma las coordenadas espaciales que se generan en la simulación, aplicando transformaciones a éstas coordenadas (ver apéndice 3) y dá como resultado el espacio tridimensional.

De ésta forma todas las salidas necesarias, tanto para el modelo bidimensional como tridimensional son cubiertas.

### 3.6 Experimentos Realizados

El modelo implementado permite realizar diversos tipos de experimentos que corresponden a variaciones sobre una situación normal. Una situación normal es aquella en la que se utilizan los dos ojos y sin la presencia de lentes que desvien la luz. El experimento para una situación tal comienza con la definición de las coordenadas de un conjunto de puntos localizados en el espacio visual; a continuación empiezan a actuar sobre los datos las etapas anteriormente mencionadas, teniéndose la posibilidad de obtención de resultados en cada etapa, o bien hasta el final de la simulación. Los resultados están disponibles en las formas de despliegue anteriormente definidas.

Una primera variación sobre la situación normal puede ser la inclusión de lentes de desviación de luz. Collet [2] ha mostrado en anfibios que al colocárseles lentes que aumentan o disminuyen la disparidad en la retina se provoca que el animal ataque a distancias más próximas o más lejanas de las que realmente se encuentra el objeto, probando con ello que la estereópsis binocular es el mecanismo más importante para codificar profundidad. Por lo anterior se consideró conveniente incluir en el programa la posibilidad de simular la presencia de lentes que aumentan o disminuyen la disparidad. Cabe hacer notar que la acomodación del cristalino no se afecta por la acción de los lentes.

El experimento en este caso es semejante al normal, con la diferencia de que ahora se tiene la posibilidad de elegir entre dos tipos de lentes, de alejamiento y de acercamiento aparentes; además se debe definir un ángulo de desviación de la luz.

Otra variación del experimento es considerar que sólo se tiene

un ojo. Ingle [5] realizó experimentos en anuros imposibilitados de ver con un ojo, los cuales mostraron que la determinación de la profundidad se lograba en forma aproximada, gracias únicamente a mecanismos monoculares (acomodación).

Para éste caso, nuestra implementación permite simular la carencia de un ojo, manteniendo las condiciones de los experimentos antes mencionados (con presencia o ausencia de lentes).

A continuación se presentan los resultados de una serie de experimentos que caracterizan las modalidades que se pueden presentar o generar con el algoritmo propuesto de percepción de profundidad por estereópsis binocular.

Mediante éste es posible realizar las siguientes simulaciones:

- Percepción por medios binoculares con ayuda de medios monoculares.
- Percepción por medios binoculares exclusivamente.
- Percepción por medios monoculares exclusivamente.
- Simular la existencia de lentes que desvien la luz, existiendo la modalidad de alejamiento y acercamiento.

Las condiciones utilizadas fueron las siguientes :

Distancia interocular igual a cuatro unidades.

El valor de las constantes del proceso fueron mantenidas a lo largo de los experimentos :

Cte acomodación = +0.1

Cte otras neuronas = +1.0

Modelo Teorico Propuesto

Cte vecinas = +0.4

Estímulos presentes en las siguientes posiciones

x	y
-2	7
0	+5
+8	+8
+5	+5
+1	+9

Estos estímulos cubren un rango amplio del espacio binocular del animal.

EXPERIMENTO I  
Condiciones

No se utilizan lentes. Experimento binocular

Matriz hacia la cual debe converger el algoritmo

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----parejas activas "vtrld \* vtrli"-----

Esta etapa muestra la activación de las neuronas debido a los estímulos presentados .Se observa la existencia de puntos fantasma.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	4.0	0.0	0.0	4.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

## Modelo Teorico Propuesto

### -----otras neuronas-----

Se puede observar en esta etapa la generación de un contraste más marcado entre las regiones que definen la activación de un receptor y las regiones que no están activas (representadas por valores negativos)

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	2.0	-2.0	-2.0	2.0	-2.0	-2.0	-2.0	2.0	-2.0	2.0	-2.0	-2.0
2	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
3	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
4	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
5	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
6	2.0	-2.0	-2.0	2.0	-2.0	-2.0	-2.0	2.0	-2.0	2.0	-2.0	-2.0
7	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
8	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
9	2.0	-2.0	-2.0	2.0	-2.0	-2.0	-2.0	2.0	-2.0	2.0	-2.0	-2.0
10	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
11	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
12	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0

### -----acción de vecina-----

Observese la variación que existe entre las regiones activas y las no activas. Este paso permite realizar una limitación de la región más probable de existencia de estímulos mediante un mecanismo de cooperatividad entre neuronas vecinas que coodifican una misma profundidad.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	-2.4	-2.4	0.0	-2.4	-2.4	-2.4	0.0	-2.4	0.0	-2.4	-2.4
2	-5.0	-3.4	-3.4	-5.0	-3.4	-3.4	-3.4	-5.0	-3.4	-5.0	-3.4	-5.0
3	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
4	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
5	-5.0	-3.4	-3.4	-5.0	-3.4	-3.4	-3.4	-5.0	-3.4	-5.0	-3.4	-5.0
6	-2.0	-2.8	-2.8	-2.0	-2.8	-2.8	-2.8	-2.0	-2.8	-2.0	-2.8	-2.0
7	-5.0	-3.4	-3.4	-5.0	-3.4	-3.4	-3.4	-5.0	-3.4	-5.0	-3.4	-5.0
8	-5.0	-3.4	-3.4	-5.0	-3.4	-3.4	-3.4	-5.0	-3.4	-5.0	-3.4	-5.0
9	-2.0	-2.8	-2.8	-2.0	-2.8	-2.8	-2.8	-2.0	-2.8	-2.0	-2.8	-2.0
10	-5.0	-3.4	-3.4	-5.0	-3.4	-3.4	-3.4	-5.0	-3.4	-5.0	-3.4	-5.0
11	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0
12	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0	-3.0

### -----acción del factor de acomodación-----

### Modelo Teorico Propuesto

Se introduce la acción monocular con objeto de ayudar a la convergencia del algoritmo. Físicamente esta etapa correspondería a la acomodación del cristalino en los animales. El algoritmo realiza una comparación entre los valores de cada posición y la información proporcionada por el mecanismo de acomodación. Mientras esta información presente mayor diferencia, mayor será la acción inhibitoria presentada.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-0.4	-2.8	-2.8	-0.3	-2.7	-2.6	-2.5	-0.1	-2.6	-0.6	-4.2	-174.3
2	-7.3	-5.7	-5.7	-7.4	-5.9	-6.0	-6.1	-7.9	-4.5	-9.6	-196.2	
3	-6.3	-6.3	-6.4	-6.5	-6.6	-6.7	-7.0	-7.4	-8.8	-211.7		
4	-7.3	-7.4	-7.5	-7.6	-7.7	-8.0	-8.5	-9.9	-224.2			
5	-10.4	-8.9	-9.0	-10.7	-9.4	-9.9	-11.4	-234.9				
6	-2.1	-2.9	-3.1	-2.5	-3.9	-5.4	-231.1					
7	-12.5	-7.2	-11.4	-13.5	-13.4	-239.1						
8	-13.7	-12.3	-7.7	-15.9	-236.2							
9	-2.2	-3.3	-4.7	-218.1								
10	-16.3	-9.0	-218.9									
11	-16.4	-204.5										
12	-187.2											

La etapa siguiente elimina los elementos que no logran pasar un umbral y entran en consideración solamente los sobrevivientes. Esto se puede considerar como una etapa de filtrado agudo.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-0.4	-2.8	-2.8	-0.3	-2.7	-2.6	-2.5	-0.1	-2.6	-0.6	-4.2	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	-2.1	-2.9	-3.1	-2.5	-3.9	-5.4	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	-2.2	-3.3	-4.7	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

Lo siguiente es realizar un proceso iterativo que establece un umbral para la eliminación de elementos con poco peso localizados hasta este momento. Podrán observarse los cambios de peso que sufren los elementos sobrevivientes. Los elementos que no pasen el

Modelo Teorico Propuesto

umbral son eliminados y los que si lo pasen son ayudados para que sigan compitiendo entre si.

\*\*\* comienzan las iteraciones para el ojo izq. \*\*\*

iteración #		1											
i\d	1	2	3	4	5	6	7	8	9	10	11	12	
1	3.0	0.0	0.0	3.3	0.0	0.0	0.0	3.8	0.0	2.7	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	2.5	0.8	0.5	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	2.4	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

iteración #		2											
i\d	1	2	3	4	5	6	7	8	9	10	11	12	
1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	1.2	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	2.4	0.0	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

iteración #		3											
i\d	1	2	3	4	5	6	7	8	9	10	11	12	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.1	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
6	1.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

### Modelo Teorico Propuesto

La iteración anterior establece el punto final de proceso para un ojo. Obsérvese la existencia de un sólo elemento en los renglones que representan las areas activas desde el principio del proceso.

Se inicia en este momento el proceso para el otro ojo, siendo las interpretaciones similares a las realizadas en el caso anterior. Debe indicarse que el proceso en forma real deberia realizarse en paralelo para ambos ojos, lo cual es una idea mas realista de lo que sucede en el proceso de visión, pero se implanto de esta forma por no contar con mecanismos de procesamiento en paralelo.

Inicia proceso para ojo derecho.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----parejas activas "vtrld \* vtrli"-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	4.0	0.0	0.0	4.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Modelo Teorico Propuesto

-----otras neuronas-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	1.0	-4.0	-4.0	1.0	-4.0	-4.0	-4.0	1.0	-4.0	1.0	-4.0	-4.0
2	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	
3	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0		
4	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0			
5	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0				
6	1.0	-4.0	-4.0	1.0	-4.0	-4.0	-4.0					
7	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0						
8	-3.0	-4.0	-4.0	-3.0	-4.0							
9	1.0	-4.0	-4.0	1.0								
10	-3.0	-4.0	-4.0									
11	-3.0	-4.0										
12	-3.0											

-----acción de vecina-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-1.0	-6.0	-6.0	-3.0	-6.0	-4.0	-6.0	-3.0	-8.0	-3.0	-6.0	-4.0
2	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8	-4.8	-3.8	-4.4	
3	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8	-4.8	-3.8		
4	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8	-4.8			
5	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8				
6	-1.0	-6.0	-6.0	-3.0	-6.0	-4.0	-6.0					
7	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0						
8	-3.4	-4.4	-4.4	-3.8	-4.4							
9	-1.0	-6.0	-6.0	-3.0								
10	-3.4	-4.4	-4.4									
11	-3.4	-4.4										
12	-3.4											

-----acción del factor de acomodación-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-1.2	-8.3	-9.3	-3.7	-11.4	-10.5	-13.5	-3.1	-17.9	-3.7	-19.4	-188.2
2	-3.6	-6.7	-7.7	-4.4	-9.9	-10.6	-12.1	-4.1	-9.7	-5.2	-205.9	
3	-3.4	-6.7	-7.8	-4.3	-10.0	-10.7	-12.4	-4.6	-16.6	-208.0		
4	-3.5	-5.5	-7.9	-4.2	-10.1	-11.0	-12.9	-6.1	-230.8			
5	-3.5	-6.9	-6.0	-4.1	-10.4	-11.5	-14.4	-227.9				
6	-1.0	-8.6	-9.7	-3.1	-12.5	-13.1	-241.7					
7	-3.5	-5.5	-8.4	-4.3	-7.1	-238.8						
8	-3.6	-7.3	-6.0	-5.8	-234.3							
9	-1.4	-9.4	-11.8	-219.2								
10	-4.2	-9.0	-213.1									
11	-5.4	-197.2										
12	-175.5											

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-1.2	-3.6	-3.4	-3.5	-3.5	-1.0	-3.5	-3.6	-1.4	-4.2	-5.4	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	-3.7	-4.4	-4.3	-4.2	-4.1	-3.1	-4.3	-5.8	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	-3.1	-4.1	-4.6	-6.1	0.0							
9	0.0	0.0	0.0	0.0								
10	-3.7	-5.2	0.0									
11	0.0	0.0										
12	0.0											

\*\*\* comienzan las iteraciones para el ojo derecho \*\*\*

iteración # 1

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	3.8	0.0	0.0	1.1	0.0	0.0	0.0	2.7	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0			
6	4.2	0.0	0.0	2.2	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	3.4	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

iteración # 2

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.4	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	0.8	0.0	0.0	2.5	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	0.0	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

iteración # 3

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.8	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.8	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Con los resultados obtenidos del proceso en ambos ojos se realiza una suma y se genera el resultado final. La matriz que se muestra representa el espacio que esta viendo el animal y que concuerda con el presentado como una serie de estímulos.

\*\*\*\*\*la matriz final es\*\*\*\*\*

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	2.7	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

EXPERIMENTO II  
Condiciones

Modelo Teorico Propuesto

No se utilizan lentes. Experimento monocular. Se simulará el atrofiamiento de un ojo y el algoritmo utilizará. Sólomente mecanismos monoculares para la determinación del espacio en el que se desenvuelve.

Matriz hacia la cual debe converger el algoritmo

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----parejas activas "vtrld # vtrli"-----

Obsérvese que no existe parejas activas debido a la falta de un ojo.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----otras neuronas-----

Al no existir regiones activas no entra en funcionamiento las acción de otras neuronas y se mantienen los pesos originales.

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----acción de vecina-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----acción del factor de acomodación-----

Esta etapa sera la unica que intervenga en el proceso ademas de la parte iterativa. Con estos elementos se debera determinar el espacio de estimulación presentado al animal.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-0.4	-0.4	-0.4	-0.3	-0.3	-0.2	-0.1	-0.1	-0.2	-0.6	-1.8	-171.9
2	-2.3	-2.3	-2.3	-2.4	-2.5	-2.6	-2.7	-2.9	-1.1	-4.6	-192.8	
3	-3.3	-3.3	-3.4	-3.5	-3.6	-3.7	-4.0	-4.4	-5.8	-208.7		
4	-4.3	-4.4	-4.5	-4.6	-4.7	-5.0	-5.5	-6.9	-221.2			
5	-5.4	-5.5	-5.6	-5.7	-6.0	-6.5	-8.0	-229.9				
6	-0.1	-0.1	-0.3	-0.5	-1.1	-2.6	-228.3					
7	-7.5	-3.8	-8.0	-8.5	-10.0	-235.7						
8	-8.7	-8.9	-4.3	-10.9	-232.8							
9	-0.2	-0.5	-1.9	-216.1								
10	-11.3	-5.6	-215.5									
11	-13.4	-201.5										
12	-184.2											

Después de filtrar mediante un umbral, ningún elemento quedo

Modelo Teorico Propuesto

presente en el ojo atrofiado ( no podria existir ninguno)

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

\*\*\* comienzan las iteraciones para el ojo izq. \*\*\*

Las iteraciones finalmente mostrarán que el ojo izquierdo ( el atrofiado) no presenta información que ayude a la convergencia del algoritmo hacia la región de estimulación presentada.

iteración #	1	2	3	4	5	6	7	8	9	10	11	12
1\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Inicia proceso en ojo derecho.

Este es el ojo que no esta atrofiado y por tanto deberá ser capaz de determinar la profundidad de la mayoría de los estímulos. Sólo podrá observar aquellos puntos que estén cerca de él y no estén ocultos por un punto anterior.

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----parejas activas "vtrld \* vtrli"-----

Este ojo si presentó un espacio de parejas activas y generó un espacio con puntos fantasma además de los reales.

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	4.0	0.0	0.0	4.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

-----otras neuronas-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	1.0	-4.0	-4.0	1.0	-4.0	-4.0	-4.0	1.0	-4.0	1.0	-4.0	-4.0
2	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
3	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
4	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
5	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
6	1.0	-4.0	-4.0	1.0	-4.0	-4.0	-4.0	1.0	-4.0	1.0	-4.0	-4.0
7	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
8	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
9	1.0	-4.0	-4.0	1.0	-4.0	-4.0	-4.0	1.0	-4.0	1.0	-4.0	-4.0
10	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
11	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0
12	-3.0	-4.0	-4.0	-3.0	-4.0	-4.0	-4.0	-3.0	-4.0	-3.0	-4.0	-4.0

-----acción de vecina-----

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-1.0	-6.0	-6.0	-3.0	-6.0	-4.0	-6.0	-3.0	-8.0	-3.0	-6.0	-4.0
2	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8	-4.8	-3.8	-4.4	
3	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8	-4.8	-3.8		
4	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8	-4.8			
5	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0	-4.4	-3.8				
6	-1.0	-6.0	-6.0	-3.0	-6.0	-4.0	-6.0					
7	-3.4	-4.4	-4.4	-3.8	-4.4	-4.0						
8	-3.4	-4.4	-4.4	-3.8	-4.4							
9	-1.0	-6.0	-6.0	-3.0								
10	-3.4	-4.4	-4.4									
11	-3.4	-4.4										
12	-3.4											

-----acción del factor de acomodación-----

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-1.2	-8.3	-9.3	-3.7	-11.4	-10.5	-13.5	-3.1	-17.9	-3.7	-19.4	-188.2
2	-3.6	-6.7	-7.7	-4.4	-9.9	-10.6	-12.1	-4.1	-9.7	-5.2	-205.9	
3	-3.4	-6.7	-7.8	-4.3	-10.0	-10.7	-12.4	-4.6	-16.6	-208.0		
4	-3.5	-5.5	-7.9	-4.2	-10.1	-11.0	-12.9	-6.1	-230.8			
5	-3.5	-6.9	-6.0	-4.1	-10.4	-11.5	-14.4	-227.9				
6	-1.0	-8.6	-9.7	-3.1	-12.5	-13.1	-241.7					
7	-3.5	-5.5	-8.4	-4.3	-7.1	-238.8						
8	-3.6	-7.3	-6.0	-5.8	-234.3							
9	-1.4	-9.4	-11.8	-219.2								
10	-4.2	-9.0	-213.1									
11	-5.4	-107.2										
12	-175.5											

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	-1.2	-3.6	-3.4	-3.5	-3.5	-1.0	-3.5	-3.6	-1.4	-4.2	-5.4	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	-3.7	-4.4	-4.3	-4.2	-4.1	-3.1	-4.3	-5.8	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	-3.1	-4.1	-4.6	-6.1	0.0							
9	0.0	0.0	0.0	0.0								
10	-3.7	-5.2	0.0									
11	0.0	0.0										
12	0.0											

\*\*\* comienzan las iteraciones para el ojo derecho \*\*\*

iteración # 1

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	3.8	0.0	0.0	1.1	0.0	0.0	0.0	2.7	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	4.2	0.0	0.0	2.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	3.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

iteración # 2

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.4	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.8	0.0	0.0	2.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

iteración # 3

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.8	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Modelo Teorico Propuesto

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.8	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Puede observarse que el algoritmo convergió con la información que presento un solo ojo. Nótese que el estímulo faltante en la matriz de convergencia es un punto que se encuentra en la misma región de acción del estímulo (6.1) el cual está más cercano al ojo y por tanto fué el que sobrevivió.

\*\*\*\*\*la matriz final es\*\*\*\*\*

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.9	0.0	1.5	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.8	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

En las siguientes páginas, se muestran en forma gráfica experimentos realizados en dos dimensiones.

Las representaciones se hacen primero a través de orópteros, y posteriormente a través de mallas gráficas, siendo éstas últimas representaciones de situaciones particulares de visión.

Mediante los orópteros se muestra una secuencia completa de simulación bajo condiciones normales (sin lentes y binocular)

presentándose en orden las siguientes etapas:

Parejas activas , otras neuronas del ojo derecho (fig. 3-13), acción de acomodación, primera iteración para el ojo derecho (fig. 3-14), segunda iteración para el ojo derecho y estado final (fig. 3-15).

En la secuencia anterior es interesante notar que al inicio de la simulación existía un grupo grande de puntos fantasma (puntos no existentes que aparecen en las intersecciones de neuronas definidas por dos cuerpos reales), siendo el sistema capaz de converger al espacio real en un número finito de iteraciones (un promedio de 5 iteraciones).

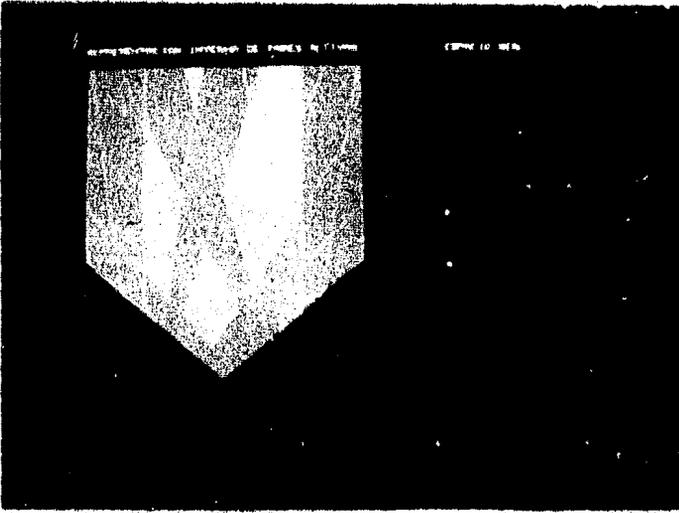
En el estado final puede observarse un estímulo más oscuro que los demás, siendo la causa de esto el que sólo sea visto por el ojo derecho.

Las siguientes dos fotografías corresponden a la representación de un experimento en el que se integró la acción de lentes de alejamiento aparente. Se consideraron prismas que desvían la luz en 8 grados. En ellas se observan la etapa de parejas activas y el resultado final (fig. 3-16).

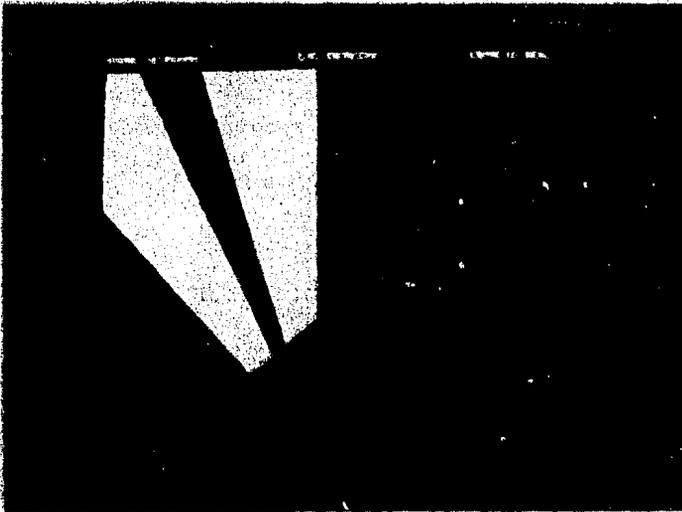
Se puede observar claramente en la última fotografía la percepción de objetos alejados debida a la acción de los lentes, lo cual se puede comparar viendo el espacio real (a la derecha).

El siguiente experimento mostrado representa una simulación que considera la existencia de lentes de acercamiento (también se tomaron 8 grados de desviación de luz). En la secuencia sólo se observan las etapas de parejas activas y estado final (fig. 3-17).

Modelo Teorico Propuesto

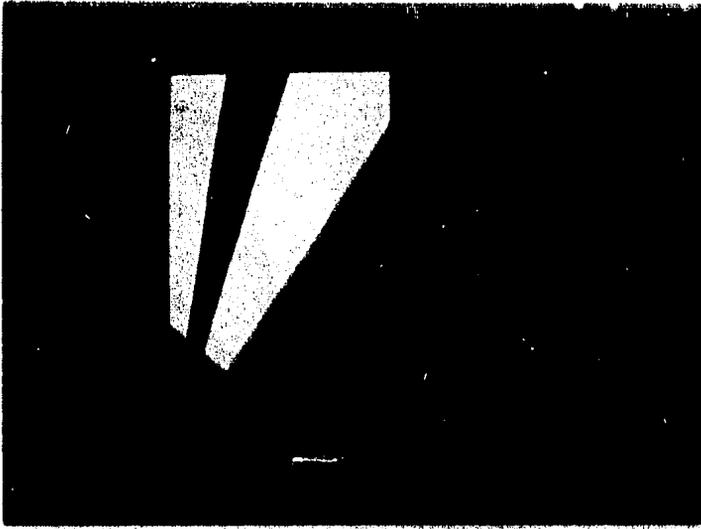


(a)

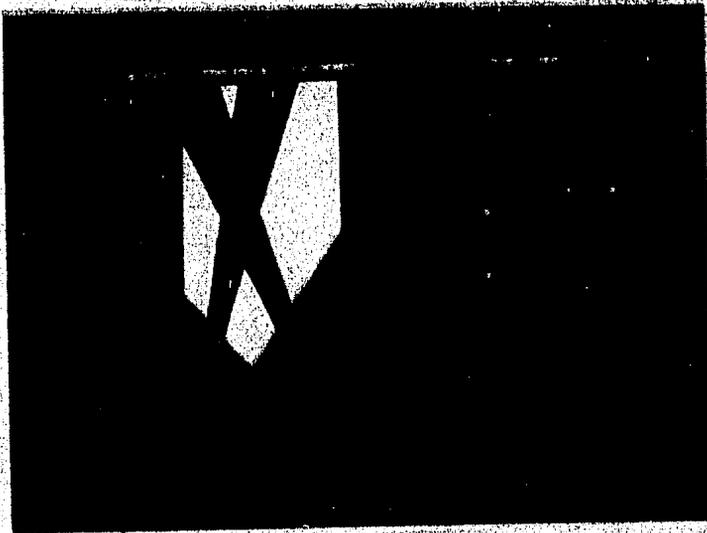


(b)

Figura 3-13: Orópteros para parejas activas (a) y para otras neuronas del ojo derecho (b)

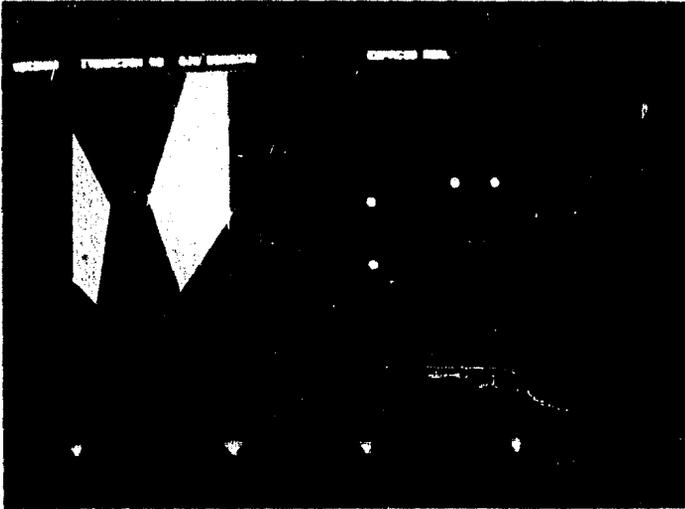


(a)

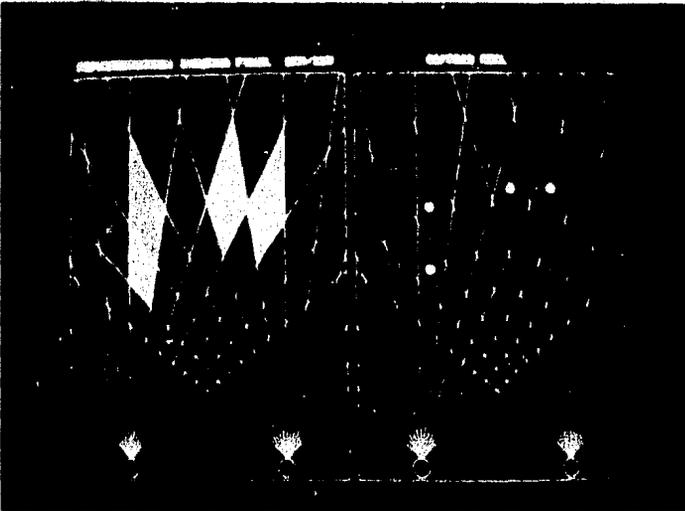


(b)

Figura 3-14: Orópteros para acción de acomodación (a) y para la primera iteración del ojo derecho (b)



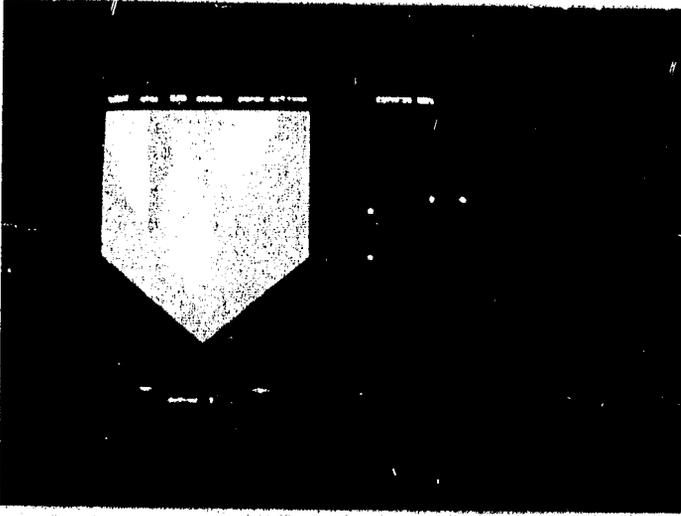
(a)



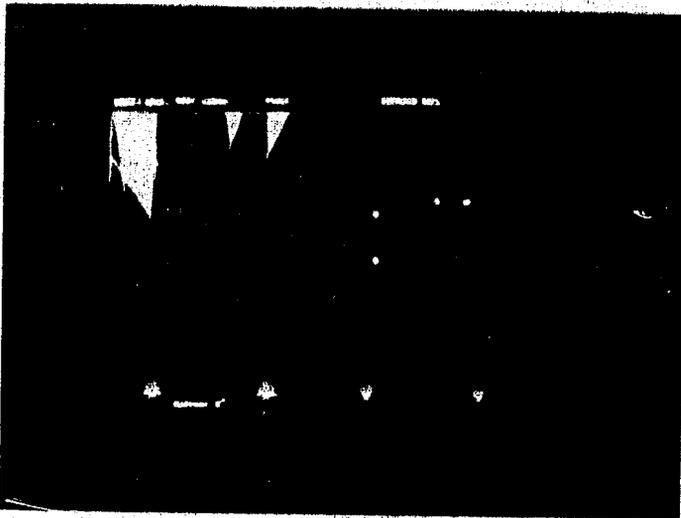
(b)

Figura 3-15: Orópteros para la segunda iteración del ojo derecho (a) y para el estado final (b).

Modelo Teorico Propuesto

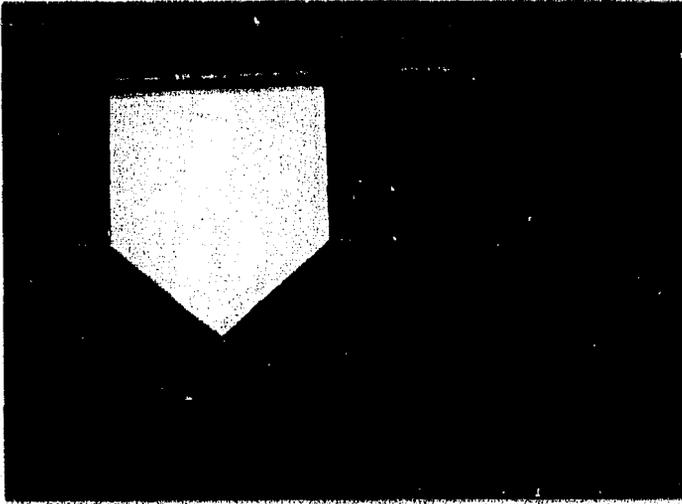


(a)

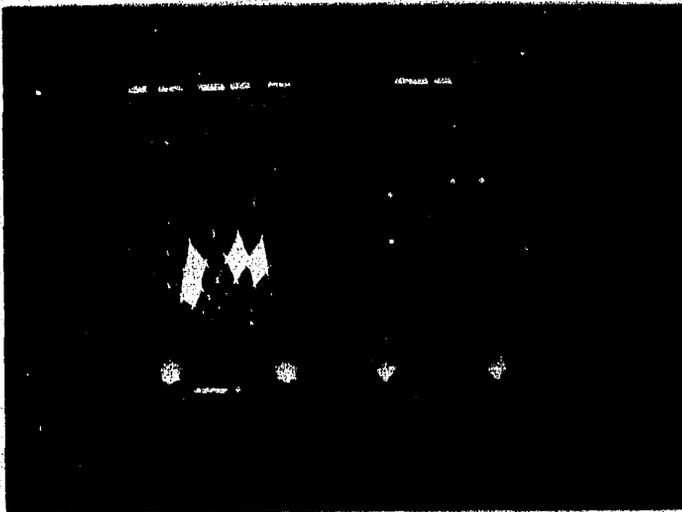


(b)

Figura 3-16: Orópteros para parejas activas (a) y para el estado final (b) usando lentes de alejamiento aparente



(a)



(b)

Figura 3-17: Oróptero para la etapa de parejas activas (a) y para el estado final (b) usando lente de acercamiento aparente de 8 grados

## Modelo Teorico Propuesto

En el estado final se observa que el mecanismo de visión consideró que los estímulos estaban en posiciones más cercanas con respecto a la realidad.

Los siguientes experimentos utilizan como mecanismo de despliegue mallas gráficas (ver en la sección de la simulación en computadora digital la forma de despliegue de resultados) en las cuales las alturas representan el peso del estímulo. En la primera fotografía se observa un experimento binocular bajo condiciones normales. Se contrasta una iteración del ojo derecho (a la izquierda) con el estado final de la simulación (a la derecha) (fig. 3-18 (a)). En la iteración se puede observar que existe una gran cantidad de puntos fantasmas, los cuales desaparecen en el estado final.

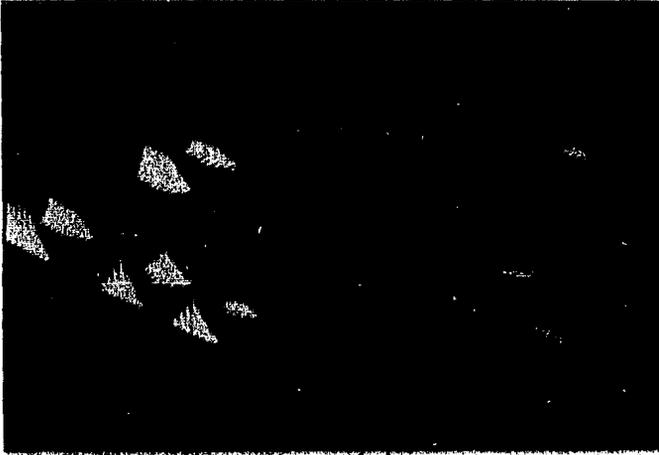
A continuación se muestran 3 fotografías correspondientes a las etapas de una simulación binocular con lentes de acercamiento (figs. 3-18 (b) y 3-19

Compárese la etapa de parejas activas con el resultado final.

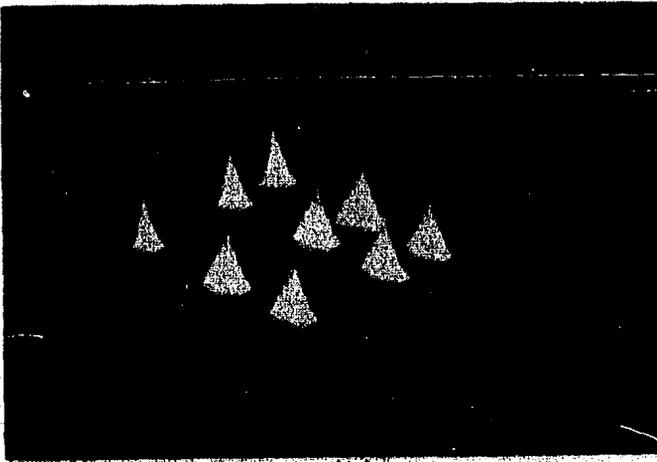
El modelo propuesto considera la posibilidad de prescindir de la información monocular. La siguiente secuencia de fotografías presenta una simulación binocular sin acomodación (figs. 3-20 y 3-21 (a))

Se presenta el estado de parejas activas y una iteración para cada ojo. Se observa que el sistema logra converger aún sin utilizar acomodación. No se muestra el estado final, pero éste corresponde a la suma de los efectos observados en las dos fotografías anteriores.

El grupo de fotografías mostrado a continuación corresponde a



(a)

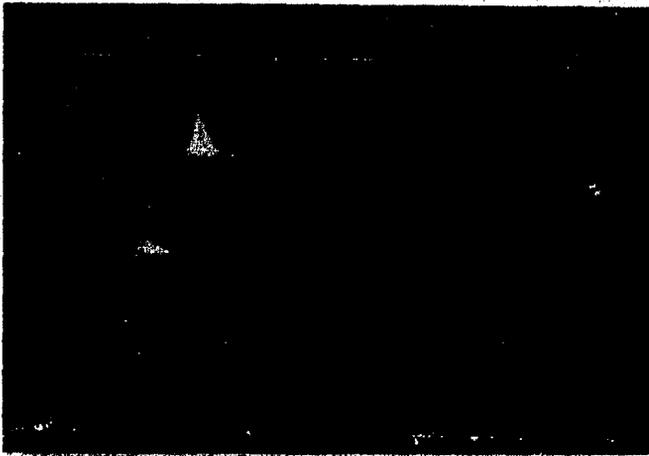


(b)

Figura 3-18: Mallas gráficas que muestran el contraste entre una iteración y el resultado final de un experimento (a) y la etapa de parejas activas con fantasmas (b)

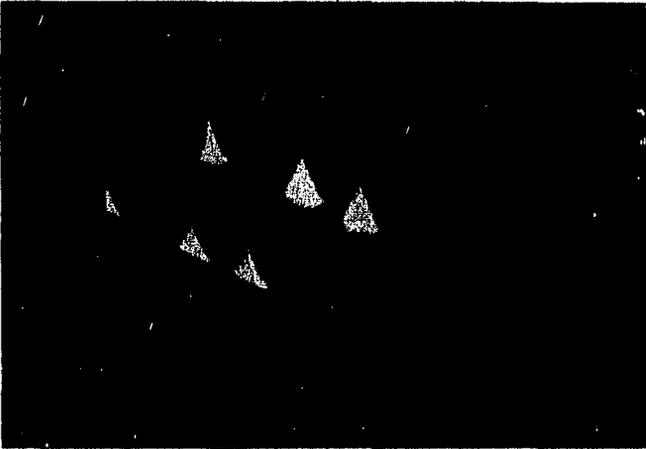


(a)

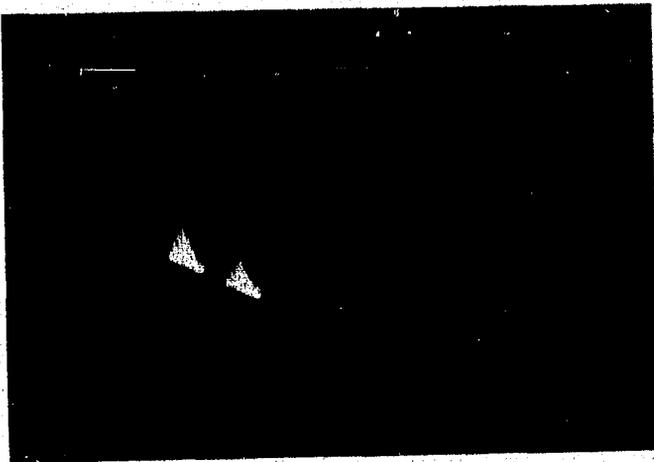


(b)

Figura 3-19: Mallas de una etapa de iteración para el ojo derecho (a) y la etapa final (b)

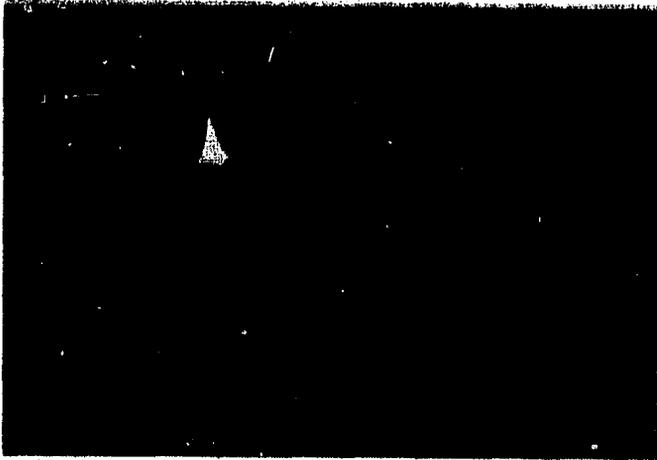


(a)

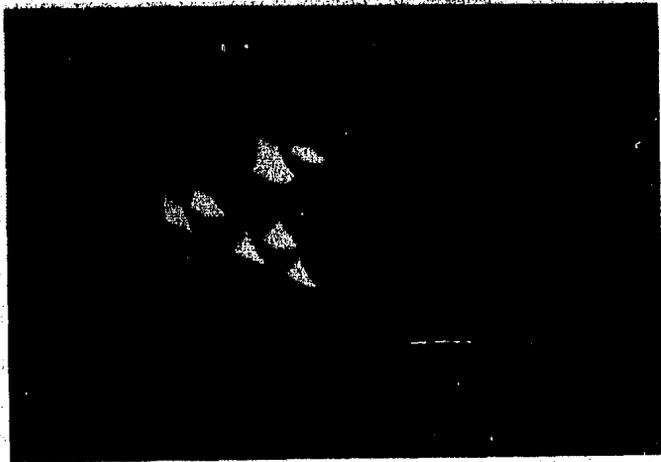


(b)

Figura 3-20: Mallas para parejas activas (a) e iteración 1 del ojo izquierdo (b) sin considerar acomodación



(a)



(b)

Figura 3-21: Mallas para la iteración 1 del ojo derecho sin acomodación (a) y para un experimento monocular sin lentes, (iteración ojo derecho) (b)

experimentos monoculares. El primer experimento es una simulación monocular sin lentes formado por dos fotografías; una iteración sobre el ojo que funciona (derecho) (fig. 3-21 (b)) y el estado final (fig. 3-22 (a)). Si comparamos estos resultados con el espacio real, observamos en el estado final la desaparición de un objeto que está tapado por otro más cercano (desapareció el estímulo más alejado sobre el eje 1).

Las siguientes dos secuencias representan experimentos monoculares utilizando lentes de alejamiento y de acercamiento respectivamente. Ambas están formadas por dos iteraciones y por el estado final. Puede observarse que no se perdió ningún objeto que debiese verse, además se nota la acción de alejamiento y acercamiento aparente (figs. 3-22 (b), 3-23 y 3-24 (a)).

Finalmente mostramos un experimento monocular sin acomodación en la cual el modelo logró captar aún un objeto, pero perdió los otros dos cuya intensidad era relativamente más baja.

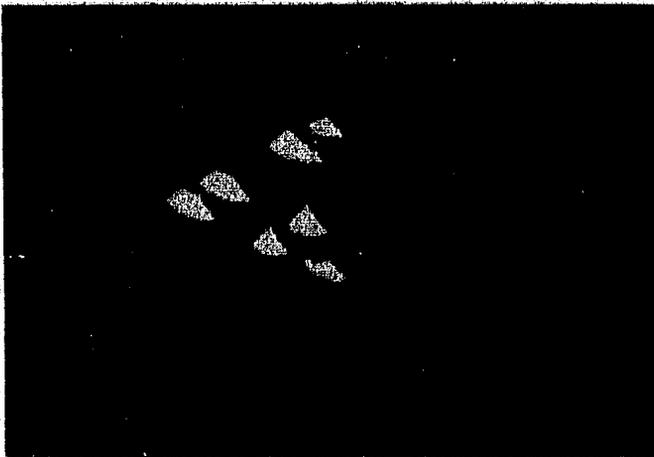
### 3.7 Discusion de Resultados

La simulación del modelo arrojó una serie de resultados para cada una de las situaciones que se considera existen en la naturaleza del fenómeno de visión de los anfibios. Estos resultados mostraron concordancia con los resultados experimentales dados a conocer por Ingle [5] y Collet [2].

Además de la similitud con los resultados experimentales, se observó un proceso de convergencia al estado final bastante rápido, siendo en promedio un máximo de cuatro iteraciones lo necesario para la convergencia. Lo anterior seguramente es útil en la realidad, en donde se requiere identificar rápidamente el ambiente en el cual se mueve un animal, o en su caso, un mecanismo dotado

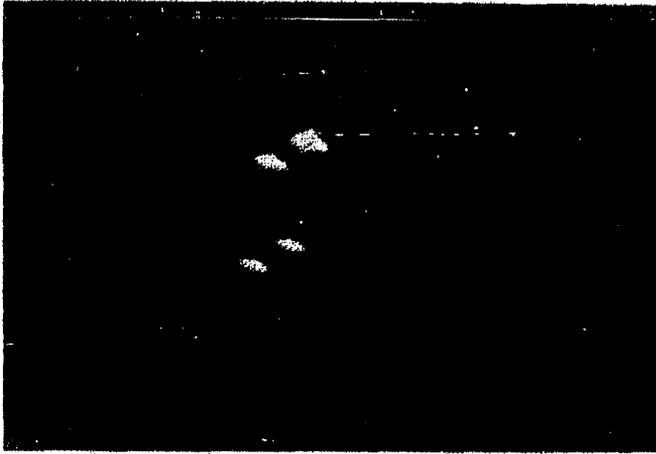


(a)

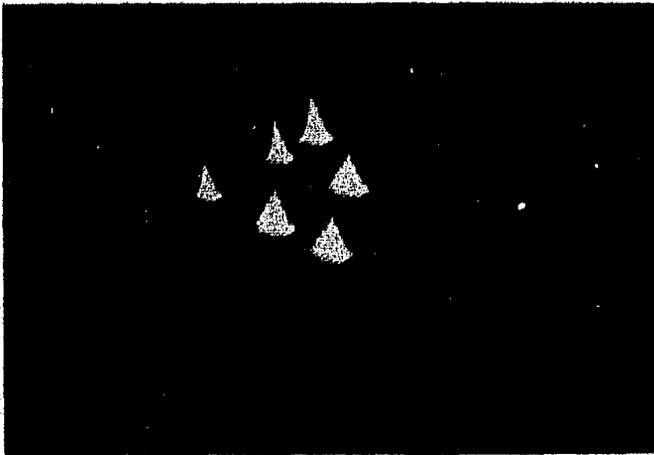


(b)

Figura 3-22: Mallas para el estado final del experimento monocular sin lentes (a) y para la primera iteración del experimento monocular con lentes de alejamiento (b)

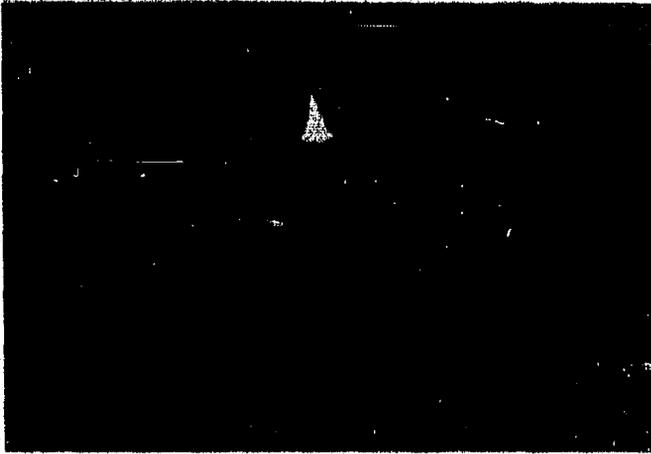


(a)



(b)

**Figura 3-23: Mallas para el estado final de un experimento monocular con lentes de alejamiento (a) y para la iteración 1 de un experimento monocular con lentes de acercamiento (b)**



(a)



(b)

**Figura 3-24: Malla para el estado final de un experimento monocular con lentes de acercamiento (a) y para el estado final de un experimento monocular sin acomodación (b)**

de visión.

Al realizar el análisis de sensibilidad se localizó cierta robustez en el modelo en cuanto a la variación de los parámetros utilizados. Es posible lograr una combinación variada de diversos parámetros que alteran los factores monocular y binocular sin observar variaciones significativas en el resultado.

Entre el conjunto de estímulos que puede recibir un mecanismo con visión pueden existir algunos vistos parcialmente. Los resultados finales permiten identificar éstos estímulos mostrándolos ya sea en intensidades menores o en pesos mínimos, distinguiéndolos de los que excitan ambos ojos.

Lo anterior permite afirmar que se cuenta con un algoritmo completo con las características de ser rápido y sencillo.

#### 4. Analisis de sensibilidad

##### 4.1 Introduccion

Una vez establecido el modelo matemático, es necesario determinar los valores (o rangos de valores) que pueden tomar los parámetros que intervienen en el mismo, de forma que su funcionamiento sea similar (lo más posible) al sistema real.

Este proceso de identificación del modelo se puede llevar a cabo mediante un análisis de sensibilidad de parámetros que muestre los efectos que produzcan las variaciones en cada uno de ellos.

El análisis también evalúa la robustez que pueda tener el modelo y los valores límite de los parámetros para que pueda mantener un comportamiento adecuado.

Para realizar este análisis, debe observarse -a partir de la ecuación que aparece en la segunda sección del capítulo 3- la existencia de tres constantes que actúan en el proceso.

La constante  $k_3$  interviene en la acción de relación de las neuronas vecinas;  $k_1$  en la acción del proceso de acomodación y  $k_2$  en la acción que hemos denominado otras neuronas.

De las tres,  $k_2$  y  $k_3$  son constantes para el proceso binocular y  $k_1$  para el monocular.

Se puede demostrar que el término asociado a  $k_2$  no presenta mucha relevancia en el proceso y básicamente su función se reduce a mostrar más notoriedad entre los elementos activos y los no activos.

#### 4.2 Metodologia

Comprender el funcionamiento del mecanismo variando los valores de las constantes hubiera sido complejo si no se hubiera utilizado una herramienta como la computadora digital. Con los resultados obtenidos se realizaron gráficas que permitieron obtener el volúmen de convergencia para los diversos valores de las constantes. Para planear un conjunto inicial de valores se consideró lo siguiente:

La constante  $k_3$  no podría ser muy grande porque el término

$$| ( w(i, j) - w(i+1, j) ) | + | ( w(i, j) - w(i-1, j) ) |$$

puede arrojar valores extremadamente grandes y con  $k_3$  grande este sería el elemento dominante.

$k_1$  tampoco podría ser grande porque el término

$$\text{disp}(i, j) - \text{ac}(i)$$

es el resultado de la acción monocular, la cual solo es un medio de ayuda para poder converger más rápido.

Lo anterior indica que debemos iniciar con valores pequeños o escalados en las constantes  $k_1$  y  $k_3$ , siendo  $k_2$  una constante que prácticamente podría tomar cualquier valor. Sin embargo la terceta  $( 0, 0, 0 )$  no sería válida porque se eliminarían todas las acciones y la función  $dw(i, j)/dt$  sería practicamente negativa para todas las parejas  $( i, j )$ .

Considerando lo anterior se iniciaron las pruebas con valores pequeños de  $k_1$ ,  $k_2$  y  $k_3$ .

Las condiciones de experimentación fueron las siguientes:

Sin lentes

Distancia interocular igual a 4 unidades

Conjunto de puntos fijo

#### 4.3 Simulacion en Computadora Digital

Corridas tendientes a determinar el espacio de convergencia del algoritmo en función de la variación de las constantes  $k_1$ ,  $k_2$ , y  $k_3$

Se presenta una serie de corridas manteniendo fijas las constantes que alteran el fenómeno de acción de "otras neuronas" y acomodación.

En primer lugar se varía el parámetro de acción de vecinas en un cierto rango de valores significativos. En este proceso se mantienen constantes los demás parámetros generando resultados que permiten realizar gráficas de los niveles de convergencia del algoritmo.

Las condiciones de experimentación son:

Sin utilización de lentes

Distancia interocular igual a cuatro unidades

Experimento binocular

Para el siguiente grupo de estímulos

-2	7
0	+5
+8	+8
+5	+5
+1	+9

Se tuvieron los resultados siguientes. Estas posiciones cubren los

Analisis de Sensibilidad

límites del espacio visual considerado.

Matriz hacia la cual debe converger el algoritmo

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1.0	0.0	0.0	1.0	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0					
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0					
9	1.0	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

Corrida # 1

Factores: acomodación= 0.2 otras neuronas=1 acción de vecinas= -100

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	2.5	3.5	0.0	4.5	6	643.2	1211.1	1062.7	805.4	469.2	0.0
2	2.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0					
7	3.8	0.0	0.0	0.0	0.0	0.0	0.0					
8	9.1	0.0	0.0	0.0	0.0	0.0	0.0					
9	1074.6	0.0	0.0	0.0								
10	388.7	0.0	0.0									
11	0.0	0.0										
12	0.0											

Corrida # 2

Factores Acomodación = 0.2 Otras neuronas = 1 Acción vecinas = -10

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	2.5	3.5	0.0	4.5	0.0	67.2	131.1	102.7	85.4	19.2	0.0
2	2.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0					
7	3.8	0.0	0.0	0.0	0.0	0.0	0.0					
8	9.1	0.0	0.0	0.0	0.0	0.0	0.0					
9	114.6	0.0	0.0	0.0								
10	28.7	0.0	0.0									
11	0.0	0.0										
12	0.0											

Analisis de Sensibilidad

Corrida # 3

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = -1

Matriz de convergencia final

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	2.5	0.0	0.0	0.0	0.0	0.0	23.1	0.0	13.4	0.0	0.0
2	1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	18.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 4

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = -0.5

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	17.1	0.0	9.4	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	13.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 5

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.7	0.0	5.4	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	7.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 6

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.4

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 7

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 1.5

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 8

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 1.1

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 9

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.8

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 10

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.6

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 11

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.5

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 9

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.8

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 10

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.6

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.4	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 11

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.5

Analisis de Sensibilidad

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 12

Factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.45

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.8	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.1	0.0	0.0	2.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	1.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 13

factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.4

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.3	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	6.2	0.0	0.0	5.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 14

Analisis de Sensibilidad

factores Acomodación = 0.2 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

A continuación se muestra un análisis experimental de la influencia de la constante que afecta el proceso de acomodación. Las condiciones experimentales se mantuvieron.

Nota: en algunas matrices se nota un defasamiento de los resultados debido a que el formato de salida del programa no podía ser mayor a 80 líneas.

Matriz hacia la que debe converger el algoritmo

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 1

Factores Acomodacion = -1000 Otras neuronas = 1 Vecinas = 0.3  
 Multiplicar los valores mostrados por 10

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32180.0	0.0	10072.0	13420.4	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30474.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	43062.1	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	29158.3	0.0		
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	60907.6	0.0			
6	18872.0	0.0	0.0	18791.0	0.0	30716.6	0.0					
7	0.0	17496.0	0.0	0.0	18711.0	0.0						
8	0.0	0.0	16030.0	0.0	0.0							
9	14244.0	0.0	0.0	0.0								
10	0.0	96474.2	0.0									
11	13420.9	0.0										
12	0.0											

Corrida # 2

Factores Acomodacion = -100 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	32181.8	0.0	10073.5	13415.6	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30475.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4306.2	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2915.8	0.0		
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6090.8	0.0			
6	188727.2	0.0	0.0	187919.2	0.0	30711.7	0.0					
7	0.0	174961.6	0.0	0.0	187116.6	0.0						
8	0.0	0.0	160305.2	0.0	0.0							
9	142447.2	0.0	0.0	0.0								
10	0.0	96475.9	0.0									
11	13418.1	0.0										
12	0.0											

Corrida # 3

Factores Acomodacion = -45 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16096.4	0.0	5037.9	665.6	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15239.9	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	215.3	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	145.8	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	304.5	0.0				
6	9443.7	0.0	0.0	9398.2	0.0	153.6	0.0					
7	0.0	8749.0	0.0	0.0	9356.5	0.0						
8	0.0	0.0	8016.4	0.0	0.0							
9	7124.4	0.0	0.0	0.0								
10	0.0	4824.9	0.0									
11	668.1	0.0										
12	0.0											

Corrida # 4

Factores Acomodacion = -0.5 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1615.0	0.0	505.2	61.6	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1526.3	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	21.5	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.6	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30.5	0.0			
6	951.3	0.0	0.0	942.0	0.0	15.4	0.0					
7	0.0	875.8	0.0	0.0	936.3	0.0						
8	0.0	0.0	802.7	0.0	0.0							
9	714.4	0.0	0.0	0.0								
10	0.0	483.6	0.0									
11	64.1	0.0										
12	0.0											

Corrida # 5

Factores Acomodacion = -0.1 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	327.8	0.0	102.3	7.9	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	307.3	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.3	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.9	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	7.8	0.0				
6	196.5	0.0	0.0	190.3	0.0	3.1	0.0					
7	0.0	175.9	0.0	0.0	187.8	0.0						
8	0.0	0.0	161.5	0.0	0.0							
9	144.6	0.0	0.0	0.0								
10	0.0	97.7	0.0									
11	10.4	0.0										
12	0.0											

Corrida # 6

Factores Acomodacion = 0 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.4	0.0	1.6	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	2.1	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

Corrida # 7

Factores Acomodacion = 0.001 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.4	0.0	1.6	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 8

Factores Acomodacion = 0.01 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.7	0.0	1.7	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 9

Factores Acomodacion = 0.1 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	0.0	2.3	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	2.7	0.0	0.0	1.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	3.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 10

Factores Acomodacion = 0.05 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.7	0.0	1.9	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.9	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	2.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 11

Factores Acomodacion = 0.3 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.2	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	10.1	0.0	0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 12

Factores Acomodacion = 0.2 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 13

Factores Acomodacion = 0.25 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.5	0.0	3.3	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	8.7	0.0	0.0	5.3	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	4.4	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0											
12	0.0											

Corrida # 14

Factores Acomodacion = 0.29 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	9.9	0.0	0.0	5.9	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	4.6	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

Corrida # 15

Factores Acomodacion = 0.5 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	17.6	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0				
6	14.7	0.0	0.0	8.9	0.0	0.0	0.0					
7	0.0	0.0	0.0	0.0	0.0	0.0						
8	0.0	0.0	0.0	0.0	0.0							
9	5.7	0.0	0.0	0.0								
10	0.0	0.0	0.0									
11	0.0	0.0										
12	0.0											

Corrida # 16

Factores Acomodacion = 1 Otras neuronas = 1 Vecinas = 0.3

Matriz de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	27.8	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	21.4	0.0	0.0	11.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 17

Factores Acomodacion = 10 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	22.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 18

Factores Acomodacion = 50 Otras neuronas = 1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Este es el análisis experimental de la influencia de la constante que interviene en el proceso de acción de "otras neuronas". Las condiciones experimentales se mantuvieron. Se

## Analisis de Sensibilidad

incluye un análisis para casos especiales de valores de parámetros.

Matriz hacia la cual debe converger el algoritmo

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 1

Factores Acomodación = 0.2      Otras neuronas = -1000      Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 2

Factores Acomodación = 0.2      Otras neuronas = -100      Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Analisis de Sensibilidad

Corrida # 3

Factores Acomodación = 0.2 Otras neuronas = -50 Vecinas = 0.3  
Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 4

Factores Acomodación = 0.2 Otras neuronas = -10 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 5

Factores Acomodación = 0.2 Otras neuronas = -1 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	6.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 6

Factores Acomodación = 0.2 Otras neuronas = -2 Vecinas = 0.3

Analisis de Sensibilidad

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.7	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	7.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 7

Factores Acomodación = 0.2 Otras neuronas = -5 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 8

Factores Acomodación = 0.2 Otras neuronas = -3 Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12.9	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.5	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 9

Factores Acomodación = 0.2 Otras neuronas = -0.1 Vecinas = 0.3

Matriz final de convergencia

Analisis de Sensibilidad

i\c	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 10  
 Factores Acomodación = 0.2      Otras neuronas = 0      Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 11  
 Factores Acomodación = 0.2      Otras neuronas = 2      Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 12  
 Factores Acomodación = 0.2      Otras neuronas = 10      Vecinas = 0.3

Matriz final de convergencia

Analisis de Sensibilidad

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 13

Factores Acomodación = 0.2

Otras neuronas = 100

Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Corrida # 14

Factores Acomodación = 0.2

Otras neuronas = 1000

Vecinas = 0.3

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.9	0.0	3.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

A continuación se presentan una serie de corridas que presentan características particulares por los valores de las constantes

Analisis de Sensibilidad

Factores Acomodación = 0.2 Otras neuronas = 0.2 Vecinas = 0.2

Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.5	0.0	3.8	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	5.4	0.0	0.0	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	5.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Factores Acomodación = 10 Otras neuronas = 10 Vecinas = 10  
Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Escalamiento de 10

Factores Acomodación = 2 Otras neuronas = 10 Vecinas = 3  
Matriz final de convergencia

i\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Escalamiento de 0.3

Factores Acomodación = 0.6 Otras neuronas = 3 Vecinas = 0.9  
Matriz final de convergencia

Análisis de Sensibilidad

1\d	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.2	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	3.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

4.4 Discusion de los Resultados

En seguida se muestra en forma tabular, el resumen de los resultados obtenidos durante los experimentos de análisis de sensibilidad. La columna "Acomodación" representa los diversos valores que tomó la constante k1 (factor de acomodación); en la columna "Otras Neuronas" se muestran los valores con que se varió la constante k2 (factor de acción de las otras neuronas); la columna "Vecinas" contiene los valores correspondientes a la variación de k3 (acción de las vecinas). En la última columna se muestra el "Porcentaje de Convergencia", el cual se define como la diferencia entre el número real de objetos en el espacio menos el número de valores vistos por el modelo; esta última cantidad puede ser mayor o menor que la primera, por lo que se tienen valores positivos y negativos del Porcentaje de Convergencia.

TABLA DE RESULTADOS DE LAS SIMULACIONES

Acomodación	Otras neuronas	Vecinas	Convergencia %
0.01	-50	0.3	0
0.01	-10	0.3	0
0.01	-5	0.3	0
0.01	-3	0.3	100
0.01	-0.6	0.3	100
0.01	-0.1	0.3	100
0.01	0.1	0.3	100
0.01	1.0	0.3	100
0.01	10	0.3	100
0.01	1000	0.3	100

Analisis de Sensibilidad

0.01		-5	0.4	0
0.01	mayor de	-3	0.4	100
0.01	menor de	-3	0.5	0
0.01	mayor de	-3	0.5	40
0.05	menor de	-3	0.3	0
0.05	mayor de	-3	0.3	100
0.2	menor de	-3	0.3	0
0.2		-3	0.3	60
0.2		-2	0.	80
0.2		-0.6	03	80
0.2	mayor de	-0.5	0.3	100
0.02		1	menor de -1	-100
0.02		1	-0.9	0
0.02		1	-0.8	60
0.02		1	-0.5	100
0.02		1	de -5 a 0.3	100
0.02		1	de 0.4 a 0.49	80
0.02		1	de 0.49 a 0.8	40
0.02		1	mayor de 0.8	0
0.05		1	menor de -0.5	-100
0.05		1	-0.5	0
0.05		1	-0.4	60
0.05		1	-0.3	60
0.05		1	-0.2	100
0.05		1	de -0.2 a 0.45	100
0.05		1	0.5	40
0.05		1	0.8	40
0.05		1	1.1	0
0.05		1	mayor de 0.9	0
0.01		1	menor de 0	-100
0.01		1	de 0 a 0.45	100
0.01		1	0.5	40
0.01		1	0.8	40
0.01		1	mayor de 0.8	0
mayor de -0.1		1	0.3	-100
0		1	0.3	60
de 0 a 0.001		1	0.3	60
de 0.01 a 0.25		1	0.3	100
0.29		1	0.3	80
de 0.29 a 0.5		1	0.3	80
de 0.51 a 1		1	0.3	60
10		1	0.3	20
mayor de 50		1	0.3	0
menor de -0.1		1	0.4	0
de 0 a 0.001		1	0.4	60
de 0.01 a 0.24		1	0.4	100
de 0.25 a 0.49		1	0.4	80
de 0.50 a 0.99		1	0.4	60
1		1	0.4	40
mayor de 10		1	0.4	0
menor de -0.05		1	0.5	-100
0		1	0.5	0
de 0 a 0.001		1	0.5	0
de 0.001 a 0.01		1	0.5	20
de 0.01 a 5		1	0.5	40
mayor de 10		1	0.5	0

A partir de la tabla anterior se observó que los valores

válidos que pueden tomar los parámetros son:

Para  $k_1$ , entre 0 y 0.25 Para  $k_2$ , a partir de -3 Para  $k_3$ , entre 0.1 y 0.5

Sin embargo conviene utilizar valores alejados de las cotas para obtener resultados más confiables, ya que estas fronteras son estimaciones aproximadas, pero pueden existir variaciones. De cualquier manera es útil tener la libertad de modificar estos parámetros fuera de los rangos establecidos.

En base a los datos anteriores se generaron unas gráficas bidimensionales que muestran la variación de los parámetros (figs. 4-1, 4-2 y 4-3) y una gráfica que muestra el espacio de convergencia tridimensional. Ver figura 4-4.

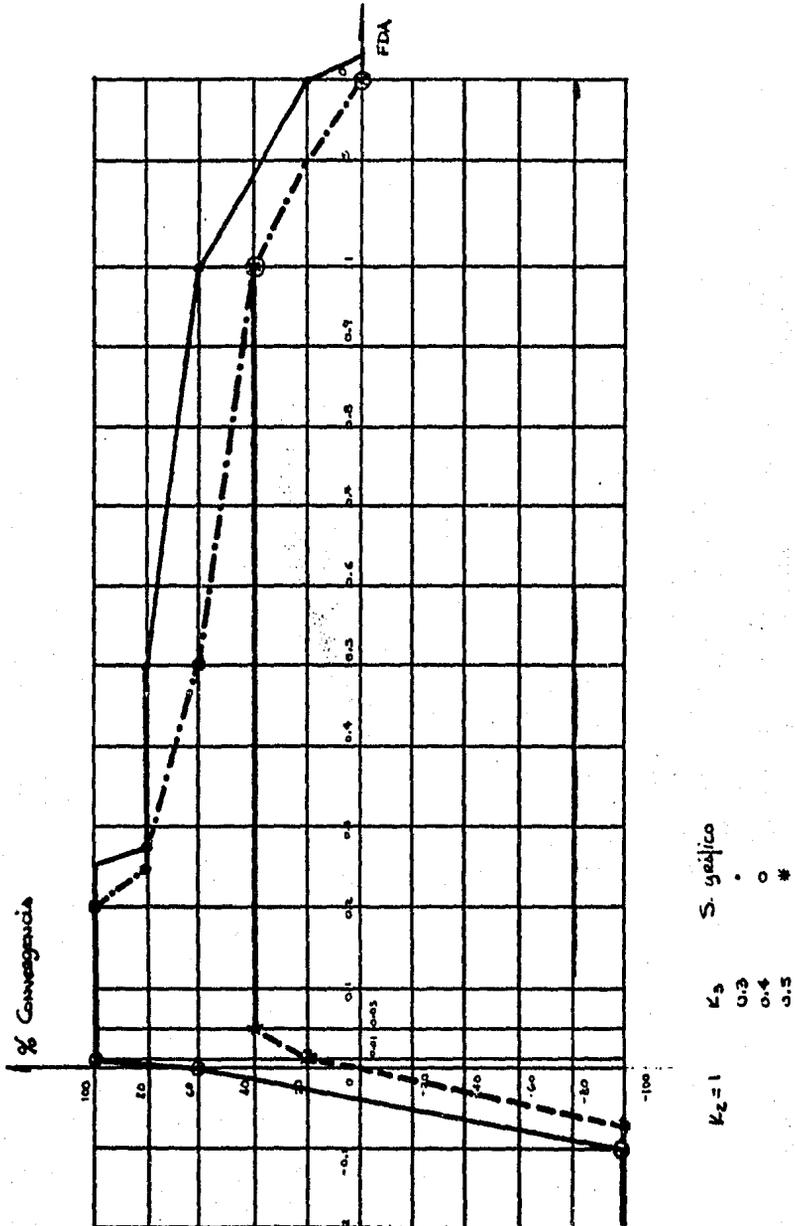


Figura 4-1: Gráficas de comportamiento del algoritmo ante la variación del peso del factor de acomodación

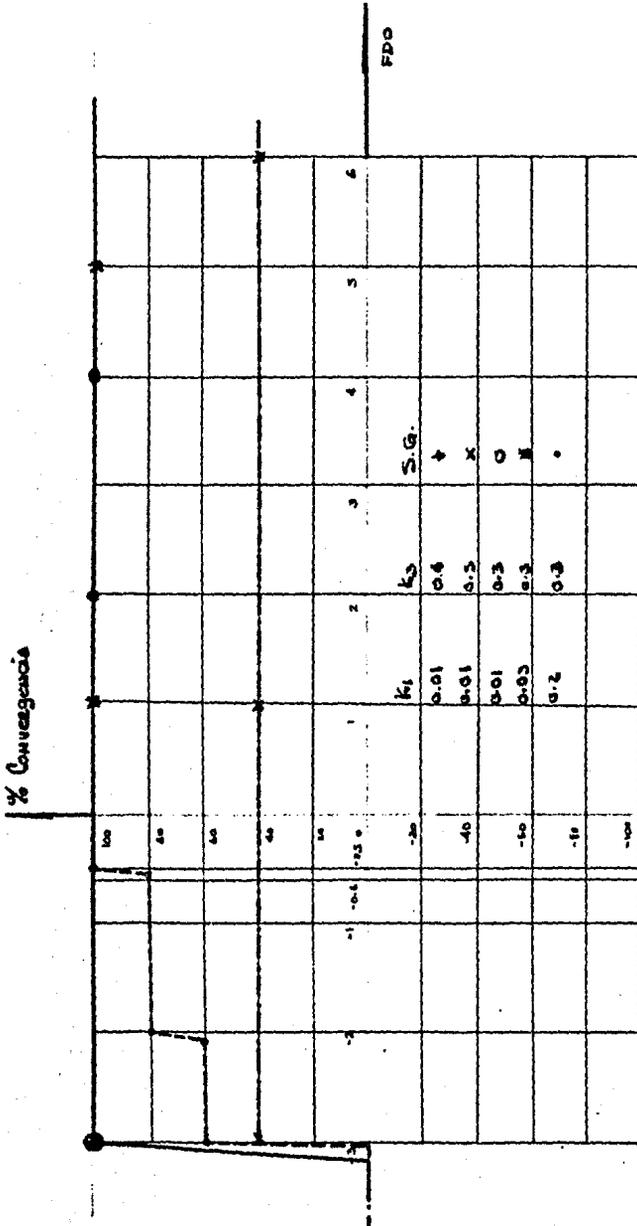


Figura 4-2: Gráficas de comportamiento del algoritmo ante la variación del peso del factor de acción de otras neuronas

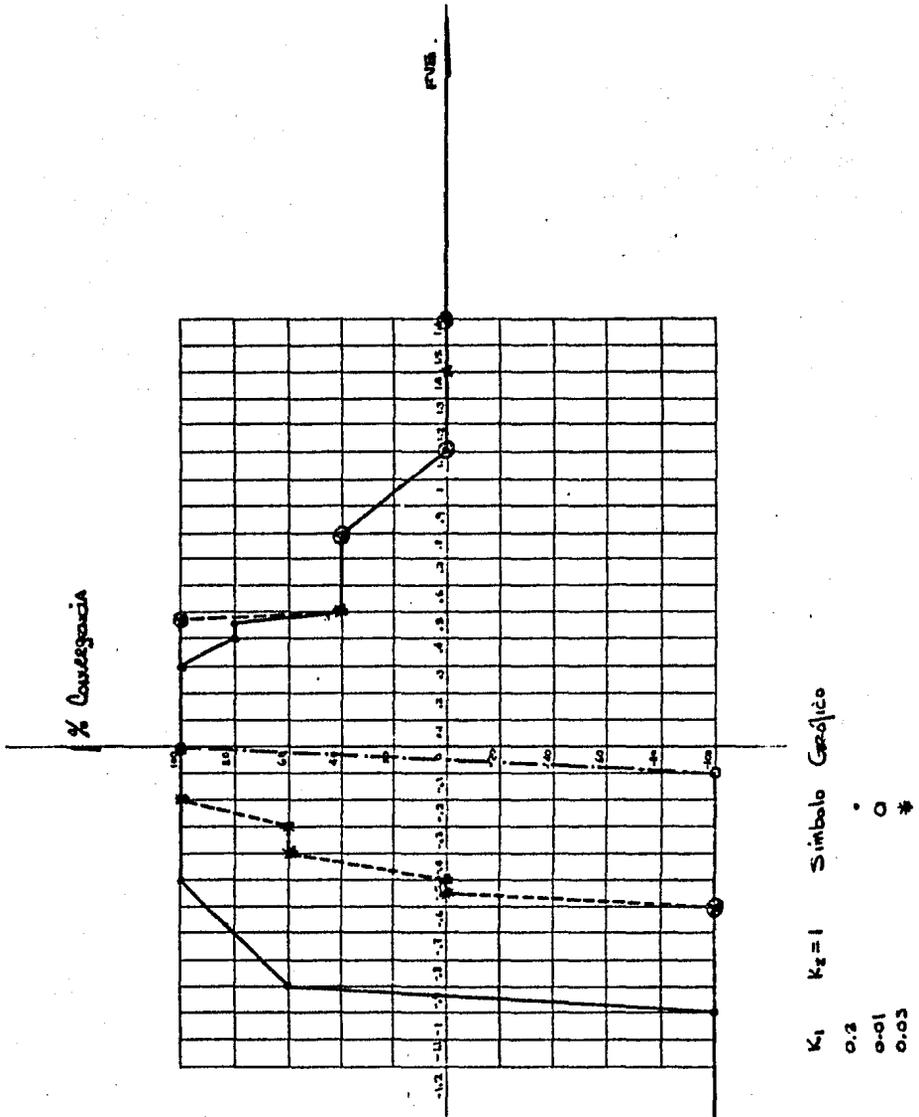


Figura 4-3: Gráficas de comportamiento del algoritmo ante la variación del peso del factor de acción de vecinas

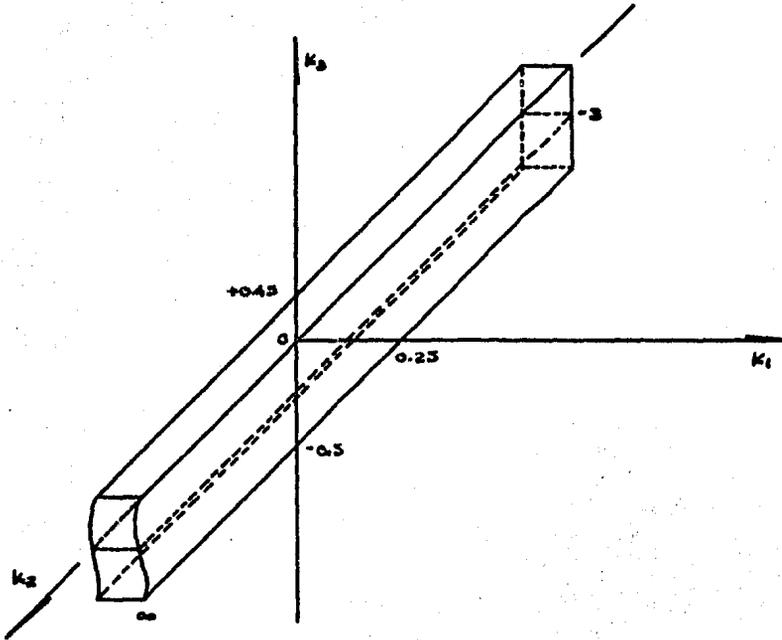


Figura 4-4: Espacio Tridimensional de Convergencia

## 5. Conclusiones

### 5.1 Evaluación del Modelo

Como se estableció en la última sección del capítulo 3 , y en el capítulo 4, el modelo se pudo ajustar de forma que los errores en la detección de profundidad no fueran representativos, aunque en ambientes complejos (de más de 10 elementos en el campo visual) o cuando los elementos se localizaban en posiciones alejadas (mayores que 15 veces la distancia entre los ojos) se observó que éstos aumentaban considerablemente.

El modelo se comporta aceptablemente en ambientes muy sencillos (menos de 5 objetos) cuando contiene sólo información binocular. Con la presencia de lentes que modifican la disparidad, se obtuvieron resultados que indicaron que la información binocular tiene prioridad sobre la monocular.

En general, el comportamiento del modelo concordó con los resultados experimentales, aunque cabe hacer notar que el grado de resolución que el modelo pueda alcanzar en la determinación de profundidad está en función de un incremento en el número de fotorreceptores; de los cuales se consideraron 12 para cada ojo en el modelo bidimensional, y 144 en el tridimensional.

Aunque nuestro modelo teórico no tiene un límite en el número de fotorreceptores que se tienen en el ojo, para implementarlo se utilizó un número reducido de los mismos, dado que esto simplificó considerablemente el proceso, y permitió una interpretación más sencilla de los resultados.

Si se implementara el modelo con un número mayor de fotorreceptores, los rangos de percepción de profundidad con mínimo

error se harían más amplios, sacrificándose con ésto sin embargo el grado de simplicidad; una implementación que considere pocos fotorreceptores se desenvuelve eficientemente en ambientes más o menos sencillos teniéndose entonces la ventaja de una implementación menos costosa pero de alcance limitado.

## 5.2 Comparacion y Contraste del Modelo con Otros Modelos

Los modelos con que intenta explicar y resolver el problema de percepción de profundidad tal como lo realizan los anuros, deben considerar las evidencias biológicas encontradas mediante la experimentación, además de posibles estructuras, cuya existencia se podrá demostrar posteriormente a la simulación.

El presente modelo ha integrado información anatómica del sistema visual de los anfibios e insectos que sugiere mecanismos específicos por medio de los cuales el sistema nervioso de éstos animales codificará la profundidad de los objetos, de ésta forma, el modelo se integra al grupo que considera mecanismos biológicos existentes además de otras estructuras que no han sido localizadas experimentalmente.

Entre los diversos modelos de percepción de profundidad, el modelo de Amari-Arbib, ha sido utilizado como base para modelos más completos y representativos, en particular el de Donald H. House. Este modelo considera cooperatividad entre elementos que codifican la misma profundidad, y competencia entre elementos que, al estar en la misma orientación codifican profundidades distintas. Además considera histéresis en los elementos, lo cual le permite tener al animal cierta capacidad de memoria con respecto a la escena en que se encuentra.

Si bien este modelo es de los más completos, las ecuaciones

## Conclusiones

que lo rigen no permiten un proceso rápido de convergencia. Y comparado con el nuestro, éste considera muchos más elementos para lograr la convergencia hacia un estado final. Matemáticamente nuestro modelo es una simplificación de los modelos de campo de proyección como el de House. En ambos casos se tienen mecanismos de iteración de cooperación y competencia semejantes y, por su propia estructura se presentan errores en la determinación exacta de dirección y profundidad de los objetos, debido principalmente a la poca cantidad de receptores considerados en cada retina que provocan el fenómeno de fragmentación espacial [4].

El problema de fragmentación está resuelto al colocar una gran cantidad de receptores en ambas retinas.

Los modelos de Marr y Poggio [7] así como el de Dev [3] proponen procesos de cooperatividad y competencia entre los diversos grupos de elementos para definir la profundidad de los objetos considerando que existen grupos de células que codifican diferentes profundidades a diferencia de nuestro modelo que utiliza solamente un grupo que permite codificar todas las profundidades.

El hecho de que los otros modelos utilicen niveles de células para cada profundidad, implica que su implementación sea mas compleja, y sugiere que probablemente no existan tales estructuras biológicas. Mientras que con el nuestro, existiendo sólo un estrato de células, se hace más factible tanto su implementación como su posible existencia biológica.

Si consideramos el estado actual de los mecanismos industriales de visión por computadora encontraremos que en la mayoría de los casos tienen limitadas sus funciones debido a que sus sistemas visuales son muy simples (sensores tipo sonar, que

equivaldrían a tener uno sólo de nuestros fotoreceptores)

Si se aplicara una implementación de nuestro modelo a la robótica, se lograrían un grado de inteligencia mayor en los robots, que ampliaría su campo de aplicaciones.

Las limitantes de poca resolución que presenta el modelo, considerando el ambiente biológico no son elementos definitivos, ya que un anuro percibe profundidad de objetos a no más de 1 metro, ya que le interesan los objetos localizados en un pequeño entorno.

En sí, el modelo comparado con los otros mencionados, resuelve el problema de percepción de profundidad de una manera más sencilla y con menos elementos. En el caso de los robots significaría aumentarle la complejidad a su sistema visual, pero el resultado de esto redundaría en un gran beneficio.

### 5.3 Etapas Futuras

Se pueden esperar dos cauces de investigación futura a partir de éste trabajo, uno de tipo biológico y el otro relacionado con la implementación para robótica del modelo propuesto.

**Investigación Biológica.** El modelo presenta un mecanismo de percepción de profundidad que integra información anatómica y fisiológica del sistema visual de los anfibios. Se sugiere con esto la existencia de estructuras específicas por medio de las cuales el sistema nervioso de estos animales realizaría la codificación de la profundidad de los objetos. Esto plantea hipótesis que pueden ser estudiadas experimentalmente.

El modelo nos ha permitido simular el funcionamiento del sistema visual ante diversas situaciones, corroborándose datos experimentales, pero también pudiéndose utilizar para tratar de

## Conclusiones

predecir el comportamiento del sistema ante situaciones nuevas.

**Investigación en Robótica.** El algoritmo propuesto en el presente modelo puede ser aplicado para el diseño de máquinas con capacidad de percibir la profundidad de los objetos por medio de mecanismos binoculares. Los elementos biológicos en que se basa el modelo pueden ser substituidos por dispositivos optoelectrónicos y módulos que realicen las funciones definidas matemáticamente para su aplicación en robótica.

## I. Manual de Usuarios del Sistema

Este manual describe la forma de utilizar el programa "SIMOJO", el cual constituye la implementación en computadora del modelo de ésta tesis.

La utilización de este sistema no requiere conocimiento de programación. El programa es totalmente interactivo, y por tanto el flujo de operación puede ser cambiado a preguntas del tipo "si/no".

Los datos numéricos suministrados pueden ser reales o enteros, separados por espacios.

El sistema está implementado en la computadora Foonly-F2.

Una vez en sesión, para iniciar una corrida debe especificarse el nombre del archivo ejecutable del sistema.

Los mensajes para que el usuario proporcione información son los siguientes:

\* usara lentes s/n?

Si se especifican lentes, deberán proporcionarse los siguientes datos:

\* tipo de lente:  
acercamiento aparente ... (1)  
alejamiento aparente ... (2)

La selección anterior invoca mecanismos para simular la acción de prismas.

\* angulo de desviación de la luz (de 0 a 30 grados)?

La siguiente información que requiere el sistema es:

\* distancia interocular?

Un valor adecuado podría ser de 4, que fué con el que se realizaron los experimentos en este trabajo.

La siguiente pregunta se refiere al valor de los factores que intervienen en el proceso:

\* factores de acomodación, acción de otras y acción de vecinas?

Para proporcionar la información adecuada en este punto, referirse al capítulo 4 (Análisis de Sensibilidad)

El siguiente requerimiento sería indicar si se requiere realizar un experimento monocular o binocular. El sistema preguntará:

\* experimento monocular s/n?

En caso afirmativo se realiza una simulación en la cual la acción del ojo izquierdo queda inhibida y sólo funcionará el derecho.

La siguiente fase es proporcionar las coordenadas de los estímulos:

\* coordenadas x,y,z para terminar <control-z>

Donde <control-z> significa oprimir las teclas "control" y "z" simultáneamente.

El origen del eje coordenado estará en el punto focal del ojo izquierdo, estando el ojo derecho en la parte positiva del eje x. Las únicas coordenadas que el sistema aceptará como válidas

Manual de Usuarios del Sistema  
serán aquellas que caigan en la región binocular del animal. Si  
algun dato no es válido, el sistema replicará.

Si se desea un experimento bidimensional, la componente z  
deberá ser 0.

Una vez dados estos datos, el programa comenzará a ejecutarse,  
y cada vez que se despliegue una matriz, el sistema preguntará si  
se desea graficar.

\* grafico s/n?

Para dar una respuesta afirmativa, se debe tener conectado un  
dispositivo de graficación.

Al terminar la simulación el sistema preguntará:

\* otra simulación s/n?

En caso afirmativo, el sistema comenzará a pedir de nuevo los  
datos.

Listado del Programa de Simulacion

II. Listado del Programa de Simulacion

Este es el listado del programa SIMOJO que realiza la simulación del modelo de percepción de profundidad. El programa esta realizado en el lenguaje Pascal y actualmente esta implementado en la computadora Foonly F2, aunque el codigo es estándar.

No se incluyen las rutinas de graficación, ya que no forman parte del modelo.

```

(*****
(*****
(**
(**          TITULO:          S I M O J O          (**
(**                                     (**
(**          USO:            Sirve para simular mecanismos visuales (**
(**                          de percepcion de profundidad en algunos (**
(**                          animales. (**
(**                                     (**
(**          COMENTARIOS:    Programa de tipo interactivo que va so- (**
(**                          licitando los datos que requiere. (**
(**                                     (**
(**          DATOS DE ENTRADA:  i) Tipo de simulacion (**
(**                          1.1) Con lentes (**
(**                          - acercamiento aparente (**
(**                          - alejamiento aparente (**
(**                          1.2) Sin lentes (**
(**                          1.3) Monocular (**
(**                          ii) Valor de algunas constantes usadas (**
(**                          por el algoritmo. (**
(**                          iii) Distancia entre los ojos. (**
(**                          iv) Coordenadas cartesianas de los (**
(**                          puntos a ser vistos. (**
(**                                     (**
(**          DATOS DE SALIDA:  Se va mostrando una matriz, que repre- (**
(**                          senta internamente el espacio por medio (**
(**                          de las relaciones entre las celulas re- (**
(**                          tinales, en sus diferentes etapas; y fi- (**
(**                          nalmente se muestra el medio que se per- (**
(**                          cibio. (**
(**                                     (**
(**          RUTINAS QUE LLAMA: sacomodacion, adotras, avecina, captura, (**
(**                          cooplan, graderiv, inicializacion,msuma, (**
(**                          neuroactivas, paresactivas, prueba, (**
(**                          visibles. (**
(**                                     (**
(**          FECHA:           Octubre 31 de 1984' (**
(**                                     (**
(**          AUTORES:        Alvarez Medina Gabriel Mauricio (**
(**                          Cardenas Garcia Sergio Ramon (**
(**                          Palomino Haddas Carlos (**
(**                                     (**

```

Listado del Programa de Simulacion

```
(**                                     **)
(**           F . I .           U . N . A . M .           **)
(**                                     **)
(*****
(*****)
```

```
program simojo(input,output);
```

```
const
  dimension = 3 ; (* dimensiones consideradas en el modelo *)
  distnegat = -1; (* distancia negativa utilizada para eliminar
                    puntos *)
  maxdesv   = 35; (* maximo valor de refraccion de los lentes *)
  numneur   = 12; (* numero de neuronas *)
  neurext   = 13; (* neurona extendida *)
  pi        = 3.14159;
  radian    = 0.01745;
  totcoord  = 50; (* numero maximo de puntos *)
```

```
type
  mat      = array [1..numneur,1..numneur] of real;
  matcu    = array [1..numneur] of mat;
  vreal    = array [1..numneur] of real;
  vente    = array [1..numneur] of integer;
  vecor    = array [1..totcoord,1..3] of real;
  vetcu    = array [1..numneur] of vente;
  vrecu    = array [1..numneur] of vreal;
```

```
var
  blente,    (* true si se utiliza lente para modificacion de vision*)
  btiplent,  (* true indica lente de acercamiento aparente
              false indica lente de alejamiento aparente *)
  ciclope,   (* true si solo se utiliza un ojo *)
  converge,  (* true si el algoritmo converge *)
  existe,    (* true cuando existen puntos en un plano *)
  finlec,    (* true para fin de lectura *)
  : boolean;
```

```
  i,        (* indice para las iteraciones *)
  kk,
  numcoord, (* numero de coordenadas capturadas y verificadas *)
  plano,    (* indicador del "plano" o region del espacio *)
  plncoord  (* numero de coordenadas en un plano *)
  : integer;
```

```
  continua  (* s/n indica si sigue o no con la simulacion *)
  : char;
```

```
  desviacion, (* angulo de refraccion del lente *)
  distojo,    (* distancia interocular *)
  fdotras,    (* factor que interviene en el proceso de la accion
              de otras neuronas *)
  fvecina,    (* factor que interviene en el proceso de la accion
              de la vecina *)
  facom       (* factor que interviene en la accion del factor de
              acomodacion *)
  : real;
```

```
  vtrld,     (* transmisores liberados del ojo derecho *)
```

Listado del Programa de Simulacion

```

vtrli      (* transmisores liberados del ojo izquierdo *)
           : vetcu;

vfac       (* factores de acomodacion del ojo izquierdo *)
           : vrecu;

plvcoord,  (* coordenadas de los puntos en un plano *)
vcoord     (* coordenadas de los puntos capturados y verificados *)
           : vecor;

mderi,     (* matriz de derivadas *)
mderd      (* matriz de derivadas para ojo derecho *)
           : matcu;

mdistancias (* matriz de distancias definida por los pares de
            : mat;
            neuronas *)
graf        (* archivo de graficacion *)
           : text;
    
```

```

(*****
(*)
(*)      s e c c i o n   d e   p r o c e d i m i e n t o s
(*)
(*****
(*)
(*)      TITULO:      MENSAJE
(*)
(*)      USO:         Despliega un mensaje indicando la posi-
(*)                  cion de un punto dado.
(*)
(*)      FORMA DE LLAMADO: mensaje(sw)
(*)
(*)      PARAMETROS DE ENTRADA: sw : selector de mensaje de error.
(*)
(*)      SALIDA:      Mensaje seleccionado de error.
(*)
(*)      RUTINAS QUE LLAMA: ---
(*)
(*****
    
```

```

procedure mensaje
( sw:integer );
begin
  case sw of
    1: writeln("region ciega izquierda.");
    2: writeln("solo veo con el ojo izquierdo.");
    3: writeln("region nasal.");
    4: writeln("solo veo con el ojo derecho.");
    5: writeln("region ciega derecha.");
    6: writeln("no veo.");
    7: writeln("region ciega superior o inferior");
  end
  ("endcase*");
end
("endpro mensaje*");
    
```

Listado del Programa de Simulacion

```

(*****)
(*)
(*)          TITULO:          FILTRACOORD          (*)
(*)
(*)          USO:            Acepta y filtra las  coordenadas de los (*)
(*)                          de los puntos.      (*)
(*)
(*)          FORMA DE LLAMADO:  filtracoord(numcoord, vcoord) (*)
(*)
(*)          PARAMETROS DE ENTRADA:  --- (*)
(*)
(*)          PARAMETROS DE SALIDA:  numcoord : numero de puntos binoculares. (*)
(*)                          vcoord   : vector con coordenadas de los (*)
(*)                          puntos. (*)
(*)
(*)          RUTINAS QUE LLAMA:  mensaje (*)
(*)
(*****)

```

procedure filtracoord

```

( var numcoord : integer;
  var vcoord   : vecor );

```

```

var i:integer;

```

```

  x,
  y,
  z,
  tanz,
  angizq,
  angder :real;

```

```

begin

```

```

  i:=1;
  numcoord:=0;
  read(x,y,z);
  finlec := false;
  while not eof do
    begin
      writeln(x,y,z);
      if y > 0 then
        begin
          tanz := z/y;
          (* posibilidad de estar en campo binocular *)
          (* calculo del angulo de inclinacion del segmento que
             va del centro del ojo al punto de coord x,y *)
          if (x = 0) or (x = distojo) then
            x := x + 0.01;
          angizq:=arctan(y/x)*180/pi;
          angder:=arctan(y/(x-distojo))*180/pi;
          if angder<0 then angder:=angder + 180;
          if angizq<0 then angizq:=angizq + 180;
          if (angizq > 135) and (angder > 135) then mensaje(1) else
          if (angizq > 45) and (angizq <= 135) and (angder > 135)
             then mensaje(2) else
          if (angder > 135) and (angizq <45) then mensaje(3) else
          if (angder<135) and (angder >= 45) and (angizq < 45) then
             mensaje(4) else
            if (angder < 45) and (angizq <= 45) then mensaje(5)
          else

```

Listado del Programa de Simulacion

```

begin
  if (tanz <= 1) and (tanz >= -1) then
    begin
      (* coordenada filtrada *)
      vcoord[i.1]:=x; (* coordenada x *)
      vcoord[i.2]:=y; (* coordenada y *)
      vcoord[i.3]:=z; (* coordenada z *)
      i:=i+1;
      numcoord:=numcoord+1
    end
    else mensaje(7)
  end
  (*endif*)
end
else mensaje(6);
(*endif*)
writeln("tecllee coordenada siguiente o "ctrl-z" para terminar");
read(x,y,z);
end
(*endwhile*);
reset(input)
end
(* endpro filtracoord *);

(*****
(*)
(*)          TITULO:          FILINT          (*)
(*)
(*)          USO:            Captura y filtra la distancia interocular (*)
(*)
(*)          FORMA DE LLAMADO: filint(distojo) (*)
(*)
(*) PARAMETROS DE ENTRADA:  --- (*)
(*)
(*) PARAMETROS DE SALIDA:  distojo : distancia entre los ojos. (*)
(*)
(*)          RUTINAS QUE LLAMA:  --- (*)
(*)
(*****

procedure filint
( var distojo : real );
begin
  repeat
    writeln("distancia interocular : ");
    read(distojo);
    writeln(distojo);
    writeln
  until distojo > 0
end
(*endpro filint*);

(*****
(*)
(*)          TITULO:          LENTE          (*)
(*)
(*)          USO:            Captura y filtra las características de (*)
(*)                          de los lentes. (*)
(*)
(*****

```

Listado del Programa de Simulacion

```

(*) FORMA DE LLAMADO: lente(blente,btiplent,desviacion) (*)
(*) (*)
(*) PARAMETROS DE ENTRADA: --- (*)
(*) (*)
(*) PARAMETROS DE SALIDA: blente : indica si usara o no lente. (*)
(*) btiplent : indica el tipo de lente. (*)
(*) desviacion : angulo de refraccion. (*)
(*) (*)
(*) RUTINAS QUE LLAMA: --- (*)
(*) (*)
(*****

```

```

procedure lente
( var blente,
  btiplent : boolean;
  var desviacion : real );

var ulente : char;
  tlente : integer;
begin
  blente:=false;
  btiplent :=false;
  desviacion := 0;
  writeln('usara lentes ? s/n ');
  repeat
    repeat
      read (ulente);
      until ulente <> ' ';
    until ulente in ['n','s'];
    if ulente = 's' then
      begin
        blente:=true;
        repeat
          writeln('tipo de lente ?');
          writeln(' acercamiento aparente....(1)');
          writeln(' alejamiento aparente....(2) : ');
          read(tlente);
          writeln(tlente);
          writeln
        until (tlente >= 1) and (tlente <= 2);
        btiplent:= tlente = 1;
        repeat
          writeln('angulo de refraccion 0 a 35 grados : ');
          read(desviacion);
          writeln(desviacion);
          writeln;writeln
        until ( desviacion >= 0 ) and ( desviacion <= 70 )
        end
      else blente:=false
        (*endif*)
      end
    end
  (*endpro lente*);

```

```

(*****
(*)
(*) TITULO: FACTORES (*)
(*) (*)
(*) USO: Captura los factores utilizados en los (*)
(*) procesos de factor de acomodacion accion (*)

```

Listado del Programa de Simulacion

```

(*)                                     de otras y accion de vecina                                     *)
(*)                                     *)
(*)          FORMA DE LLAMADO:          factores(facom, fdotras, fvecina)          *)
(*)                                     *)
(*) PARAMETROS DE ENTRADA:              ---                                     *)
(*)                                     *)
(*) PARAMETROS DE SALIDA:               facom : constante de acomodacion.          *)
(*)                                     fdotras : constante de accion de otras.          *)
(*)                                     fvecina : constante de accion de vecinas.          *)
(*)                                     *)
(*)          RUTINAS QUE LLAMA:         ---                                     *)
(*)                                     *)
(*****)

```

```

procedure factores
( var facom,
  fdotras,
  fvecina : real);
begin
  writeln('Factores que intervienen en el proceso de factor de acomodacion');
  writeln(' accion de otras y vecinas?');
  read(facom, fdotras, fvecina);
end
(* endpro factores *);

```

```

(*****)
(*)                                     *)
(*)          TITULO:                    CAPTURA                                     *)
(*)                                     *)
(*)          USO:                       Dirige el proceso de captura y filtrado          *)
(*)                                     de informacion utilizada                                     *)
(*)                                     *)
(*)          FORMA DE LLAMADO:          captura(blente, btiplent, ciclope,          *)
(*)                                     desviacion,distojo,facom,          *)
(*)                                     fdotras, fvecina,          *)
(*)                                     numcoord, vcoord)          *)
(*)                                     *)
(*) PARAMETROS DE ENTRADA:              ---                                     *)
(*)                                     *)
(*) PARAMETROS DE SALIDA:               blente      : indica si usara o no lente.          *)
(*)                                     btiplent   : indica el tipo de lente.          *)
(*)                                     ciclope    : indica experimento monocu -          *)
(*)                                     lar.          *)
(*)                                     desviacion : angulo de refraccion.          *)
(*)                                     distojo    : distancia entre ojos.          *)
(*)                                     facom      : constante de acomodacion.          *)
(*)                                     fdotras   : constante de accion de o -          *)
(*)                                     tras.          *)
(*)                                     fvecina   : constante de accion de ve -          *)
(*)                                     cinas.          *)
(*)                                     numcoord  : numero de puntos binocula -          *)
(*)                                     res.          *)
(*)                                     vcoord   : vector con coordenadas de          *)
(*)                                     los puntos.          *)
(*)                                     *)
(*)          RUTINAS QUE LLAMA:         factores, filint, filtracoord, lente.          *)
(*)                                     *)
(*****)

```

Listado del Programa de Simulacion

```

procedure captura
( var blente,
  btiplent,
  ciclope : boolean;
  var desviacion,
  distojo,
  facom,
  fdotras,
  fvecina : real;
  var numcoord : integer;
  var vcoord : vector );
var unojo : char; (*igual a "s" si se desea un solo ojo*)
begin
  (* captura de los parametros de especificacion de los lentes *)
  writeln('proporciona la siguiente informacion. ');
  writeln;
  lente(blente, btiplent, desviacion);
  filint(distojo);
  factores(facom, fdotras, fvecina);
  writeln('experimento monocular? s/n');
  repeat
    read(unojo)
  until unojo in ['n', 's'];
  ciclope:=unojo = 's';
  writeln('coordenadas x,y,z de los puntos. ');
  writeln('para finalizar teclee "ctl-z"');
  filtracoord(numcoord, vcoord)
end
(*endpro captura*);

```

```

(*****
(*)
(*)          TITULO:          CALDIST          (*)
(*)
(*)          USO:            Calcula la distancia desde el centro del (*)
(*)                          ojo izquierdo al punto especificado por (*)
(*)                          los angulos angder y angizq.          (*)
(*)
(*)          FORMA DE LLAMADO: caldist(distojo, angder, angizq) (*)
(*)
(*) PARAMETROS DE ENTRADA:   distojo : distancia entre ojos.      (*)
(*)
(*) PARAMETROS DE SALIDA:   angder  : angulo de inclinacion derecho. (*)
(*)                          angizq  : angulo de inclinacion izqu. (*)
(*)
(*)          RUTINAS QUE LLAMA: --- (*)
(*)
(*****

```

```

function caldist
( distojo : real;
  var angder,
  angizq : real ):real;
var aux : real;
begin
  (* se verifica que los angulos no sean identicos, pues se tendria
  una distancia al punto infinita *)
  if angder = angizq then angizq:=angizq+0.1;
  aux:=distojo/sin(( angder-angizq)*radian)*sin((180-angder)*radian);

```

Listado del Programa de Simulacion

```

caldist:=sqrt(sqr(aux)-aux*distojo*cos(angizq*radian)+
              (sqr(distojo)/4))
end
(*endpro caldist*);

(*****
*)
*)          TITULO:          INICIALIZACION          *)
*)
*)          USO:            inicializaciones necesarias. *)
*)
*)          FORMA DE LLAMADO: inicializacion(distojo, mderd, mderi, *)
*)                                mdistancias, vfac, *)
*)                                vtrld,vtrli) *)
*)
*)          PARAMETROS DE ENTRADA: distojo      : distancia entre ojos. *)
*)
*)          PARAMETROS DE SALIDA:  mderd       : matriz de derivadas del ojo *)
*)                                derecho. *)
*)                                mderi       : matriz de derivadas del ojo *)
*)                                izquierdo. *)
*)                                mdistancias : matriz de distancias. *)
*)                                vfac       : vector de factores de aco - *)
*)                                modacion. *)
*)                                vtrld     : vector de transmisores li - *)
*)                                berados derechos *)
*)                                vtrli    : vector de transmisores li - *)
*)                                berados izquierdos. *)
*)
*)          RUTINAS QUE LLAMA:    caldist. *)
*)
*)
*)
(*****)

procedure inicializacion
( distojo      : real;
  var mderd,
      mderi    : matcu;
      mdistancias : mat;
  var vfac     : vrecu;
  var vtrld,
      vtrli   : vetcu );
var
  i,
  j,
  k,
  lim : integer;
  angder,
  angizq : real;

begin
  (* inicializacion de las matrices de derivadas para ambos ojos *)
  for k := 1 to numneur do
    for i := 1 to numneur do
      for j := 1 to numneur do
        begin
          mderd[k][i,j] := 0;
          mderi[k][i,j] := 0
        end;
      end;
    end;
  (*endfor*)

```

Listado del Programa de Simulación

```
(* inicializacion del factor de acomodacion *)
for k:=1 to numneur do
  for i:=1 to numneur do
    begin
      vfac[k][i] :=-10*i;
      vtrld[k][i]:=0;
      vtrli[k][i]:=0
    end;
  (*endfor*)
  (* generacion de la matriz de distancias *)
  for i:=1 to numneur do
    for j:=1 to numneur do
      begin
        angder:=i*7.5+41.25;
        angizq:=138.75-j*7.5;
        mdistancias[i,j]:=caldist(distojo,angder,angizq)
      end;
    (*endfor*)
  (*endfor*)

  (* a las combinaciones de receptores que no pueden ocurrir,se les
  asigna un valor negativo de distancia al centro del ojo *)
  lim:=numneur;
  for i:= 2 to numneur-1 do
    begin
      for j:=numneur downto lim do
        mdistancias[i,j]:=distnegat;
        lim:=lim-1
      end
    (*endfor*)
  end
(*endpro inicializacion*);
```

```
(*****
(*)
(*)          TITULO:          CORRIMIENTO          (*)
(*)
(*)          USO:            Realiza los corrimientos necesarios en (*)
(*)                          vtrli y vtrld para simular la accion del (*)
(*)                          lente.                (*)
(*)
(*)
(*)          FORMA DE LLAMADO: corrimiento(corr, movneur, vtrld, vtrli) (*)
(*)
(*)          PARAMETROS DE ENTRADA: corr      : indica sentido del corrimiento. (*)
(*)                          movneur : numero de neuronas a recorrer. (*)
(*)
(*)          PARAMETROS DE SALIDA: vtrld      : vector de transmisores li - (*)
(*)                          berados derechos (*)
(*)                          vtrli       : vector de transmisores li - (*)
(*)                          berados izquierdos. (*)
(*)
(*)          RUTINAS QUE LLAMA: --- (*)
(*)
(*)
(*****)
```

```
procedure corrimiento
( corr,
  movneur : integer;
```

Listado del Programa de Simulacion

```

var
  vtrli,
  vtrld : vente );
var
  i, transfer : integer;
  vecaux, vecaux2 : vente;
begin
  for i := 1 to numneur do
    begin
      vecaux[i] := 0;
      vecaux2[i] := 0;
    end;
  (*endfor*)
  transfer := 1;
  for i := movneur + 1 to numneur do
    begin
      if corr = 0 then
        begin
          (*corrimento a la izquierda*)
          vecaux[transfer] := vtrli[i];
          vecaux2[transfer] := vtrld[i];
          transfer := transfer + 1;
        end
      else
        begin
          (*corrimento a la derecha*)
          vecaux[i] := vtrli[transfer];
          vecaux2[i] := vtrld[transfer];
          transfer := transfer + 1;
        end;
      end(*for*);
    for i := 1 to numneur do
      begin
        (*actualizacion de los corrimentos*)
        vtrli[i] := vecaux[i];
        vtrld[i] := vecaux2[i];
      end;
    end(*endfor*);
  end
  (*endpro corrimento*);

(*****
(*)
(*)          TITULO:          PARESACTIVAS          (*)
(*)
(*)          USO:            mderi = vtrld * vtrli, es decir, determi- (*)
(*)                          na los pares de neuronas activas posibles (*)
(*)                          (*)
(*)                          (*)
(*)          FORMA DE LLAMADO: paresactivas(distojo,vtrld,vtrli,mderi) (*)
(*)                          (*)
(*) PARAMETROS DE ENTRADA:   distojo      : distancia entre ojos.      (*)
(*)                          vtrld       : vector de transmisores li - (*)
(*)                          berados derechos (*)
(*)                          vtrli      : vector de transmisores li - (*)
(*)                          berados izquierdos. (*)
(*)                          (*)
(*) PARAMETROS DE SALIDA:   mder        : matriz de derivadas del ojo (*)
(*)

```

Listado del Programa de Simulacion

```
(*
(*          RUTINAS QUE LLAMA:      ---
(*
(*
(*.....*)
```

```
procedure paresactivas
(  distojo : real;
  vtrld,
  vtrli : vente;
  var mder : mat );
var
  i, j : integer;
begin
  writeln;writeln;
  writeln("  :10,-----pares activas "vtrld * vtrli"-----");
  writeln;writeln;
  for i := 1 to numneur do
    for j := 1 to numneur do
      mder[i,j] := vtrli[i]*vtrld[j]*distojo
    end
  end
(*endpro paresactivas *);
```

```
(*.....*)
(*
(*          TITULO:          GRADERIV
(*
(*          USO:            Realiza la graficacion histografica de la
(*                          matriz de derivadas
(*
(*          FORMA DE LLAMADO:  graderiv(mder)
(*
(*  PARAMETROS DE ENTRADA:    mder : matriz de derivadas del ojo.
(*
(*  PARAMETROS DE SALIDA:    ---
(*
(*          SALIDA:          Impresion de la matriz de derivadas.
(*
(*          RUTINAS QUE LLAMA:  ---
(*
(*.....*)
```

```
procedure graderiv
( mder : mat );
var
  i, j : integer;
begin
  writeln;
  write("i\d");
  for i := 1 to numneur do
    write(i:4," ");
  writeln;
  for i := 1 to numneur do
    begin
      write(i:2);
      for j := 1 to (numneur+1-1) do
        write(mder[i,j]:6:1);
      writeln;
    end
  end
```

Listado del Programa de Simulacion

```
end
(* endpro graderiv *);
```

```
(*****
(*)
(*)          TITULO:          NEUROACTIVAS          (*)
(*)
(*)          USO:            Determina las neuronas que se activaran
(*)                          en funcion de los puntos aceptados.
(*)
(*)          FORMA DE LLAMADO: Neuroactivas(blente,btiplemt,desviacion,
(*)                          distojo, numcoord, vcoord,
(*)                          vfac,vtrld,vtrli)
(*)
(*)          PARAMETROS DE ENTRADA;   blente      : indica si usara o no lente.
(*)                          btiplemt   : indica el tipo de lente.
(*)                          desviacion : angulo de refraccion.
(*)                          distojo    : distancia entre ojos.
(*)                          numcoord   : numero de puntos binocula-
(*)                                  res.
(*)                          vcoord     : vector con coordenadas de
(*)                                  los puntos.
(*)
(*)          PARAMETROS DE SALIDA:   vfac       : vector de factores de aco -
(*)                                  modacion.
(*)                                  vtrld   : vector de transmisores li -
(*)                                  berados derechos
(*)                                  vtrli   : vector de transmisores li -
(*)                                  berados izquierdos.
(*)
(*)          RUTINAS QUE LLAMA:      corrimiento, graderiv.
(*)
(*)
(*****)
```

procedure neuroactivas

```
( blente,
  btiplemt : boolean;
  desviacion,
  distojo : real;
  numcoord : integer;
  vcoord : vecor;
  var
  vfac : vreal;
  var
  vtrld,
  vtrli : vente );
var
  i, j, naci, nacd, movneur : integer;
  angder, angizq : real;
  mdreal : mat;
begin
  for i := 1 to numneur do
    for j := 1 to numneur do
      mdreal[i,j] := 0;
    for i := 1 to numcoord do
      begin
        (*determinacion de angulos de coincidencia en el ojo
        correspondientes al punto considerado*)
```

Listado del Programa de Simulacion

```

anglq := arctan(vcoord[i,2]/vcoord[i,1])*180/pi;
angder := arctan(vcoord[i,2]/(vcoord[i,1]-distojo))
        * 180/pi;
if angder < 0 then angder := angder + 180;
if anglq < 0 then anglq := anglq + 180;
(*determinacion de las neuronas que corresponden al angulo
calculado *)
naci := trunc((anglq-45)/7.5)+1;
nacd := trunc((135-angder)/7.5)+1;
if naci < 1 then naci := 1;
if nacd < 1 then nacd := 1;
vtrli[naci]:= 1;
vtrld[nacd]:= 1;
mdreal[naci,nacd]:= 1;
(*el valor del factor de acomodacion es actualizado si
la distancia del punto considerado es menor que la
que la previa*)
if (vfac[naci] > mdistancias[naci, nacd]) or
(vfac[naci]<0) then
vfac[naci] := mdistancias[naci,nacd];
end;
(*endfor*)
graderiv(mdreal);
if blente then
begin
(*mapeo de la cantidad de difraccion en el ojo*)
movneur := trunc(desviacion/(90/numneur));
if btiplent then
(*lente de acercamiento*)
(*corrimiento a la izquierda*)
corrimiento(0, movneur, vtrld, vtrli)
else
corrimiento(1, movneur, vtrld, vtrli);
end;
(*endif*)
end
(* endpro neuroactivas *);

(*****
(*)
(*)          TITULO:          AVECINA          (*)
(*)
(*)          USO:            Introduce al proceso la accion de las ve- (*)
(*)                          cinas.          (*)
(*)
(*)          FORMA DE LLAMADO: Avecina(factor,mder) (*)
(*)
(*) PARAMETROS DE ENTRADA:   fvecina      : constante de accion de ve- (*)
(*)                          cinas.          (*)
(*)
(*) PARAMETROS DE SALIDA:   mder : matriz de derivadas del ojo. (*)
(*)
(*)          RUTINAS QUE LLAMA: --- (*)
(*)
(*****)

procedure avecina
( fvecina : real;
  var mder : mat );

```

Listado del Programa de Simulacion

```

type
mext = array[0..neurext,1..numneur] of real; (* matriz extendida *)
var
  maext : mext;
  i, j : integer;
begin
  for j := 1 to numneur do
    begin
      maext[0,j] := mder[1,j];
      maext[neurext,j] := mder[numneur,j]
    end
  (* endfor *);
  for i := 1 to numneur do
    for j := 1 to numneur do
      maext[i, j] := mder[i, j]
    (* endfor *);
    writein;writeln;
    writeln("  :17,-----accion de vecina-----");
    writeln;
    for i:= 1 to numneur do
      for j := 1 to numneur do
        mder[i,j] := maext[i,j]
          - fvecina*(abs(maext[i,j] - maext[i+1,j]) + abs(maext[i,j] -
            maext[i-1,j]))
      end
    end
  (* endpro avecina *);

(*****
(*)
(*)          TITULO:          ADAPTAM          (*)
(*)
(*)          USO:            Adapta la matriz de diferencias de mane- (*)
(*)                          ra que las diagonales que contengan va- (*)
(*)                          lores nulos son decrementadas y las que (*)
(*)                          no son aumentadas. (*)
(*)
(*)          FORMA DE LLAMADO: adaptam(mdife) (*)
(*)
(*)          PARAMETROS DE ENTRADA: --- (*)
(*)
(*)          PARAMETROS DE SALIDA: mdife : matriz de diferencias. (*)
(*)
(*)          RUTINAS QUE LLAMA: --- (*)
(*)
(*****

```

```

procedure adaptam
( var mdife : mat);
  var vecm : vreal;
  i,
  j,
  k,
  l : integer;

begin
  for i := 1 to numneur do
    vecm[i] := 9999;
  for i := 1 to numneur do
    for j := 1 to (numneur - i + 1) do

```

Listado del Programa de Simulacion

```

if (mdife[i,j] < vecm[i]) and (mdife[i,j] <> 0) then
  vecm[i] := mdife[i,j];
for i := 1 to numneur do
  for j := 1 to (numneur - i + 1) do
    if mdife[i,j] = 0 then
      begin
        k := 1;
        l := 1;
        if i > j then
          k := i - j + k
        else if j > i then
          l := j - i + l;
        while (k + l) < (numneur + 1) do
          begin
            mdife[k,l] := vecm[k] / 2;
            k := k + 1;
            l := l + 1
          end; (* while *)
        end; (* if *)
      end
    endpro adaptam *);
(*****
(*)
(*)          TITULO:          ACOMODACION          (*)
(*)
(*)          USO:            Introduce la accion del factor de acomoda- (*)
(*)                          cion al proceso.          (*)
(*)
(*)          FORMA DE LLAMADO:  aacomodacion(facom,mdistancias,vfac,mder) (*)
(*)
(*)  PARAMETROS DE ENTRADA:    facom           : constante de acomodacion. (*)
(*)                          mdistancias : matriz de distancias.      (*)
(*)
(*)                          vfac          : vector de factores de aco - (*)
(*)                          modacion.          (*)
(*)
(*)  PARAMETROS DE SALIDA:    mder : matriz de derivadas del ojo.      (*)
(*)
(*)          RUTINAS QUE LLAMA:  adaptam, mdife.          (*)
(*)
(*)
(*****

```

```

procedure aacomodacion
( facom : real;
  mdistancias : mat;
  vfac : vreal;
  var mder : mat );
var
  i, j : integer;
  mdife : mat;
begin
  writeln;writeln;
  writeln(' :8.-----accion del factor de acomodacion-----');
  writeln;
  for i := 1 to numneur do
    for j := 1 to numneur do
      mdife[i,j] := facom * abs(mdistancias[i,j] - vfac[i]);
    (* endfor *)
  end;
end;

```

Listado del Programa de Simulacion

```

adaptam(mdife);
for i := 1 to numneur do
  for j := 1 to numneur do
    mder[i,j] := mder[i,j] - mdife[i,j];
  (* endfor *)
end
(* endpro acomodacion *);

```

```

(*****
(*)
(*)          TITULO:          PRUEBA          (*)
(*)
(*)          USO:            Esta rutina obtiene los promedios de los (*)
(*)                          renglones de mder y a los valores de un (*)
(*)                          renglon que no pasen del promedio los ha- (*)
(*)                          ce zero.          (*)
(*)
(*)          FORMA DE LLAMADO: prueba(mder, converge) (*)
(*)
(*)          PARAMETROS DE ENTRADA:  --- (*)
(*)
(*)          PARAMETROS DE SALIDA:  converge: verdadera si el algoritmo con - (*)
(*)                          verge.          (*)
(*)                          mder      : matriz de derivadas del ojo. (*)
(*)
(*)          RUTINAS QUE LLAMA:    --- (*)
(*)
(*****

```

```

procedure prueba
( var mder : mat ;
  var converge : boolean);
var
  cont, i, j : integer;
  sigma, suma2, umbral : real;
begin
  for i := 1 to numneur do
    begin
      sigma := 0;
      cont := 0;
      for j := 1 to (numneur-1) do
        if mder[ i,j ] <> 0 then
          begin
            cont := cont + 1;
            sigma := sigma + mder[i,j];
          end;
      suma2 := 0;
      for j := 1 to numneur do
        if mder[ i,j ] <> 0 then
          begin
            mder[i,j] := mder[i,j]-(sigma/cont);
            suma2 := suma2 + mder[i,j];
          end;
      if cont <> 0 then
        umbral := suma2/cont;
      if cont = 1 then
        umbral := 0;
      for j := 1 to numneur do

```

Listado del Programa de Simulacion

```

adaptam(mdife);
for i := 1 to numneur do
  for j := 1 to numneur do
    mder[i,j] := mder[i,j] - mdife[i,j];
  (* endfor *)
end
(* endpro acomodacion *);

```

```

(*****
(*)
(*)          TITULO:          PRUEBA          (*)
(*)
(*)          USO:            Esta rutina obtiene los promedios de los (*)
(*)                      renglones de mder y a los valores de un (*)
(*)                      renglon que no pasen del promedio los ha- (*)
(*)                      ce zero. (*)
(*)
(*)          FORMA DE LLAMADO: prueba(mder, converge) (*)
(*)
(*)          PARAMETROS DE ENTRADA:  --- (*)
(*)
(*)          PARAMETROS DE SALIDA:  converge: verdadera si el algoritmo con - (*)
(*)                      verge. (*)
(*)                      mder      : matriz de derivadas del ojo. (*)
(*)
(*)          RUTINAS QUE LLAMA:    --- (*)
(*)
(*****

```

```

procedure prueba
( var mder : mat ;
  var converge : boolean);
var
  cont, i, j : integer;
  sigma, suma2, umbral : real;
begin
  for i := 1 to numneur do
    begin
      sigma := 0;
      cont := 0;
      for j := 1 to (numneur-1) do
        if mder[ i,j ] <> 0 then
          begin
            cont := cont + 1;
            sigma := sigma + mder[i,j];
          end;
      suma2 := 0;
      for j := 1 to numneur do
        if mder[ i,j ] <> 0 then
          begin
            mder[i,j] := mder[i,j]-(sigma-mder[i,j]);
            suma2 := suma2 + mder[i,j];
          end;
      if cont <> 0 then
        umbral := suma2/cont;
      if cont = 1 then
        umbral := 0;
      for j := 1 to numneur do

```

Listado del Programa de Simulacion

```

(*****
(*)
(*)          TITULO:          VISIBLES          (*)
(*)
(*)          USO:             Realiza la ultima etapa tendiente a de - (*)
(*)                          terminar los puntos visibles por el anuro (*)
(*)
(*)          FORMA DE LLAMADO: visibles(mder, vtrli) (*)
(*)
(*)          PARAMETROS DE ENTRADA: vtrli      : vector de transmisores libera - (*)
(*)                                  dos izquierdos. (*)
(*)
(*)          PARAMETROS DE SALIDA: mder       : matriz de derivadas del ojo. (*)
(*)
(*)          RUTINAS QUE LLAMA:  --- (*)
(*)
(*****

```

```

procedure visibles
( vtrli : vente;
  var mder : mat );
  const
    umbral = -1;
  var
    i, imax, j, jmax, suma : integer;
    maximo : real;
  begin
    suma := 0;
    for i := 1 to numneur do
      suma := suma + vtrli[i];
      suma := suma * umbral;
      for j := 1 to numneur do
        begin
          maximo := suma;
          for j := 1 to numneur do
            if mder[i,j] > maximo then
              begin
                maximo := mder[i,j];
                imax := i;
                jmax := j;
              end;
            if maximo <= suma then
              for j := 1 to numneur do
                mder[i,j] := 0;
              end;
          for i := 1 to numneur do
            for j := (numneur - i + 1) to numneur do
              mder[i,j] := 0;
            writeln;writeln;
            graderiv( mder );
          end
        endpro visibles *);

```

```

(*****
(*)
(*)          TITULO:          MAPDER          (*)
(*)
(*)          USO:             Hace un mapeo de los puntos del espacio (*)
(*)                          visual con el fin de utilizar las ruti- (*)
(*)

```

Listado del Programa de Simulacion

```

(*) nas centradas en el ojo izquierdo para *)
(*) el ojo derecho. *)
(*) *)
(*) FORMA DE LLAMADO: mapder(distojo, numcoord, vcoord) *)
(*) *)
(*) PARAMETROS DE ENTRADA: distojo : distancia entre ojos. *)
(*) numcoord : numero de puntos binocula- *)
(*) res. *)
(*) *)
(*) PARAMETROS DE SALIDA: vcoord : vector con coordenadas de *)
(*) los puntos. *)
(*) *)
(*) RUTINAS QUE LLAMA: --- *)
(*) *)
(*****)

```

```

procedure mapder
( distojo : real;
  numcoord : integer;
  var vcoord : vecor );

var
  i : integer;

begin
  for i:= 1 to numcoord do
    vcoord[i,1] := distojo - vcoord[i,1];
  end
(* endpro mapder *);

```

```

(*****)
(*) *)
(*) TITULO: TRANSP *)
(*) *)
(*) USO: Transpone la matriz "mder". Esta transpo- *)
(*) sicion es util cuando se realiza la parte *)
(*) del algoritmo centrada en el ojo dere- *)
(*) cho. *)
(*) *)
(*) *)
(*) FORMA DE LLAMADO: transp(mder) *)
(*) *)
(*) PARAMETROS DE ENTRADA: --- *)
(*) *)
(*) PARAMETROS DE SALIDA: mder : matriz de derivadas delojo. *)
(*) *)
(*) RUTINAS QUE LLAMA: --- *)
(*) *)
(*****)

```

```

procedure transp
( var
  mder : mat);

var
  i,j : integer;
  mataux : mat;

begin

```

Listado del Programa de Simulacion

```

for i := 1 to numneur do
  for j := 1 to numneur do
    mataux[i,j] := mder[i,j];
  for i := 1 to numneur do
    for j := 1 to numneur do
      mder[i,j] := mataux[j,i];
    end
  endpro transp *);
(*****
(*)
(*)          TITULO:          OJODERE          (*)
(*)
(*)          USO:            Realiza todo el algoritmo centrado en el (*)
(*)                          ojo derecho.      (*)
(*)
(*)          FORMA DE LLAMADO:  ojodere(blente, btiplent, converge, (*)
(*)                          desviacion, distojo, (*)
(*)                          facom, fdotras, fvecina, mderd, (*)
(*)                          mdistancias, numcoord, vcoord, (*)
(*)                          vfac, vtrld, vtrli) (*)
(*)
(*)          PARAMETROS DE ENTRADA:  blente      : indica si usara o no lente. (*)
(*)                          btiplent   : indica el tipo de lente. (*)
(*)                          desviacion : angulo de refraccion. (*)
(*)                          mdistancias: matriz de disancias. (*)
(*)
(*)          PARAMETROS DE SALIDA:  converge   : verdadero si el algoritmo (*)
(*)                          converge. (*)
(*)                          distojo     : distancia entre ojos. (*)
(*)                          facom      : constante de acomodacion. (*)
(*)                          fdotras    : constante de accion de o- (*)
(*)                          tras. (*)
(*)                          fvecina    : constante de accion de ve- (*)
(*)                          cinas. (*)
(*)                          mderd     : matriz de derivadas del ojo (*)
(*)                          derecho. (*)
(*)                          numcoord   : numero de puntos binocula- (*)
(*)                          res. (*)
(*)                          vcoord    : vector con coordenadas de (*)
(*)                          los puntos. (*)
(*)                          vfac      : vector de factores de acom. (*)
(*)                          vtrld     : vector de trans. liberado (*)
(*)                          derecho. (*)
(*)                          vtrli    : vector de trans. liberado (*)
(*)                          izquierdo. (*)
(*)                          derecho. (*)
(*)
(*)          RUTINAS QUE LLAMA:  sacomodacion, adotras, avecina, graderiv, (*)
(*)                          mapder, neuroactivas, paresactivas, prueba, (*)
(*)                          visibles. (*)
(*)
(*)
(*)
(*****
)
procedure ojodere
( blente,
  btiplent,
  converge : boolean;

```

## Listado del Programa de Simulacion

```

desviacion,
distojo,
facom,
fdotras,
fvecina : real;

var
  mder : mat;

  mdistancias : mat;

  numcoord : integer;

var
  vcoord : vecor;

  vfac : vreal;

  vtrld,
  vtrli : vente);

var
  i : integer;

begin
  for i:= 1 to numneur do
    begin
      vfac[i] := -10 * i;
      vtrld[i] := 0;
      vtrli[i] := 0;
    end;
  (* endfor *);
  i := 1;
  mapder(distojo, numcoord, vcoord);
  neuroactivas( blente, btiplent, desviacion, distojo, numcoord
    , vcoord, vfac, vtrld, vtrli);
  paresactivas(distojo, vtrld, vtrli, mder);
  transp(mder);
  graderiv(mder);
  transp(mder);
  sdotras(fdotras, vtrli, mder );
  transp(mder);
  graderiv(mder);
  transp(mder);
  avecina( fvecina, mder );
  transp(mder);
  graderiv(mder);
  transp(mder);
  acomodacion( facom, mder, vfac, mdistancias );
  transp(mder);
  graderiv(mder);
  transp(mder);
  visibles(vtrli, mder);
  converge := false;
  writeln;
  writeln(' :7, *** comienzan las iteraciones para el ojo derecho ***');
  writeln;
  while not converge do

```

Listado del Programa de Simulacion

```

begin
  writeln;writeln(' iteracion # ',i);writeln;
  prueba(mder, converge);
  transp(mder);
  graderiv(mder);
  transp(mder);
  i := i + 1
end
(* endwhile *);
transp(mder);
graderiv(mder);
end
(* endpro ojodere *);

```

```

(*****)
(*
(*          TITULO:          OJOIZQ
(*
(*          USO:            Realiza todo el algoritmo centrado en el
(*                          ojo izquierdo.
(*
(*          FORMA DE LLAMADO:  ojoizq(blente, btiplent, ciclope, converge,
(*                          desviacion, distojo,
(*                          facom, fdotras, fvecina, mderi,
(*                          mdistancias, numcoord, vcoord,
(*                          vfac, vtrld, vtrli)
(*
(*          PARAMETROS DE ENTRADA:  blente      : indica si usara o no lente.
(*                          btiplent   : indica el tipo de lente.
(*                          ciclope    : verdadero si sera experimento
(*                                      binocular.
(*                          desviacion : angulo de refraccion.
(*                          mdistancias: matriz de disancias.
(*
(*          PARAMETROS DE SALIDA:  converge   : verdadero si el algoritmo
(*                                      converge.
(*                          distojo    : distancia entre ojos.
(*                          facom      : constante de acomodacion.
(*                          fdotras    : constante de accion de o-
(*                                      tras.
(*                          fvecina    : constante de accion de ve-
(*                                      cinas.
(*                          mderi      : matriz de derivadas del ojo
(*                                      derecho.
(*                          numcoord   : numero de puntos binocula-
(*                                      res.
(*                          vcoord     : vector con coordenadas de
(*                                      los puntos.
(*                          vfac       : vector de factores de acom.
(*                          vtrld      : vector de trans. liberado
(*                                      derecho.
(*                          vtrli      : vector de trans. liberado
(*                                      izquierdo.
(*                                      derecho.
(*
(*          RUTINAS QUE LLAMA:  acomodacion, adotras, avecina,graderiv,
(*                          neuroactivas,paresactivas,prueba,
(*                          visibles.
(*
(*****

```

Listado del Programa de Simulacion

(\*  
 (\*\*\*\*\*)

```

procedure ojoizq
(
  blente,
  btiplent,
  ciclope,
  converge : boolean;

  desviacion,
  distojo,
  facom,
  fdotras,
  fvecina : real;

var
  mderi : mat;

  mdistancias : mat;

  numcoord : integer;

var
  vcoord : vecor;

  vfac : vreal;

  vtrld,
  vtrli : vente);

var
  i, kk : integer;

begin
  neuroactivas( blente, btiplent, desviacion, distojo, numcoord
    , vcoord, vfac, vtrld, vtrli);
  if ciclope then
    for kk:=1 to numneur do vtrli[kk]:=0;
  paresactivas(distojo, vtrld, vtrli, mderi);
  graderiv( mderi );
  adotras(fdotras, vtrli, mderi );
  graderiv( mderi );
  avecina( fvecina, mderi );
  graderiv( mderi );
  acomodacion( facom, mderi, vfac, mdistancias );
  graderiv( mderi );
  visibles(vtrli, mderi);
  converge := false;
  writeln;
  writeln(' :7, '*** comienzan las iteraciones para el ojo izq. ***');
  writeln;
  i := 0;
  while not converge do
    begin
      writeln; writeln(' iteracion # ', i); writeln;
      prueba(mderi, converge);
      graderiv( mderi );
      i := i + 1
    end
  end
end
  
```

Listado del Programa de Simulacion

```
(* endwhile *);
end(*ojoizq*);
```

```
(*****
(*)
(*)          TITULO:          MSUMA          (*)
(*)          USO:            Realiza la suma de mderd y mderi. (*)
(*)          FORMA DE LLAMADO: msuma(mderi,mderd) (*)
(*)          PARAMETROS DE ENTRADA: mderi      : matriz de derivadas del ojo (*)
(*)                                     izquierdo. (*)
(*)          mderd          : matriz de derivadas del ojo (*)
(*)                                     derecho. (*)
(*)          PARAMETROS DE SALIDA: mderi      : resultado (*)
(*)          RUTINAS QUE LLAMA:  --- (*)
(*)          *****
```

```
procedure msuma
(var
  mderi : mat;
  mderd : mat);

var
  i,j : integer;

begin
  for i := 1 to numneur do
    for j := 1 to numneur do
      mderi[i,j] := mderi[i,j] + mderd[i,j];
    end
  end
  * endpro msuma *);
```

```
(*****
(*)
(*)          TITULO:          COOPLAN          (*)
(*)          USO:            Determina coordenadas de puntos en cada (*)
(*)                          "plano" (*)
(*)          FORMA DE LLAMADO: cooplan(numcoord, plano, vcoord, (*)
(*)                          existe, plncoord,plvcoord) (*)
(*)          PARAMETROS DE ENTRADA: numcoord : numero de coordenadas capturadas (*)
(*)          plano          : indicador de "plano" o region (*)
(*)          vcoord        : coordenadas de los puntos (*)
(*)          PARAMETROS DE SALIDA: existe   : true cuando existen puntos en (*)
(*)                          un plano (*)
(*)          plncoord      : numero de coord. en un plano (*)
(*)          plvcoord      : cordenadas puntos en un plano (*)
(*)          RUTINAS QUE LLAMA:  --- (*)
(*)          *****
```

Listado del Programa de Simulacion

```

procedure cooplan
( numcoord, plano : integer;
  vcoord           : vecor;
  var
    existe         : boolean;
  var
    plncoord       : integer;
  var
    plvcoord       : vecor);

var
  liminf, limsup  : real;
  i : integer;
begin
  if cos(pi/4 - plano * pi/24) <> 0 then
    liminf := sin(pi/4 - plano * pi/24) / cos(pi/4 - plano * pi/24);
  if cos(pi/4 - (plano - 1) * pi/24) <> 0 then
    limsup := sin(pi/4 - (plano - 1) * pi/24)/cos(pi/4 - (plano-1)*pi/24);
  plncoord := 0;
  for i := 1 to numcoord do
    if ((vcoord[i,3]/vcoord[i,2])<=limsup) and ((vcoord[i,3]/vcoord[i,2]) >
      liminf) then
      begin
        plncoord := plncoord + 1;
        plvcoord[plncoord,1] := vcoord[i,1];
        plvcoord[plncoord,2] := sqrt(sqrt(vcoord[i,2])+sqr(vcoord[i,3]));
        existe := true
      end
    end
  end(*cooplan*);

```

```

(*****
*****
**                                     **
**           PROGRAMA PRINCIPAL           **
**                                     **
*****
*****

```

```

begin
  i := 1;
  continua := 's';
  while continua <> "n" do
    begin
      captura( blente, btiplent, ciclope, desviacion, distojo, facom, fdotras,
        fvecina, numcoord, vcoord);
      inicializacion(distojo, mderd, mderi, mdistancias, vfac, vtrld, vtrli);
      if numcoord > 0 then
        begin
          for plano := 1 to numneur do
            begin
              existe := false;
              cooplan(numcoord, plano, vcoord, existe, plncoord, plvcoord);
              if existe then
                begin
                  ojoizq(blente, btiplent, ciclope,
                    converge, desviacion, distojo,
                    facom, fdotras, fvecina, mderi[plano],
                    mdistancias, plncoord, plvcoord, vfac[plano],
                    vtrld[plano] , vtrli[plano]);

```

## Listado del Programa de Simulacion

```
ojodere(blente, btiplent, converge,
desviacion, distojo,
facom, fdotras, fvecina, mderd[plano],
mdistancias, plncoord, plvcoord, vfac[plano],
vtrid[plano], vtrli[plano]);
msuma(mderi[plano], mderd[plano]);
writeln(' ':10, 'PARA EL PLANO ', plano:3);
writeln(' ':8, '*****la matriz final es*****');
graderiv(mderi[plano]);
puntos(mderi[plano], plano);
end
end;
writeln;writeln;
end
else
writeln('**ningun par de coordenadas binoculares**');
writeln('otra simulacion? s/n');
continua := 'n';
read(continua);
while continua = ' ' do
read(continua);
writeln(continua);
end
end.
```

### III. Mecanismos de Graficacion

El proceso de simulación del mecanismo de percepción de profundidad que se propone en la presente tesis, es totalmente numérico. Una forma de poder ver que esta simulación está funcionando, es a través de las matrices que presentan los valores de las diferentes combinaciones posibles de fibras activadas de ambos ojos. Estos valores muestran el proceso de simulación en forma numérica, siendo en ocasiones tedioso seguir la simulación a través de esta representación, por lo cual se propusieron mecanismos gráficos para la mejor y más cómoda visualización del proceso. Antes de implementar cualquier rutina de graficación se analizaron las posibilidades de equipo disponible para lograr resultados gráficos. De las diferentes opciones con que se contaba ( Chromatics, Grinnell, Foovision, Cromemco III ), se optó por la primera, Chromatics, debido a diversas razones :

a) Equipo orientado totalmente a procesos gráficos, con pantalla de alta resolución, y rutinas realizadas en hardware.

b) Comunicación directa con la computadora FOONLY-F2 del IIMAS en la cual se desarrolló el software utilizado en la simulación.

c) Disponibilidad del equipo.

Respecto a las representaciones gráficas que requeríamos encontramos básicamente dos :

a) Representación de orópteros

b) Representación directa de la matriz de derivadas  $(dw(1,j)/dt)$

La primera representación muestra planos de visión en los

cuales se observan ambos ojos y su espacio de proyección.

Cada área de proyección representa al espacio o dominio de un receptor en particular. En el caso actual, este espacio presenta una abertura de 7.5 grados para cada receptor, contándose con 12 receptores por plano de visión, con lo cual se cubre un espacio total por plano de visión de 90 grados.

La región binocular del animal que se desea simular, presenta las intersecciones de los diversos dominios de los receptores de ambos ojos. Cada una de las intersecciones de la región binocular puede ser colocada en una tonalidad de un solo color, siendo la tonalidad ( desde negra a blanca ), el peso que representa esa intersección en particular.

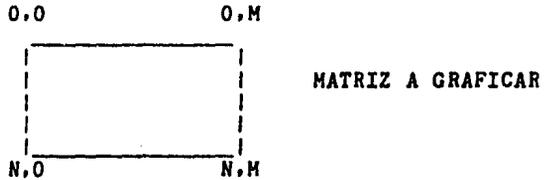
Esta representación es bastante clara, pero limitada por la cantidad disponible de colores. Sólo es posible distinguir las diferencias marcadas entre intersecciones, pero las que presentan un peso semejante no son susceptibles de distinguirse, siendo ésto en ocasiones importante para fines de comprensión.

La representación directa de la matriz de derivadas resuelve el problema de distinción que se presenta en la representación anterior, aunque la visualización de la posición de los puntos en el plano de visión no es tan clara.

La graficación de esta matriz se realizó a través de una malla, donde las alturas relativas representan el peso de cada intersección y dan una idea clara de la diferencia entre la diversas intersecciones.

Para representar la malla, en la cual cada intersección representa un valor en la matriz a través de una altura, se

utilizaron los siguientes elementos : Supongamos que contamos con la siguiente matriz, la cual deseamos graficar



Donde  $N+1$  es el número de renglones de la matriz y  $M+1$  el número de columnas. Una malla puede ser colocada en cualquier lugar de una pantalla de graficación especificando simplemente 4 puntos de control, los cuales determinan el borde de la malla. Estos puntos de control tendrán las siguientes coordenadas:

	X	Y	I	J
P1	x1	y1	0	0
P2	x2	y2	N	0
P3	x3	y3	N	M
P4	x4	y4	0	M

La tabla anterior indica que el punto P1 de coordenada  $(x_1, y_1)$  deberá corresponder a los índices  $(0, 0)$  de la matriz, y en forma semejante para los demás puntos.

Lo anterior establece los límites para todos los índices de la matriz, dicho en otras palabras, para graficar el contenido de las localidades  $I, J$  de la matriz, primero se mapean  $I, J$  a un punto en la pantalla y luego se le da una altura proporcional al contenido de la localidad  $I, J$  (ver figura 5-1).

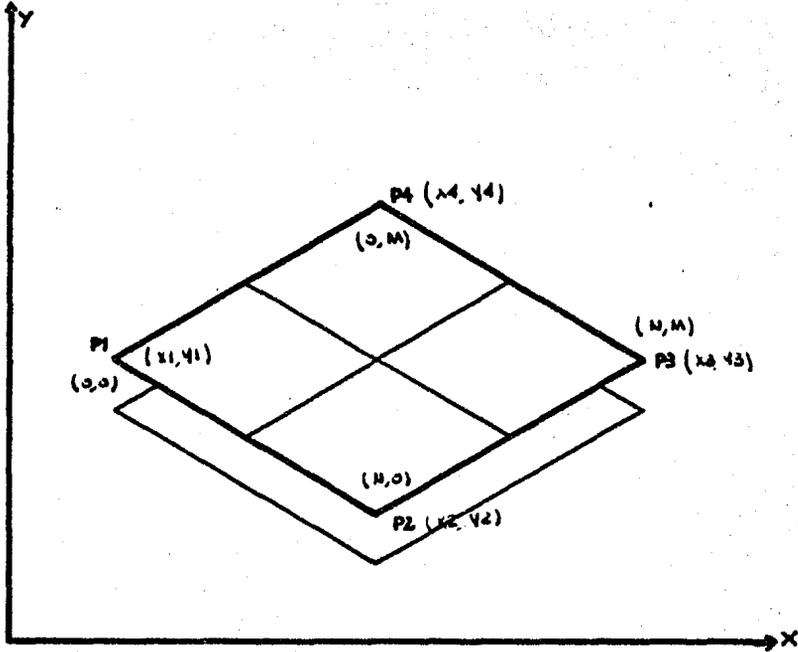


Figura 5-1: Elementos de una Malla de Graficacion

Esto crea la necesidad de encontrar ecuaciones de mapeo de I, J a (X,Y) o sea:

$$F( I ) = X$$

$$F( J ) = Y$$

lo cual se resuelve de la siguiente forma

Establecemos las ecuaciones paramétricas de los bordes de la malla :

Recta de 1 a 2

$$Xa = X1 + (( X2-X1 )/N ) * I$$

$$Ya = Y1 + (( Y2-Y1 )/N ) * I$$

Recta de 2 a 3

$$Xb = X2 + (( X3-X2 )/M ) * J$$

$$Yb = Y2 + (( Y3-Y2 )/M ) * J$$

Recta de 3 a 4

$$Xc = X4 + (( X3-X4 )/N ) * I$$

$$Yc = Y4 + (( Y3-Y4 )/N ) * I$$

Recta de 4 a 1

$$Xd = X1 + (( X4-X1 )/M ) * J$$

$$Yd = Y1 + (( Y4-Y1 )/M ) * J$$

Un punto interior de la malla estará representado por la intersección de dos rectas.

Tomemos las siguientes ecuaciones para localizar la ecuación que mapea un punto dentro de la malla

Recta AC

$$\begin{aligned} \text{Mac} &= (Yc - Ya) / (Xc - Xa) \\ Y &= Ya + \text{Mac} * (X - Xa) \end{aligned}$$

Recta BD

$$\begin{aligned} \text{Mbd} &= (Yd - Yb) / (Xd - Xb) \\ Y &= Yb + \text{Mbd} * (X - Xb) \end{aligned}$$

Resolviendo el sistema de ecuaciones obtenemos las ecuaciones de X,Y que corresponden a cualquier indice I,J de la matriz.

$$\begin{aligned} X &= (Ya - Yb + \text{Mbd} * Xb - \text{Mac} * Xa) / (\text{Mbd} - \text{Mac}) \\ Y &= Ya + \text{Mac} * (X - Xa) \end{aligned}$$

La representación en la pantalla requiere mapear a puntos enteros ( la coordenada de un pixel -elemento básico de la imagen- de es una pareja de números enteros ), por lo cual los valores obtenidos de X y Y deberán ser redondeados a valores enteros, lo anterior involucra cierto error, pero en forma práctica no sera representativo ( +/- 1 pixel de la posición real )

Una vez mapeado el punto, se le da una altura proporcional al valor de la matriz en la localidades I,J.

La graficación de una malla presenta el problema de interpolación de valores. El caso particular que manejamos en la simulación es de una matriz de 12x12, que al presentarlo sería una malla muy abierta y con cortes abruptos, por lo que se propuso una etapa de ampliación de la malla, lo que hizo necesario un proceso de interpolación.

Los pasos anteriores deberían mantener inalterable la

## Mecanismos de Graficacion

información que se pretendía desplegar, así como mostrar una suavización en la malla.

Además se considero el problema de la visibilidad de los elementos de la malla.

Los elementos utilizados para la graficación de la malla fueron los siguientes:

MD, el cual representa a la matriz de derivadas.

MG, que representa una matriz con valores interpolados, su dimensión final es de 45 por 45.

Se realizaron dos tipos de interpolaciones:

- a) De columnas
- b) De renglones

La matriz original de 12 x 12 (m x n) debía tener entonces una cantidad de 45 - n columnas interpoladas y 45 - m renglones interpolados.

Para la interpolación de columnas se utilizó la influencia de cada elemento de MD sobre un punto particular de MG. De esta manera, los elementos cercanos tendrán más influencia que los lejanos y siempre en proporción a su peso (valor en la entrada MD).

La ecuación de interpolación utilizada, tomando las consideraciones anteriores fué la siguiente:

$$MG(i, j) = \sum_{l=0}^{\# \text{ elem de MD} - 1} MG(i, j) + MD(D, l + 1) \times e^{-\text{aten} \times |l \times 4 + 1 - j|}$$

## Mecanismos de Graficacion

Donde las condiciones iniciales son  $MG(i, j) = 0$ ;  $\alpha$  es una constante que indica cuanto queremos restarle de influencia de un punto a otro de MG.  $D$  toma valores entre 1 y 12.

En la ecuación anterior  $i$  representa las columnas (iniciando en 0) de un renglón en MD y  $j$  representa una columna (iniciando en 1) de un renglón de MG.

Se observa que mientras  $i$  y  $j$  estén mas separadas, la influencia del punto  $MD(D, i+1)$  será mínima, y el peso del punto  $MG(i, j)$  no variará significativamente.

Para la graficación en tres dimensiones se recurrió a un paquete de graficación desarrollado en el IIMAS formado por un editor de cuerpos (EDY83) y un generador de imágenes tridimensionales (DESPL75), el cual realiza una transformación de las coordenadas espaciales del objeto definido en el editor, a coordenadas normalizadas (ampliación de coordenadas  $x, y, z$  a coordenadas  $x', y', z', 1$ ) [10].

Los resultados obtenidos, fueron bastante satisfactorios, ya que muestran graficas de buena calidad que permiten en forma clara comprender el modelo.

## BIBLIOGRAFIA

- [1] Romer Alfred S. y Parsons Thomas S.  
Anatomia Comparada.  
Quinta Edición, Ed. Interamericana, Mexico, 1983.
- [2] Collet, T.  
Stereopsis in Toads.  
Nature, 1977.
- [3] Dev, P.  
Perception of Depth Surfaces in Random-Dot Stereograms: A neural model.  
Int. J. Man Machine Studies, 1975.
- [4] House, D.  
The Frog/Toad Depth Perception System- A Cooperative/Competitive Model.  
COINS Technical Report, 1982.
- [5] Ingle, D.  
The Amphibian Visual System: A multidisciplinary approach.  
Academic press, 1976.
- [6] Lara, R., Carmona, M., Daza, F. y Cruz, A.  
A Global Model of the Neural Mechanisms Responsible for Visuomotor Coordination in Toads.  
J.Theoret. Biol., 1984.
- [7] Marr, D. y Poggio, T.  
Cooperative Computation of Stereo Disparity.  
Science, 1976.
- [8] Ibarra J., Pineda J., Rodriguez J.  
4a. Escuela de Invierno del A.M.C.A. Curso de Robotica.  
A.M.C.A., 1984.
- [9] Rossel, J.  
Stereopsis in an Insect.  
Nature, 1984.
- [10] Newman, W.M., y R.F. Sproull.  
Principles of Interactive Computer Graphics.  
2a ed., Mc Graw Hill, New York, 1976.
- [11] Wallace, R.  
Peering in Locust.  
J. Exp. Biol, 1959.

Lista de Figuras

Figura 2-1:	Ojo de una rana. Romer. (1983)	7
Figura 2-2:	Conexiones dentro de la retina: b, células bipolares unidas a conos únicos; bas, bastones; c, conos; cb, células bipolares unidas a varios bastones; a y h, células horizontales y amacrinas; g, células ganglionares Romer (1983)	9
Figura 2-3:	Esquema del quiasma óptico Romer (1983)	11
Figura 2-4:	Ojo de rana (a) y de humano (b) (Lara, 1981)	12
Figura 2-5:	(a) y (b). Campos visuales monocular (blanco) y binocular (punteado) de una rana. Pipiens (c) y B. Marinus (d). (partes binoculares oscuras, parte fuera del campo visual blanca). (De Fite, 1973)	13
Figura 2-6:	Anatomía y fisiología de las células retinales (De Shepherd 1970)	13
Figura 2-7:	Modelo Simplificado de Amari-Arbib	20
Figura 2-8:	Efecto neto de un estímulo exterior en el modelo de Amari-Arbib	22
Figura 2-9:	Modelo de Percepción de Profundidad de Donald H. House	23
Figura 2-10:	Modelo de Percepción de Profundidad de Marr y Poggio	25
Figura 3-1:	Modelo de percepción de profundidad por codificación de la frecuencia neuronal	32
Figura 3-2:	Efecto de los lentes sobre la imagen percibida	38
Figura 3-3:	Esquema de la división en regiones del espacio tridimensional	45
Figura 3-4:	Región numero 6 en el espacio tridimensional. Fotoreceptores 1 del ojo izquierdo y 6 del derecho activos	47
Figura 3-5:	Puntos en el espacio real	47
Figura 3-6:	Aspecto de un sólo cuerpo desde varios puntos de referencia	50
Figura 3-7:	Aspecto de un sólo cuerpo desde varios puntos de referencia (...continuación)	50
Figura 3-8:	Aspecto de los dos cuerpos desde varios puntos de referencia	50
Figura 3-9:	Aspecto de tres cuerpos desde varios puntos de referencia	50
Figura 3-10:	Aspecto de cinco cuerpos desde varios puntos de referencia	50
Figura 3-11:	Aspecto de cinco cuerpos desde varios puntos de referencia (...continuación)	50
Figura 3-12:	Representación usando un oróptero	58
Figura 3-13:	Orópteros para parejas activas (a) y para otras neuronas del ojo derecho (b)	78
Figura 3-14:	Orópteros para acción de acomodación (a) y para la primera iteración del ojo derecho (b)	78
Figura 3-15:	Orópteros para la segunda iteración del ojo derecho (a) y para el estado final (b)	78
Figura 3-16:	Orópteros para parejas activas (a) y para el estado final (b) usando lentes de alejamiento aparente	78
Figura 3-17:	Oróptero para la etapa de parejas activas (a)	78
Figura 3-18:	Mallas gráficas que muestran el contraste entre una iteración y el resultado final de un experimento (a) y la etapa de parejas activas	84

	con fantasmas (b)	
Figura 3-19:	Mallas de una etapa de iteración para el ojo derecho (a) y la etapa final (b)	84
Figura 3-20:	Mallas para parejas activas (a) e iteración 1 del ojo izquierdo (b) sin considerar acomodación	84
Figura 3-21:	Mallas para la iteración 1 del ojo derecho sin acomodación (a) y para un experimento monocular sin lentes, (iteración ojo derecho) (b)	84
Figura 3-22:	Mallas para el estado final del experimento monocular sin lentes (a) y para la primera iteración del experimento monocular con lentes de alejamiento (b)	89
Figura 3-23:	Mallas para el estado final de un experimento monocular con lentes de alejamiento (a) y para la iteración 1 de un experimento monocular con lentes de acercamiento (b)	89
Figura 3-24:	Malla para el estado final de un experimento monocular con lentes de acercamiento (a) y para el estado final de un experimento monocular sin acomodación (b)	89
Figura 4-1:	Gráficas de comportamiento del algoritmo ante la variación del peso del factor de acomodación	117
Figura 4-2:	Gráficas de comportamiento del algoritmo ante la variación del peso del factor de acción de otras neuronas	117
Figura 4-3:	Gráficas de comportamiento del algoritmo ante la variación del peso del factor de acción de vecinas	117
Figura 4-4:	Espacio Tridimensional de Convergencia	117
Figura 5-1:	Elementos de una Malla de Graficación	159