CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA

POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN

Apareamiento denso de un par estereoscópico usando un enfoque de vida artificial

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta:

Cynthia Beatriz Pérez Castro

Ensenada, Baja California, México. Agosto del 2004.

BIBLIOTECA
CICESE

TESIS DEFENDIDA POR

Cynthia Beatriz Pérez Castro

Y aprobada por el siguiente comité:

Dr. Gustavo Olague Caballero

Director del Comité

M.C. José Duglas Frez Cárdenas

Miembro del Comité

M.C. José Luis Briseño

Miembro del Comité

Dr. Rafael de Jesús Kelly Martínez

Miembro del Comité

Dr. Carlos Alberto Brizuela Rodríguez

Coordinador del programa en Ciencias de la Computación

Dr. Federico Graef Ziehl

Director de Estudios de Posgrado

9 de Agosto del 2004

BIBLIOTECA
CICESE

CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN
SUPERIOR DE ENSENADA

POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN

Apareamiento denso de un par estereoscópico usando un enfoque de vida artificial

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN CIENCIAS

Presenta:

Cynthia Beatriz Pérez Castro

Ensenada, Baja California, México. Agosto del 2004.

RESUMEN de la tesis de **Cynthia Beatriz Pérez Castro**, presentada como requisito parcial para obtener el grado de MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, B. C. Agosto del 2004.

Apareamiento denso de un par estereoscópico usando un enfoque de vida artificial

Resumen aprobado por:

Dr. Gustavo Olague Caballero

Director de Tesis

Esta tesis describe un nuevo algoritmo para el problema de correspondencia, llamado "*algoritmo de infección*". El problema de correspondencia presenta muchos aspectos complejos que hace el proceso muy difícil. La correspondencia es esencial en la reconstrucción 3D de escenas en visión por computadora. Nuestra meta es corresponder dos imágenes (izquierda y derecha) y aplicar una nueva técnica de vida artificial que permite la generación de una imagen desde un punto de vista distinto, donde la fotografía no fué adquirida por la cámara. Basándonos en la información de la imagen y la información de profundidad 3D, el algoritmo decide cuáles partes de la escena son visibles para reproyectar la nueva imagen mientras va ahorrando un número determinado de operaciones, y con ello, tiempo en el proceso. Este algoritmo emplea un conjunto de reglas distribuidas por toda la imagen, estas reglas permiten "*adivinar*" los pixeles que corresponden en la imagen derecha.

Palabras clave: correspondencia, vision estereo, vida artificial.

ABSTRACT of the thesis presented by **Cynthia Beatriz Pérez Castro**, as a partial requirement to obtain the MASTER SCIENCE degree in COMPUTER SCIENCES. Ensenada, B. C. august 2004.

Dense Stereo Matching Using An Artificial Life Approach

Abstract approved by:

Dr. Gustavo Olague Caballero

Thesis director

This thesis introduces a new algorithm for dense stereo matching, we called "*the infection algorithm*". The correspondence problem presents many complex aspects which turns very difficult the process of image correspondence which is basic for 3D scene reconstruction in computer vision. Our aim is to match the contents of two images (left and right) and apply a new artificial life 3D rendering technique, which allows the generation of images for viewer positions from which no photograph was taken. On the basis of image contents and 3D depth information, the new rendering algorithm decides which parts of the scene are visible in order to render a new image from a different viewpoint, while saving on the number of computations and time in the process. The algorithm employs a distributed set of rules over the whole image, these rules allow "*guess*" the corresponding pixels in the right image.

Keywords: stereo matching, stereo vision, artificial life.

A mis padres

Ing. Rodolfo Pérez Rodríguez
y
Ruth Beatriz Castro de Pérez

A mi hermano

Humberto Pérez Castro.

Agradecimientos

Principalmente le doy gracias a Dios por haberme dado la oportunidad de vivir y tener a mi familia conmigo.

A mis padres y hermano por todo el amor, entrega, valor, honestidad y coraje que hicieron posible formar en mí, un carácter mas fuerte hacia los retos que presenta la vida.

Al Dr. Gustavo Olague, a quien considero mi padre de la investigación porque como a una hija, me ha enseñado el verdadero camino de la ciencia; por su paciencia, preocupación, sinceridad, consejos y formación hacia mi persona.

A los miembros de mi comité, M.C. José Frez Cárdenas, Dr. Rafael Kelly Martínez y M.C. José Luis Briseño quienes en todo momento me brindaron su ayuda. Gracias por los consejos, disponibilidad y aportaciones que me permitieron aprender y mejorar el trabajo.

A Enrique Dunn Rivera, a quien considero mi hermano de investigación. Gracias Enrique por ayudarme siempre y estar al pendiente en todo momento. No sé qué hubiera sido de mí en la maestría sino hubieras estado; pero sobre todo, cómo olvidar las aventuras en París, la nevada con agua, la tormenta, tu laptop, mi salto increíble en el metro, la llegada congelada al Inria, la corrida que hicimos para alcanzar el camión que nos dejó en la noche, entre muchas más... por lo menos sé que te divertiste.

Al Dr. Francisco Fernández por su ayuda, asesoría y entusiasmo en nuestro trabajo. Las reuniones en España fueron clave importante en el desarrollo de la tesis.

A la Dra. Évelyne Luttony al equipo del proyecto Fractales del INRIA por su amabilidad y comentarios acerca del trabajo cuando estuve en Francia.

A mis amigas Jazmín Romero González y Diana Guadalupe Escalante Salazar, por los momentos que hicieron posible que los lazos de amistad se solidificaran y llegaran a ser en mi vida como dos grandes hermanas y amigas. Gracias por levantarme en cada caída y luchar por mi cuando mis fuerzas se acababan.

A mis amigos de toda la vida, Roxana, Luis Adrián, Karina, Macario, Sarbia, Erika y Elvia.

A todos mis amigos de cicese a quienes llegué a apreciar en tan poco tiempo, Luis Carlos, Carlos Gómez, Dante, Sussy, Mike, Elisa, Julio, Adrián Macías, Christian, Luis Herrada, Angel, Victor, Rodrigo, Omar, Alfredo, Everardo, Isabel, Marcel, David, Bob, Adrián, César, Pául y Caneck.

Al Consejo Nacional de Ciencia y Tecnología y al proyecto LAFMI por el apoyo económico, por hacer posible una de mis metas profesionales y por la inigualable experiencia que tuve en la estancia en Francia y España.

Ensenada, México
9 de Agosto del 2004.

Cynthia Beatriz Pérez Castro

Tabla de Contenido

Capítulo — Página

Resumen ii
Abstract iii
Agradecimientos v
Lista de Figuras x
Lista de Tablas xvi
Tabla de símbolos xvii
I Introducción 1
I.1 Visión Estéreo 2
I.2 Objetivos de la Tesis 8
I.2.1 Objetivo General 8
I.2.2 Objetivos Específicos 8
I.3 Importancia de la Investigación 9
I.4 Aplicaciones 10
I.5 Organización de la Tesis 11
II Antecedentes 13
II.1 Métodos de Correspondencia Local 14
II.2 Métodos de Correspondencia Global 19
II.3 Conclusiones 21
III Conceptualización Geométrica del Problema 22
III.1 Introducción 22
III.2 Geometría Proyectiva 23
III.3 Geometría Afín 24
III.4 Geometría Euclidiana 25
III.5 Proceso de adquisición de la imagen 26
III.5.1 Modelo de Cámara Obscura 27
III.5.2 Transformación del Sistema Coordenado del Mundo al Sistema Coordenado de la Cámara (Mundo-Cámara) 30
III.5.3 Transformación del Sistema Coordenado de la Cámara al Sistema Coordenado de la Imagen (Cámara-Imagen) 31
III.5.4 Calibración de una cámara 36
III.5.5 Geometría Epipolar 41
III.6 Restricciones Geométricas y Físicas 47
III.6.1 Epipolar 47
III.6.2 Orientación 53
III.6.3 Orden 53
III.6.4 Unicidad 55
III.6.5 Continuidad de la Figura 57

Tabla de Contenido (Continuación)

Capítulo Página

III.6.6 Similaridad 58
III.6.7 Relajación 58
III.6.8 Suavidad 58
III.7 Conclusiones 59
IV El Algoritmo de Infección: Un Nuevo Enfoque 61
IV.1 Conceptualización sobre Vida Artificial 61
IV.2 Herramientas 62
IV.3 Autómatas Celulares 64
IV.3.1 Tipos de Vecindario 65
IV.4 Complejidad de las reglas 67
IV.4.1 Caso más sencillo: Autómata 1-D 68
IV.4.2 Caso del Autómata 2-D 69
IV.4.3 Caso del Autómata 2-D: Nuestro Caso 71
IV.5 La Correlación como Función Objetivo Local 73
IV.6 Epidemia artificial para el problema de correspondencia estéreo 75
IV.7 Descripción del algoritmo de infección 77
IV.8 Formalización del Algoritmo 83
IV.8.1 Descripción y Notación General de una Regla 86
IV.9 Conclusiones 90
V Experimentación 92
V.1 Reconstrucción 3D de la imagen 92
V.2 Simulación de las reglas 93
V.2.1 Objetivo de la Simulación 101
V.3 Implementación del algoritmo 101
V.3.1 Experimento 1: Caso del 47% de ahorro 104
V.3.2 Experimento 2: Caso del 0% de ahorro 122
V.3.3 Experimento 3: Caso del 70% de ahorro 126
V.3.4 Experimento 4: Caso del 78% de ahorro 129
V.3.5 Experimento 5: Caso del 99% de ahorro 133
V.3.6 Experimento 6 137
V.4 Conclusiones 139
VI Conclusiones y Trabajo Futuro 140
Bibliografía 143
A Especificaciones del funcionamiento del sistema 150
A.1 Módulo para la Adquisición de Fotos 150
A.2 Módulo para la Calibración de la cámara izquierda 150
A.3 Módulo para la Calibración de la cámara derecha 151
A.4 Módulo para la Generación de puntos 3D 152

Tabla de Contenido (Continuación)

Capítulo Página

A.5 Uso del Sistema 152

B Visualización de las reglas 155

B.1 Caso del 0% 156

B.2 Caso del 78% 157

B.3 Caso del 99% 160

FECHA DE INGRESO
NOV 16 2004
BIBLIOTECA CIECSE

Lista de Figuras

Figura Página

1 a) y b) Imágenes de entrada. La imagen c) muestra la visualización tridimensional de la reconstrucción que se realizó a partir de las imágenes de entrada. 2
2 Problema de correspondencia 3
3 Baja Textura. Los pixeles de la región que aparece en el recuadro tienen el mismo valor de intensidad. 5
4 Campo de Visión. La parte final de la imagen izquierda no se encuentra en la imagen derecha y la parte inicial de la imagen derecha no se encuentra en la imagen izquierda debido al campo de visión de la cámara. 5
5 Oclusión. El archivero que se encuentra en la imagen derecha no aparece en la imagen izquierda debido al movimiento de la cámara. 6
6 Patrones repetitivos. Los cuadros negros de la mira de calibración ejemplifican este problema inherente de la correspondencia. 7
7 Robot Staubli de 6 grados de libertad que se encuentra en el laboratorio de EvoVisión de CICESE. 8
8 *Modelo Geométrico de una Cámara.* La figura muestra la geometría de un modelo de cámara obscura. C es el centro de proyección y p el punto principal. El plano de la imagen está situado al frente del centro de la cámara. 28
9 *Geometría Epipolar.* Cualquier plano π que intersecte el plano de la imagen en sus correspondientes líneas epipolares l y l' es un plano epipolar. La línea base intersecta cada plano de la imagen en los epipolos e y e'. Todas las líneas epipolares se intersectan en el epipolo. 42
10 *línea Epipolar.* Se muestran 5 líneas epipolares en la imagen derecha para 5 puntos de la imagen izquierda. Note que el epipolo se encuentra fuera de la imagen. 43
11 *Correspondencia Geométrica.* Las dos cámaras son indicadas por los centros de proyección C y C' y los planos de las dos imágenes. Los centros de proyección, el punto M, y sus correspondientes puntos proyectados en las dos imágenes m y m' están sobre un mismo plano π. 45
12 *Restricción Epipolar.* Esta restricción asegura que un punto 3D que es proyectado en ambos planos de la imagen como m y m' se localizan en una línea epipolar en cada imagen. El punto m' está localizado en la línea epipolar l de la imagen izquierda y viceversa. 47
13 *Restricción de Orientación.* Un punto del contorno está definido por su posición en la imagen, y la dirección de la tangente al contorno en este punto. 53

Lista de Figuras (Continuación)

Figura Página

14 *Restricción de Orden.* M_1 y M_3 se proyectan en el mismo orden en las dos imágenes mientras que el orden de las proyecciones M_1 y M_2 es inverso, dado que el punto M_2 se encuentra dentro de la zona prohibida (zona interior) 54

15 *Restricción de Orden.* Cuando dos puntos se encuentran sobre la superficie de un objeto que está poco inclinado como el caso del objeto 1, el orden de sus proyecciones es el mismo en las dos imágenes. Si el objeto se encuentra muy inclinado como el objeto 2, las proyecciones no se ven simultáneamente en las dos imágenes si el objeto es opaco, pero si el objeto es transparente, el orden se invierte. 55

16 *Restricción de unicidad.* *a*) Para el punto m_1 existe solo un punto $m_1^{'}$ que corresponde en la otra imagen. En este caso, la restricción de unicidad se cumple. *b*) La restricción de unicidad no se respeta cuando dos puntos se proyectan en un mismo punto en una imagen y en dos puntos diferentes en la otra imagen. En este caso, M_1 y M_2 se proyectan en la imagen izquierda en un único punto m mientras que en la imagen derecha se proyecta en los dos puntos que corresponden $m_1^{'}$ y $m_2^{'}$. 56

17 *Restricción de Continuidad de la figura.* Los puntos de la imagen izquierda no son bien correspondidos en la imagen derecha debido a la existencia de contornos parecidos en una sola región. 57

18 Autómata unidimensional. La estructura general de un autómata unidimensional consiste de una rejilla de tamaño $1 \times n$. 66

19 Autómata bidimensional. La estructura general de un autómata bidimensional consiste de una rejilla de tamaño $2 \times n$. 66

20 *Estructuras de Vecindarios.* *a*) Vecindario de tres células para un autómata unidimensional. Para un autómata bidimensional: *b*) Vecindario de Von Neumann con 4 células, *c*) Vecindario de Moore con 9 células y *d*) Vecindario de Moore Extendido con 25 células. 67

21 *Vecindario usado en esta tesis.* *a*) Las nueve celdas centrales forman un vecindario de Moore, *b*) El resto de celdas se alinean desde la célula central para formar el vecindario completo. *c*) Estructura del vecindario usado en la tesis con un total de 25 vecinos. 67

22 Autómata unidimensional. 68

23 $2^3 = 8$ Configuraciones distintas de reglas. 69

24 $2^{2^3} = 256$ Mapeos distintos para 8 configuraciones. 70

25 Configuración de las reglas utilizando un vecindario de Moore con estados binarios ($2^9 = 512$). 70

Lista de Figuras (Continuación)

Figura Página

26 Ejemplo de los mapeos de las reglas con $k = 9$ y $R = 2$. Se tienen 512 configuraciones y $2^{2^9} = 1.3408 \times 10^{154}$ mapeos distintos para esas configuraciones. 71
27 Estados utilizados en el autómata 72
28 Coeficiente de correlación del punto m de la imagen izquierda con el punto m' de la imagen derecha. Esquema que representa el método para un algoritmo de búsqueda exhaustiva utilizando la correlación como función objetivo. 74
29 Relación entre cada una de las rejillas usadas en el algoritmo de infección. Los valores numéricos representan el valor del nivel de intensidad de una región de la imagen. El cuadro azul representa el estado *enfermo* y el color negro el estado *sano* 78
30 Grafo Global que representa la transición de las 14 reglas. 85
31 Subgrafo A, B, C de la regla 2 (R_2), regla 1 (R_1) y regla 3 (R_3). 86
32 Relación de la posición de 8 estados centrales 89
33 Relación de la posición de los estados que se encuentran en la parte exterior del vecindario 89
34 Reconstrucción 3D de todos los puntos de la imagen. 93
35 Proyección de los puntos a partir de un par estéreo. 94
36 Estructura inicial de las reglas alrededor del núcleo de infección. 95
37 Propagación de las reglas de transición en el ciclo 9. 96
38 Propagación de las reglas de transición en el ciclo 14. 97
39 Propagación de las reglas de transición en el ciclo 19. 97
40 Imagen final de la propagación de las reglas en el ciclo 61. 98
41 Par estéreo utilizado como datos de entrada en cada uno de los experimentos. 104
42 Experimento 47%. Resultado final del algoritmo de infección aplicando un filtro de mediana. 105
43 Cuatro etapas inciales de la evolución del algoritmo de infección. 105
44 Etapas Finales del algoritmo de infección y la imagen final del algoritmo de búsqueda exhaustiva. 106
45 Imagen que resulta después de aplicar el algoritmo de infección. 107
46 Transición de la Regla 1. El estado de la célula central no cambia dado que el pixel ya fue evaluado. 108
47 Transición de la Regla 2. El estado de la célula central no cambia, debido a que no hay información en el vecindario interno. 108
48 Transición de la Regla 3, 4, 5 y 6. Estas reglas en conjunto forman la estructura inicial de la propagación. 109

Lista de Figuras (Continuación)

Figura Página

49 Transición de la Regla 7. Esta regla indica que se requieren mínimamente tres estados EXPLOTADOS (*Enfermos*)en el vecindario interno y tres estados AUTOMÁTICOS (*Inmunes*) en el vecindario externo. 110
50 Transición de la Regla 8. Esta regla indica que se requieren mínimamente tres estados EXPLOTADOS (*Enfermos*)en el vecindario interno y tres estados AUTOMÁTICOS (*Inmunes*) en todo el vecindario. 110
51 Transición de la Regla 9 y 10. Estas reglas evitan la propagación lineal de la asignación automática. 111
52 Transición de la Regla 11. Se requiere mínimamente tres estados EXPLOTADOS (*Enfermos*) para dejar marcada la célula central como estado PROPUESTO (*Infectado*). 112
53 Transición de la Regla 12. Se requiere mínimamente tres estados PROPUESTOS (*Infectados*) para que el estado de la célula central cambie a un estado EXPLOTADO (*Enfermo*). 112
54 Regla 13. Se requiere mínimamente tres estados EXPLOTADOS (*Enfermos*) para que la célula central cambie a un estado AUTOMÁTICO (*Inmune*). 113
55 Regla 14. Se requiere mínimamente tres estados PROPUESTOS (*Infectados*) para que la célula central cambie a un estado EXPLOTADO (*Enfermo*). 114
56 Experimento 47%. Gráfica de resultados de la evolución de cada uno de los estados. 118
57 La región del recuadro se mejora en relación con el resultado del método exhaustivo. 120
58 La región lateral a la mira de calibración se mejora ligeramente. 121
59 Se aprecia la comparación entre una región de la imagen original y una región de la imagen producida por el algoritmo cuando existe buena textura en el par estéreo. 121
60 Experimento 0%. Imagen que resulta después de aplicar el algoritmo de infección a la cual se le aplicó un filtro de mediana. 122
61 Imagen que resulta de aplicar del algoritmo de búsqueda exhaustiva a la cual se le aplicó un filtro de mediana. 123
62 Experimento 0%. Gráfica de resultados de la evolución de cada uno de los estados. 125
63 Experimento 70%. Imagen resultante del algoritmo de infección a la cual se le aplicó un filtro de mediana. 126
64 Experimento 70%. Gráfica de resultados de la evolución de cada uno de los estados. 128

Lista de Figuras (Continuación)

Figura Página

65 Experimento 78%. Imagen resultante del algoritmo de infección a la cual se le aplicó un filtro de mediana. 129
66 Experimento 78%. Gráfica de resultados de la evolución de cada uno de los estados. 132
67 Experimento 99%. Imagen resultante del algoritmo de infección a la cual se le aplicó un filtro de mediana. 133
68 Experimento 99%. Gráfica de resultados de la evolución de cada uno de los estados. 136
69 Imágenes utilizadas como datos de entrada para el algoritmo de infección en este experimento. 137
70 Resultado del algoritmo de infección utilizando las reglas del 47% para este experimento. 138
71 Resultado del algoritmo de infección utilizando las reglas del 99% para este experimento. 138
72 Regla 11 para el caso del 0% de ahorro en operaciones. 156
73 Grafo que representa las transiciones de todas las reglas para el caso del 0% de ahorro. 156
74 Regla 7 para 78%. El cambio surgió en el estado final del pixel. 157
75 Regla 8 para 78%. Se cambió el estado final del pixel a un estado PROPUESTO (*Infectado*). 157
76 Regla 9 para 78%. De igual forma, el cambio fué en el estado final del pixel. 158
77 Regla 10 para 78%. Se cambió el estado final del pixel a un estado PROPUESTO (*Infectado*). 158
78 Regla 11 para 78%. Se disminuyó el número de vecinos EXPLOTADOS (*Enfermos*). 159
79 Regla 13 para 78%. Se disminuyó el número de vecinos PROPUESTOS (*Infectados*). 159
80 Grafo que representa las transiciones de cada regla cuando se ahorra un 78% de cálculos. 159
81 Regla 7 para 99%. El cambio realizado en la regla 7 fue en el estado final de la transición. 160
82 Regla 8 para 99%. El estado final cambió al estado PROPUESTO (*Infectado*). 161
83 Regla 9 para 99%. Se cambió el estado final a estado PROPUESTO (*Infectado*). 161

Lista de Figuras (Continuación)

Figura Página

84 Regla 10 para 99%. Esta regla cambia al igual que las anteriores en el estado final del pixel. 161
85 Regla 11 para 99%. Se disminuyó el número de vecinos EXPLOTADOS (*Enfermos*). 162
86 Regla 12 para 99%. Se cambió el estado final del pixel central. 162
87 Regla 13 para 99%. Se disminuyó el número de vecinos PROPUESTOS (*Infectados*). 162
88 Grafo que muestra las transiciones de cada una de las reglas para el caso de 99% de ahorro. 163

Lista de Tablas

Tabla		Página
I	Relación de los algoritmos que usan un método basado en áreas	16
II	Estructura de la Tabla de reglas	88
III	Tabla de datos de la Simulación de Reglas	95
IV	Ejecución 1. Tabla de datos.	99
V	Ejecución 2. Tabla de datos.	99
VI	Ejecución 3. Tabla de datos.	100
VII	Ejecución 4. Tabla de datos.	100
VIII	Resumen de las Tablas III , IV , V , VI y VII	100
IX	Porcentaje de Ahorro Promedio	101
X	Tabla de reglas que genera un 47% de ahorro en operaciones.	117
XI	Experimento 47%. Datos generados por el algoritmo de infección	119
XII	Tabla de reglas que genera un 0% de ahorro en operaciones	123
XIII	Experimento del 0%. Datos generados por el algoritmo de infección.	124
XIV	Tabla de reglas que genera un 70% de ahorro en operaciones	127
XV	Experimento 70%. Datos generados por el algoritmo de infección	128
XVI	Tabla de reglas que genera un 78% de ahorro en operaciones	130
XVII	Experimento 78%. Datos generados por el algoritmo de infección	131
XVIII	Tabla de reglas que genera un 99% de ahorro en operaciones	134
XIX	Experimento 99%. Datos generados por el algoritmo de infección	135

Tabla de símbolos

$\mathbf{K}$	Conjunto
n	Escalar
$\mathbf{a}$	Vector
$\mathbf{f}(x)$	Vector de funciones

Capíulo I

Introducción

Los seres humanos perciben el mundo de manera tridimensional a través de un sistema de visión natural. La percepción 3D es natural para la humanidad ya que se encuentra inmersa en espacios tridimensionales. Sin embargo, lograr simular la percepción en un robot es sumamente complicado. Las escenas reales que se perciben en un espacio 3D pueden ser vistas en un dispositivo digital como la computadora, la cual puede procesar la información visual percibida de la escena haciendo uso de complejos modelos matemáticos.

La visión por computadora es un campo relativamente nuevo y de rápido crecimiento. Los primeros experimentos fueron conducidos alrededor de los años 50's del siglo pasado y muchos de los conceptos fueron desarrollados alrededor de 1970, (Ballard y Brown, 1982). Con el rápido crecimiento que ha tenido, mantiene una estrecha relación con disciplinas como inteligencia artificial, gráficas por computadora, procesamiento de imágenes, fotogrametría, entre otras.

La visión por computadora estudia la visión humana o animal la cual modela matemáticamente apoyándose en la geometría de la escena. La visión depende del entendimiento de cámaras y del proceso físico de la formación de la imagen (Forsyth y Ponce, 2002) para obtener información de los valores de los pixeles, realizar cálculos o inferencias, poder combinar información de una secuencia de imágenes y reconocer objetos utilizando información geométrica.

I.1 Visión Estéreo

La visión estéreo es una de las áreas de investigación de la visión por computadora y estudia la extracción de la estructura tridimensional de una escena usando información de dos o más imágenes reales tomadas desde distintos puntos de vista, ver Figura 1. Los puntos tridimensionales de la Figura 1c fueron visualizados en el Lenguaje de Modelado de Realidad Virtual, VRML[1]. El estudio de la Visión Estéreo se basa en el hecho de que un punto físico es proyectado a través de la cámara en las imágenes.

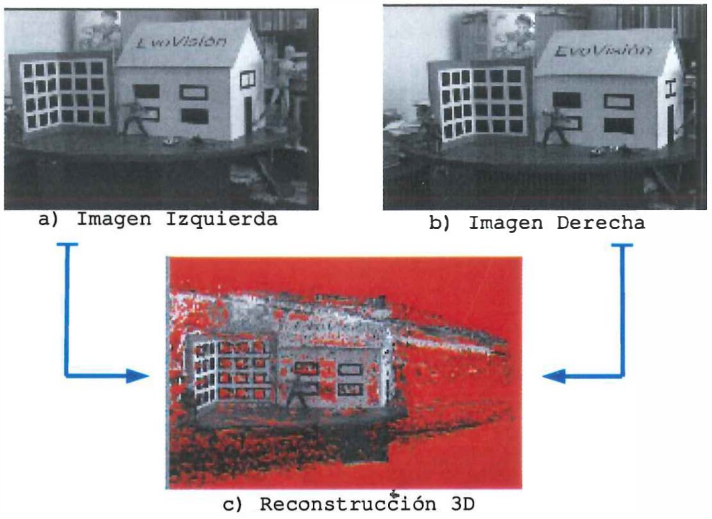


Figura 1: a) y b) Imágenes de entrada. La imagen c) muestra la visualización tridimensional de la reconstrucción que se realizó a partir de las imágenes de entrada.

Los primeros trabajos realizados en esta área fueron alrededor de 1970 a 1980 dentro de la comunidad de Image Understantding (IU) fundada por la Agencia de Proyectos de Investigación Avanzados, ARPA[2], (Barnard y Fischler, 1982). Para los años 90's

[1] Virtual Reality Modeling Language
[2] Advanced Research Proyects Agency

la investigación estéreo ya había madurado; sin embargo, se continuó estudiando la temática y permaneció el interés en diversas comunidades (Brown *et al.*, 2003).

Los principales problemas a resolver en la visión estéreo son: *la calibración, la correspondencia* y *la reconstrucción.* La *calibración* es el proceso que se sigue para obtener información de la orientación y posición de la cámara con respecto a la escena, así como también el centro óptico y la distancia focal. El *problema de correspondencia*[3] consiste en determinar cada uno de los pixeles que pertenecen al mismo punto físico y que son proyectados en las dos imágenes. La Figura 2 ejemplifica el problema de correspondencia natural del humano; los puntos falsos son los puntos que no alcanza a ver el ojo porque existe un objeto enfrente. La *reconstrucción*, se encarga de recuperar la estructura tridimensional de la escena a partir de la información 2D de las imágenes y la información geométrica de la cámara.

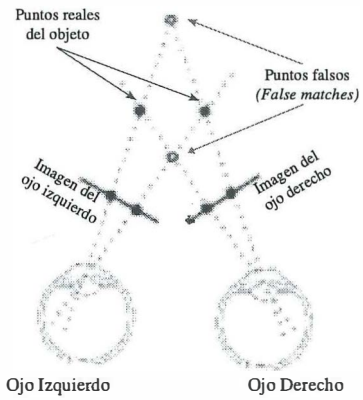


Figura 2: Problema de correspondencia

El problema de correspondencia continúa siendo uno de los principales focos de atención en la comunidad de visión ya que en la actualidad no existe la solución general

[3] *En inglés, Stereo Matching*

al problema. Para lograr una reconstrucción tridimensional de una escena, se necesita la correspondencia o apareamiento entre las imágenes y localizar los puntos proyectados de las imágenes en las dos cámaras que corresponden al mismo punto físico en el espacio.

El problema de correspondencia es definido formalmente como sigue:

Definición 1. Problema de Correspondencia:*Para un punto m en el plano-1* [4], *determinar el punto m' en el plano-2* [5] *que le corresponde. El término "corresponder" implica que ambas imágenes contienen el mismo punto físico* M. *Esto es lo que comúnmente se conoce como el problema de correspondencia (Faugeras, 1996). El punto* m *y* m' *tienen tienen coordenadas* (x, y, t) *y* (x', y', t), *mientras que* M *tiene coordenadas* (X, Y, Z, T).

La correspondencia es un problema con ambigüedades inherentes y ha sido considerado como el problema más difícil e importante en visión estéreo (Barnard y Fischler, 1982). La dificultad es debida a varios factores principalmente relacionados con la geometría de las imágenes. A continuación, se muestran las razones principales por la que el problema es difícil de resolver:

1. **Ruido**. Variaciones de luminosidad que son inevitables, imágenes borrosas y el ruido del sensor en la formación de la imagen.

2. **Regiones de baja textura**. Regiones en las cuales el nivel de intesidad de gris es muy similar o igual, ver Figura 3. El problema con las regiones que tienen baja textura es que el nivel de intensidad es muy similar, lo que conduce a decidir cuál es el pixel correcto que le corresponde en la imagen derecha porque el valor de intensidad es el mismo entre los pixeles vecinos. Los valores de intesidad de un pixel en escala de grises varía entre 0 y 255.

[4] *Imagen Izquierda*
[5] *Imagen Derecha*

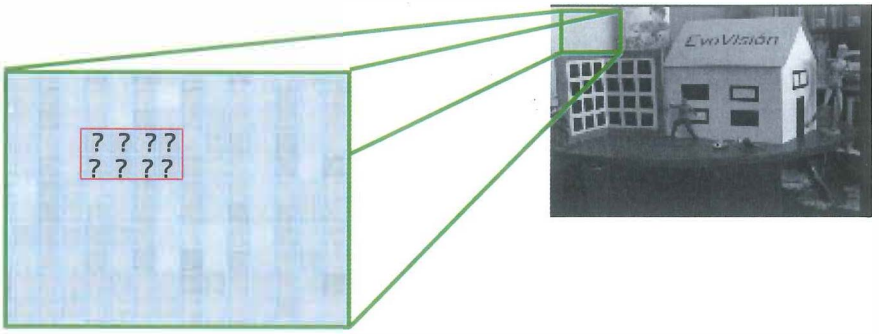


Figura 3: Baja Textura. Los pixeles de la región que aparece en el recuadro tienen el mismo valor de intensidad.

3. **Campo de Visión**. Se refiere al campo de visión que tiene la cámara al tomar la fotografía. Esto hace que en la segunda fotografía exista información nueva que no se logra ver en la fotografía anterior debido a la traslación y viceversa, ver Figura 4.

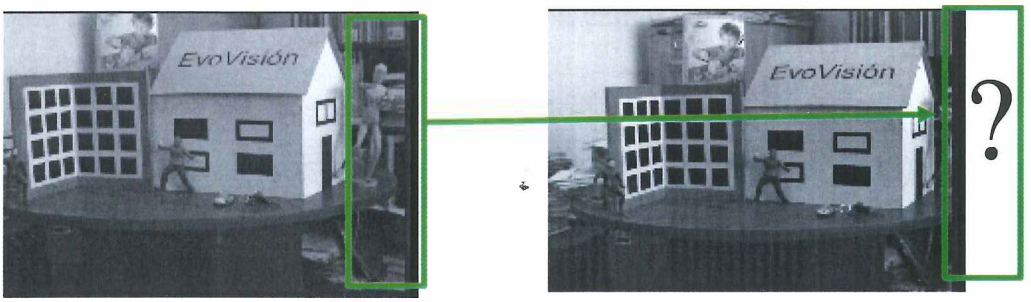


Figura 4: Campo de Visión. La parte final de la imagen izquierda no se encuentra en la imagen derecha y la parte inicial de la imagen derecha no se encuentra en la imagen izquierda debido al campo de visión de la cámara.

4. **Oclusiones**. La oclusión se refiere a un objeto que se encuentra en ambas imágenes que debido al movimiento de la cámara puede ser parcial o totalmente obstruido en alguna de las dos imágenes; sin embargo, el objeto se encuentra dentro del campo de visión de las dos imágenes. La Figura 5 muestra un ejemplo de oclusión. Aqui apreciamos que la diferencia que existe entre el campo de visión y la oclusión es que para el campo de visión, el objeto ya no se encuentra en la imagen debido a la traslación de la cámara.

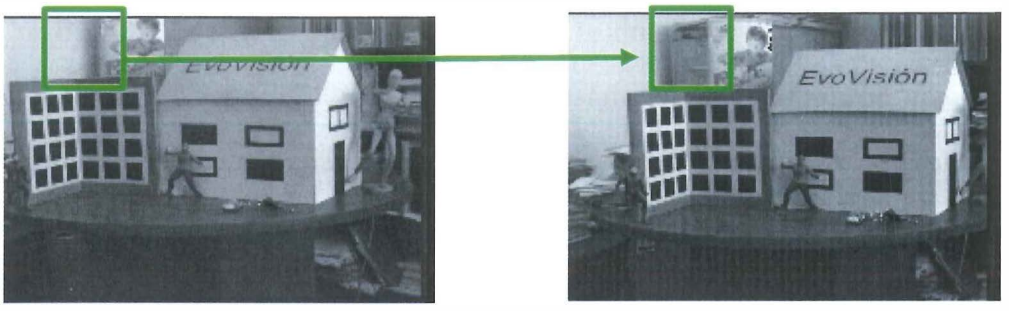


Figura 5: Oclusión. El archivero que se encuentra en la imagen derecha no aparece en la imagen izquierda debido al movimiento de la cámara.

5. **Patrones repetitivos** (*estructuras periódicas de la escena*). Imágenes que contienen estructuras que se repiten n veces dentro de la imagen. La Figura 6 muestra un ejemplo cuando se presentan patrones repetitivos en una imagen.

El problema de correspondencia puede estudiarse en dos direcciones. Una dirección es enfocándose en áreas de la imagen haciendo uso de la correlación como medida de similitud. Una forma sencilla de expresar la correlación entre dos funciones $f(x, y)$ y $w(x, y)$ es

$$c(x, y) = \sum_s \sum_t f(s, t) w(x + s, y + t) \tag{1}$$

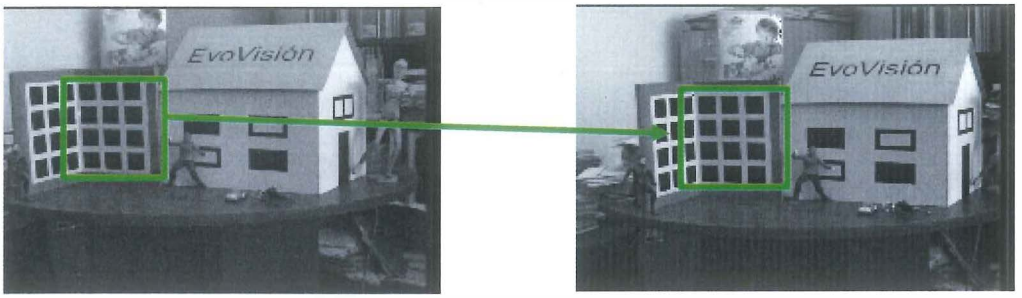


Figura 6: Patrones repetitivos. Los cuadros negros de la mira de calibración ejemplifican este problema inherente de la correspondencia.

para $x = 0, 1, 2, ..., M - 1$, $y = 0, 1, 2, ..., N - 1$, donde M y N es el tamaño de la imagen Gonzalez y Woods (2002). El enfoque basado en áreas es mucho más adecuado cuando se pretende realizar apareamientos en imágenes de tamaño grande (por ejemplo, 400×400 pixeles). La correspondencia en este caso, se conoce como Correspondencia Estéreo-Densa[6]. La otra dirección se enfoca en la información que proporcionan las curvas, vértices, líneas y contornos dentro de la imagen.

Se plantea realizar un trabajo de investigación donde se analice el problema de correspondencia usando un enfoque de vida artificial, este enfoque se detalla en el capítulo IV. La metodología está basada en la correlación como función objetivo; esto es, un estudio basado en áreas de la imagen utilizando una herramienta de vida artificial. Las imágenes estereoscópicas son adquiridas por medio de una cámara superpuesta a un brazo de un robot industrial Staubli[7], ver Figura 7. La Figura 1 a) y b) muestra el par estereoscópico adquirido por la cámara revelando los problemas clásicos encontrados en la visión estéreo que fueron mencionados anteriormente. El movimiento entre ambas imágenes es principalmente de traslación con una ligera rotación.

[6]En inglés, Dense Stereo Matching
[7]Robot manipulador de 6 grados de libertad de alta precisión.

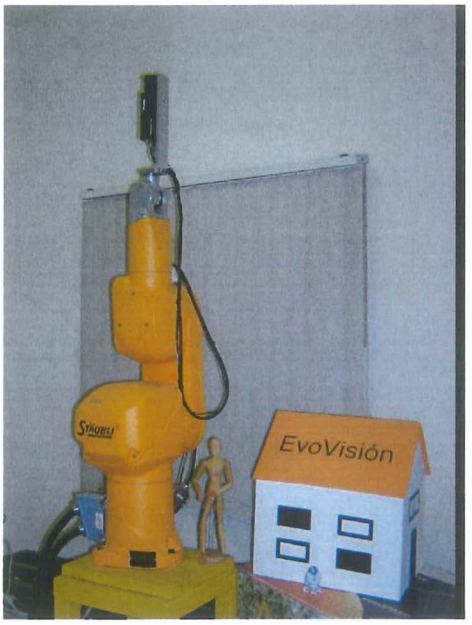


Figura 7: Robot Staubli de 6 grados de libertad que se encuentra en el laboratorio de EvoVisión de CICESE.

I.2 Objetivos de la Tesis

I.2.1 Objetivo General

El objetivo principal de la investigación es diseñar un algoritmo para el problema de correspondencias de un par estéreo que permita reducir el número de cálculos cuidando la calidad de la imagen.

I.2.2 Objetivos Específicos

- Diseñar una interfaz gráfica portable, la cual permita visualizar el proceso de correspondencia entre las dos imágenes.
- Reducir el número de evaluaciones mediante el uso de vida artificial.
- Corroborar la reconstrucción de la escena a través de la reproyección de los pixeles

en una imagen virtual (intermedia).

- Obtener la mejor calidad posible en la imagen.
- Estudio de la librería VXL[8] que fue diseñada para la investigación en visión por computadora.

I.3 Importancia de la Investigación

Nuestro trabajo de investigación pretende tener un grado de importancia dentro del entorno académico y científico al que pertenece. Se desea que sea el inicio de un estudio sobre reconstrucción 3D de un objeto, usando un mínimo número de fotografías. Por tal motivo, el objetivo principal de la tesis es diseñar un algoritmo para la correspondencia entre imágenes, debido a que la correspondencia es un paso fundamental para obtener una reconstrucción.

La razón de usar un nuevo enfoque, es basado en el hecho que el ser humano al ver un objeto, de manera natural *supone* ciertas cosas que se encuentran en la escena, sin necesidad de analizar cada objeto o detalle de manera individual. La idea de *suponer* información y hacer uso de ella en el proceso de correspondencia fue conceptualizada de esa manera con el fin de usar la información contenida en las imágenes de tal forma, que no hubiera necesidad de analizar cada uno de los pixeles de las imágenes.

La idea de *suponer* información permite ahorrar tiempo en cálculos. Este ahorro tiene la finalidad que a futuro se pueda diseñar un algoritmo que funcione en tiempo real.

La motivación que originó y mantuvo el interés sobre el análisis, estudio y desarrollo de un algoritmo para el problema de correspondencia consistió principalmente en dos

[8] En inglés, Vision x Libraries

partes. Por un lado, hasta el momento no se conoce la solución general al problema. Por otro lado, es el foco de atención en distintas comunidades como la fotogrametría, gráficas por computadora, visión por computadora, entre otras. Lo anterior, permite que se tenga la oportunidad de hacer una contribución en el área de investigación. Este trabajo es la primera aportación en el área de visión haciendo uso de vida artificial para estudiar el problema de correspondencia.

I.4 Aplicaciones

La correspondencia de imágenes es un componente clave en la gran mayoría de los procesos que involucran el análisis de imágenes. Es por ello que su campo de aplicación es muy amplio. A continuación, se muestran algunas de las aplicaciones más relevantes dentro de la computación estéreo.

- *Navegación y guía de vehículos autónomos.* Los vehículos autónomos son vehículos que de cierta forma poseen inteligencia propia, lo cual les permite moverse u orientarse en un espacio tal cual lo haría un humano: esquivando obstáculos y moviendo articulaciones, por mencionar dos ejemplos. Existen vehículos diseñados para actuar en el espacio, minas, desiertos, entre otros (Gruen, 1985; Mühlmann *et al.*, 2001).

- *Reconocimiento de objetos.* El reconocimiento de objetos se encarga de identificar características de objetos parciales o completos que aparezcan en ambas imágenes.

- *Visualización de escenas u objetos realísticos.* Es interpretar un objeto físico en tercera dimensión a través de un programa computacional. La visualización real puede ser de gran utilidad en distintos problemas de las ciencias como la medicina, la astrofísica, biología, geología, oceanología, entre otras.

- *Interpretación de fotografías aéreas para la cartografía automatizada.* Las imágenes aéreas generalmente son almacenadas en grandes secuencias usando sensores multiespectrales. La computación estéreo facilita el análisis de terrenos, campos, etc.

  Las escenas varían dependiendo del objetivo de la aplicación. Éstas pueden ser al aire libre o dentro de un espacio cerrado. En las escenas al aire libre se analizan objetos hechos por el hombre tales como edificios y caminos, aquellas que contienen solo objetos naturales y superficies, tales como las montañas, terrenos planos o rocosos, pasto y agua (Barnard y Fischler, 1982). Las escenas que se encuentran dentro de un espacio cerrado se analizan objetos o escenas a pequeña escala hechas por el hombre, o bien, se estudian aspectos del ser humano en si. En este caso, la iluminación puede ser controlada debido a que el espacio es cerrado, lo cual permite que las fotografías sean de mejor calidad.

I.5 Organización de la Tesis

El trabajo está organizado de la siguiente manera:

El Capítulo II da a conocer los antecedentes sobre los algoritmos de correspondencias existentes en la actualidad y una explicación de cómo están clasificados dentro de la comunidad de visión por computadora.

El Capítulo III introduce los conceptos fundamentales de la geometría para visión por computadora. Estos conceptos son de vital importancia durante el proceso de correspondencia. Esta información es de gran ayuda para entender a gran escala el uso del álgebra usada en este trabajo.

El Capítulo IV da una conceptualización básica sobre vida artificial y las herramientas usadas en esta joven área de investigación. Se explica el por qué del uso de esta herramienta y cómo fue enfocada al problema de correspondencia para obtener un nuevo algoritmo que dé solución al problema. Se describe de manera detallada todo el proceso que se llevó a cabo para obtener el algoritmo de correspondencias denominado *algoritmo de infección*. Se presenta en pseudocódigo el algoritmo de correspondencias que fue implementado.

El Capítulo V es la sección donde se da a conocer cómo fue llevada a cabo la experimentación. Los resultados de la experimentación se muestran numérica, gráfica y visualmente.

El Capítulo VI concluye el trabajo de investigación y se comenta la visión a futuro de este trabajo.

Capíulo II

Antecedentes

Dada dos ó más fotografías tomadas desde distintos puntos de vista de una escena, la correspondencia necesita ser establecida en las proyecciones del mismo punto físico de cada fotografía. Cuando la dificultad del problema de correspondencia fue reconocido, se propuso un conjunto de restricciones tanto geométricas como físicas, así como reglas que ayudaran a obtener las correspondencias posibles entre las imágenes de referencia.

Existen distintos enfoques para analizar el problema de correspondencia. El análisis lleva a determinar un algoritmo que permita encontrar las correspondencias entre imágenes. Los métodos utilizados en los algoritmos son clasificados en locales y globales. Los métodos locales utilizan restricciones que se pueden aplicar a los pixeles alrededor del pixel a evaluar. Los métodos globales por el contrario, se refieren al uso de restricciones que se pueden aplicar en toda la imagen.

El uso de ambos métodos en este trabajo de investigación es muy importante. Los métodos locales pueden ser muy eficientes pero son sensitivos localmente en las regiones ambigüas como, por ejemplo, las regiones con oclusiones y regiones con baja textura. *"Sensitivos localmente"* se refiere a que el algoritmo puede fallar fácilmente debido a las ambigüedades o falta de información en esa región.

Por otra parte, los métodos globales son menos sensitivos a estos problemas ya que las restricciones globales proveen ayuda adicional para estas regiones que son difíciles de lograr con una correspondencia local. Sin embargo, estos métodos son computacionalmente más costosos.

II.1 Métodos de Correspondencia Local

En la mayoría de la literatura se puede encontrar que los métodos de correspondencia locales pueden ser diferenciados con algunas variaciones, en técnicas basadas en áreas, o técnicas basadas en características (Dhond y Aggarwal, 1989; Zitnick y Kanade, 2000; Sun, 2002; Fusiello *et al.*, 1997), y otros. Brown *et al.* (2003) clasifica los métodos de correspondencia locales en tres categorías: técnicas basadas en áreas[1], métodos de gradiente y técnicas basadas en características[2].

- *Técnica basada en áreas.* Las técnicas basadas en áreas han sido investigadas extensivamente para aplicaciones comerciales en la fotogrametría estéreo. En esta técnica, usan la correlación como medida de similitud, para lo cual se comparan los valores de intensidad de un vecindario alrededor de un pixel en una imagen con los valores de intensidad del vecindario local de un pixel en la otra imagen. La correlación cruzada es usada para realizar la búsqueda del punto correspondiente en la otra imagen. Se le llama *"correspondencia densa"* cuando se realizan apareamientos con imágenes de tamaño alrededor de 400×400 pixeles. Para llevar a cabo una correspondencia densa es más conveniente usar esta técnica según Barnard y Fischler (1982).

  Los métodos basados en áreas han sido aplicados satisfactoriamente a imágenes aéreas donde la superficie varía suavemente y en aplicaciones donde las imágenes estéreo tienen buena textura (Sun, 2002).

  Fusiello *et al.* (1997) proponen un algoritmo que tiene un enfoque de múltiples ventanas. Para cada pixel realizan la correlación con nueve ventanas distintas de

[1] En inglés Area-Based Method, Block Matching, Correlation-Based Stereo
[2] En inglés Feature-Based Method

3×3 y 7×7 pixeles. Para la correlación usan la técnica de suma de diferencias de cuadrados, SSD[3]. La experimentación consiste de pequeñas imágenes sintéticas y reales no calibradas las cuales tienen un tamaño de 128×128 y 256×256 pixeles.

Lhuillier y Quan (2002) proponen un algoritmo cuasi-denso basado en la estrategia de *"primero el mejor"*, técnica conocida en Inteligencia Artificial. Las imágenes son adquiridas al aire libre, motivo por el cual las imágenes no se calibran.

Sun (2002) presenta un algoritmo rápido para el problema de correspondencia usando subregiones rectangulares y la correlación cruzada rápida. En tiempo logra superar por mucho al trabajo de Cox *et al.* (1996). Hacen uso de imágenes reales y sintéticas de tamaño 256×256 y 512×512. La diferencia entre las dos imágenes de entrada es de una ligera traslación donde no existen oclusiones y ambas contienen buena textura, lo que facilita el proceso de correspondencia.

Ramirez (2000) presenta un algoritmo de búsqueda exhaustiva para el problema de correspondencia. El algoritmo busca pixel a pixel la correspondencia izquierda-derecha y derecha-izquierda a lo largo de una línea llamada línea epipolar que se encuentra dentro de una región de búsqueda. Hace uso de la correlación ZNCC[4] como medida de similitud. El algoritmo correlaciona los pixeles de la imagen izquierda con los de la imagen derecha, si la correlación es baja, entonces, realiza una búsqueda de derecha a izquierda. La dimensión del par estéreo es de 484×768 pixeles, por lo que se considera una correspondencia densa. Se implementó el algoritmo exhaustivo de Ramirez (2000) para tomarlo como referencia en esta tesis. En la Tabla I se muestra un bosquejo de algunos algoritmos basados en áreas.

[3]En inglés, Sum of Squares Differences
[4]En inglés, Zero Normalized Cross Correlation

Tabla I: Relación de los algoritmos que usan un método basado en áreas

AUTORES	ENFOQUE	METODOLOGIA (Correlación)	TAMAÑO
Fusiello *et al.* (1997)	Múltiples ventanas adaptivas	SSD	128×128 256×256
Lhuillier y Quan (2002)	Algoritmo quasi-denso	ZNCC	256×256 512×512 768×512
Sun (2002)	Subregiones rectangulares Técnicas de máxima superficie 3D	Cruzada Rápida	256×256 512×512 1000×1000
Ramirez (2000)	Exhaustivo	ZNCC	484×768

- *Métodos de Gradiente.* Los métodos de gradiente o también llamados de flujo óptico, buscan determinar pequeñas disparidades (rotación y traslación) locales entre dos imágenes mediante la formulación de una ecuación diferencial relacionando el movimiento y la brillantez de la imagen.

  Para hacer esto, suponen que la brillantez de la imagen de un punto en la escena es constante entre las dos vistas. La métrica usada en este enfoque es la misma que puede ser usada en la técnica basada en áreas que es la suma de las diferencias de cuadrados (SSD).

  Gruen (1985) estima transformaciones afines locales entre las imágenes para compensar los efectos de perspectiva. La estimación de los parámetros del modelo debe acomodar suficientes parámetros para ser capaz de modelar las distorsiones.

  En teoría, los métodos de gradiente solo pueden estimar las disparidades más de la mitad de un pixel donde las derivativas locales son solamente válidas en ese rango. Cuando los pixeles son altamente correlacionados, uno o más pixeles pueden ser

estimados en la práctica. Sin embargo, un procesamiento jerárquico es necesario para aplicar los métodos de gradiente en visión estéreo, donde los rangos de las disparidades son mucho más grandes que un pixel.

- *Técnica basada en características.* En este enfoque, las imágenes de entrada son preprocesadas por un operador el cual extrae las características que son estables bajo el cambio producido por el punto de vista distinto de la segunda imagen. Después, el proceso de correspondencia es aplicado a los atributos asociados con las características detectadas. Las características que son robustas al cambio de perspectiva y ampliamente usadas dentro de visión son los bordes, esquinas, segmentos de línea y segmentos de curva. Las esquinas y bordes son fáciles de detectar pero hay problemas cuando se tienen oclusiones. Los segmentos de línea y curva requieren tiempo computacional extra, pero es más sencillo corregir las oclusiones. Los círculos, elipses y regiones poligonales son usados de igual forma como características para la correspondencia entre imágenes.

Tipos de Características

- *Elementos de bordes.* Existen diversos operadores de bordes para encontrar los elementos de bordes en una imagen. Por ejemplo, el detector de bordes Canny, Nalwa-Binford, Sarkar-Boyer y Sobel (Heath *et al.*, 1998). Los atributos de los elementos de borde usados para la correspondencia pueden ser: las coordenadas, orientaciones locales e intensidades locales (obscuro-iluminado o iluminado-obscuro).

- *Esquinas.* Existe una variedad de detectores de esquinas. En los 80's, tenemos a Dreschler y Nagel(1982), Kitchen y Rosenfeld(1982), Zuniga y Haralick(1983), entre otros. El detector de esquinas de Harris(1988) es uno de

los detectores de esquinas más populares dentro de la comunidad de visión. Los atributos de esquinas que pueden ser usados dentro de la correspondencia son las coordenadas de las esquinas y el tipo de uniones de las esquinas (unión-L, unión-Y, unión-A, etc). Olague y Hernández (2004) proponen un nuevo modelo de esquinas paramétrico capáz de modelar e identificar ciertas clases de variaciones de intensidad descritas por estructuras poligonales de una escena en 3D.

- *Segmentos de línea.* Para extraer los segmentos de línea, primero se debe aplicar un operador de bordes. Se hace uso de un criterio adicional para extraer los segmentos de línea como la medida para la distancia, similitud y colinealidad. Los atributos de segmentos de línea que pueden ser usados para la correspondencia son las coordenadas de los puntos y la orientación de los segmentos de línea.

- *Segmentos de curva.* No han sido muy usados por la ambigüedad que presentan, ya que un punto en una curva puede hacer correspondencia con cualquier punto en la otra curva.

- *Círculos y elipses.* Estas características se presentan principalmente en escenarios de espacio cerrado y aplicables a la detección de defectos en partes industriales. Los atributos que pueden ser usados para la correspondencia pueden ser las áreas en unidades de pixel, las coordenadas del centro de las figuras geométricas.

- *Regiones poligonales.* Estas regiones pueden ser definidas como gotas, las cuales tienen formas irregulares, por tal motivo, pueden no hacer una correspondencia perfecta con las regiones de la otra imagen. Computacionalmente es costoso extraerlas de la imagen. Los atributos que pueden ser usados en

la correspondencia son las áreas por regiones y localización de centroides en las regiones.

Birchfield y Tomasi (1999) proponen un algoritmo para resolver el problema de correspondencia en el cual hacen uso de la segmentación en el par estéro. La segmentación es hecha en pequeñas regiones de la imagen a las que posteriormente se les aplica la correspondencia. Para la iteración hacen uso de un algoritmo voráz. Las imágenes usadas son imágenes reales texturizadas con traslación.

II.2 Métodos de Correspondencia Global

Los métodos globales permiten analizar y tomar en cuenta factores que ayudan a los métodos locales a reducir la sensitividad producida por las ambigüedades. El enfoque más común dentro de estos métodos es el de programación dinámica, en el cual se hace uso de la restricción de orden para optimizar las correspondencias a lo largo de la línea epipolar.

Los enfoques que más se utilizan dentro de los métodos globales son:

- *Programación dinámica.* Es un método matemático que reduce la complejidad de problemas de optimización descomponiéndolos en subproblemas más pequeños y sencillos. Una función de costo global es calculada en etapas, con un número de transiciones entre etapas que son definidas por un conjunto de restricciones. En el caso de la correspondencia de imágenes, la restricción de orden empleando la línea epipolar permite que la función global sea determinada como una ruta de costo mínimo. El costo de la ruta óptima es la suma de los costos de las rutas parciales obtenidas recursivamente. Un ejemplo de ello, es el trabajo realizado por Ohta y Kanade (1985). Generalmente los valores de costo se almacenan en una tabla para evitar realizar las mismas operaciones, de esa manera nada más

se busca el resultado de la operación.

- *Curvas intrínsecas.* Tomasi y Manduchi (1998) proponen una alternativa para la parte global de la correspondencia entre imágenes que consiste en una representación de líneas de búsqueda llamada curvas intrínsecas. El beneficio principal de la representación de estas curvas es su invarianza a la disparidad pero continúa siendo afectada por los problemas inherentes de la correspondencia.

- *Representación en grafos.* Es la representación del problema a través de grafos. La limitante más significativa de la programación dinámica en la correspondencia es su inhabilidad de incorporar las restricciones de continuidad tanto vertical como horizontal. Muchos enfoques tratan de mantener de alguna manera la programación dinámica pero no se explotan completamente las restricciones bidimensionales que estén disponibles (Brown *et al.*, 2003). Un enfoque global alternativo que explota estas restricciones es representar el problema de correspondencia transformándolo en el problema usando grafos. Un ejemplo de ello es el trabajo hecho por Roy y Cox (1998).

II.3 Conclusiones

En este capítulo se presentó de manera general algunos métodos existentes para el problema de correspondencia así como también, las técnicas usadas en cada uno de los métodos.

La técnica basada en áreas es una de las técnicas que más se utilizan, ya que los resultados son más confiables, pero se tiene la desventaja que el costo computacional es mayor. Esta técnica es más apropiada cuando el procesamiento es denso, esto es, que el par de imágenes tengan un tamaño de aproximadamente 160,000 pixeles (400×400). El tamaño de las imágenes usadas en este trabajo son de 484×768 pixeles.

Muchos de los trabajos sobre correspondencia realizan un procesamiento previo en las imágenes para facilitar el proceso de búsqueda. Un ejemplo de ese procesamiento es rectificando las imágenes antes de iniciar la correspondencia. La rectificación de imágenes se refiere a acomodar los pixeles de las dos imágenes de tal forma que los pixeles que corresponden queden sobre la misma línea horizontal. El concepto de rectificación en visión por computadora es cuando las líneas epipolares se calculan de tal forma que quedan horizontalmente en las dos imágenes.

En este trabajo se usará la técnica basada en áreas en la parte local del proceso, utilizando la correlación, la restricción epipolar y la restricción de orientación; estos conceptos se explican en el capítulo III. En la parte global del proceso, se usará una herramienta de vida artificial que se detalla en el capítulo IV.

Capíulo III

Conceptualización Geométrica del Problema

III.1 Introducción

La correspondencia entre un par estéreo puede ser explicada geométricamente, debido a la relación física que existe entre el mundo real, el sistema de adquisición y las transformaciones geométricas que dan lugar a la formación de la imagen.

Las imágenes que son usadas en visión estéreo pueden ser reales o sintéticas. Si las imágenes a usar en el proceso son reales, éstas deben ser adquiridas de una escena por medio de una o más cámaras. De lo contrario, son diseñadas especialmente para el proceso con la ayuda de algún software de diseño.

La explicación geométrica de la relación que existe entre dos imágenes adquiridas desde distintos puntos de vista, parte del hecho de que cada cámara tiene su propia geometría la cual puede ser modelada algebraicamente para obtener los parámetros necesarios que dan a conocer la relación entre el espacio 3D y el 2D. Estos parámetros están englobados en un concepto clave llamado matriz fundamental, la cual representa la relación geométrica entre dos imágenes.

III.2 Geometría Proyectiva

No es sencillo hablar de geometría proyectiva cuando la mayoría de nosotros estamos acostumbrados a trabajar sobre un sistema Euclidiano y no proyectivo. La gometría proyectiva va mas allá de una representación métrica de los objetos, de rotarlos, trasladarlos o escalarlos; es una representación matemática general de la geometría en la cual intervienen otro tipo de transformaciones. Por ello, la geometría Euclidiana es solamente un caso especial de la geometría proyectiva y la información que maneja es insuficiente para los problemas de visión por computadora donde el uso de cámaras es esencial.

La geometría proyectiva es usada como la base de muchas técnicas en robótica, como en el estudio de la cinemática y geometría de robots manipuladores, así como también es usada en gráficas por computadora para representar las transformaciones geométricas de los objetos (Faugeras, 1996).

Espacio Proyectivo. El espacio proyectivo es representado como P^n, donde n indica la dimensión del espacio. Un punto de un espacio proyectivo n-dimensional, es representado por un $n+1$ vector de coordenadas $x = [x_1, ..., x_{n+1}]^t$, donde al menos uno de los x_i es diferente de cero. Los números x_i son algunas veces llamados coordenadas proyectivas o coordenadas homogéneas del punto, y el vector x es llamado vector de coordenadas. Dos vectores $n+1$ $[x_1, ..., x_{n+1}]^t$ y $[y_1, ..., y_{n+1}]^t$ representan el mismo punto si y sólo si existe un escalar λ tal que $x_i = \lambda \neq 0 y_i$ para toda $1 \leq i \leq n+1$ (Faugeras, 1996).

La matriz que hace posible las transformaciones proyectivas es la siguiente.

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{2}$$

- *Línea Proyectiva.* El espacio P^1 es conocido como la línea proyectiva l, es el espacio más sencillo de todos los espacios proyectivos. Un punto x en la línea es representado por las coordenadas homogéneas $[x_1, x_2]^t$. Un punto para el cual $x_2 = 0$ es un punto ideal o tambien llamado punto en el infinito de la línea.

- *Plano Proyectivo.* El espacio P^2 es conocido como el plano proyectivo, ya que en visión por computadora se modela la imagen como un plano proyectivo.

III.3 Geometría Afín

El espacio afín n-dimensional puede ser obtenido a través del espacio proyectivo; la propiedad invariante es igual a la línea en el infinito, donde dos líneas paralelas pueden intersectarse. Por ello, el grupo de transformaciones afines es solo un subgrupo del grupo de transformaciones proyectivas. El invariante afín fundamental, es el cociente de longitudes a lo largo de la línea. Esto es, dado tres puntos alineados A,B,C y un punto en el infinito D, la longitud está dada por el cociente cruzado[1]. Por definición el cociente cruzado está dado por:

$$(A, B; C, D) = \frac{\overline{AC}}{\overline{AD}} \div \frac{\overline{BC}}{\overline{BD}}$$

donde, $\overline{AC}$ es la magnitud del segmento de A a C. De igual forma para $\overline{AD}$, $\overline{BC}$ y $\overline{BD}$.

[1] En inglés Cross-Ratio

La representación matricial en coordenadas homogéneas de una transformación afín, es la siguiente:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{3}$$

o en forma compacta

$$\mathbf{x}' = H_A x = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^t & 1 \end{bmatrix} \mathbf{x}, \tag{4}$$

donde, la matriz A es una matriz de tamaño 2×2 no-singular, t_x y t_y son los valores de traslación en x y y respectivamente.

III.4 Geometría Euclidiana

La geometría Euclidiana es la geometría más restringida de las existentes ya que sólo incluye tranformaciones rígidas como las rotaciones y traslaciones, representantadas por matrices de transformación.

Las transformaciones Euclidianas modelan el movimiento de un objeto rígido. Por tal motivo, se puede observar que los parámetros de la matriz, solo son valores de ángulos que corresponden a las rotaciones del objeto y los valores de la traslación tanto en x como en y.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & t_x \\ sin\theta & cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{5}$$

o en forma compacta

$$\mathbf{x}' = H_E x = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^t & 1 \end{bmatrix} \mathbf{x}, \tag{6}$$

donde la matriz R es una matriz de rotación de 2×2 (matriz ortogonal tal que $R^t R =$

$RR^t = I$), **t** es un 2-vector de traslación y **0** un 2-vector nulo. Los casos especiales son, cuando se tiene solo rotación **t**=0 ó solo traslación, cuando **R**=I.

Jerarquía de transformaciones. Dentro de las transformaciones geométricas mostradas anteriormente existe la siguiente jerarquía.

$$Proyectiva \supset Afin \supset Euclidiana$$

Cuando aplicamos una transformación geométrica a un grupo de transformación, la estructura del espacio permanece sin cambios. En general, las transformaciones se refieren a los mapeos que existen entre cada uno de los sistemas coordenados. Cada uno de los grupos de transformaciones tiene sus propiedades invariantes. La geometría euclidiana es definida por el grupo de desplazamientos rígidos; y finalmente, la geometría proyectiva por las colineaciones proyectivas.

III.5 Proceso de adquisición de la imagen

Una cámara es una herramienta que proyecta objetos 3D (de la escena) en el espacio 2D (imagen de los objetos). A una cámara se encuentra ligado un sistema coordenado de origen C llamado centro de proyección. (C, x) y (C, y) son los ejes paralelos al plano de la imagen correspondientes a la dirección de las filas y las columnas de los pixeles de la imagen. El eje (C, z), llamado eje óptico, está orientado hacia la escena y es perpendicular al plano de la imagen.

Para la formación de la imagen intervienen tres sistemas de coordenadas, todos en espacio Euclidiano:

1. El sistema coordenado del mundo expresado como X, Y, Z.
2. El sistema coordenado de la cámara expresado en milímetros (mm) como x', y', z'.

3. El sistema coordenado de la imagen expresado en pixeles como u, v.

Para pasar de coordenadas definidas en el sistema coordenado del mundo a coordenadas de la imagen expresadas en pixeles, se necesitan tres pasos:

1. *Transformación mundo-cámara.* Los puntos tridimensionales expresados en el sistema coordenado del mundo experimentan un cambio al pasar al sistema coordenado de la cámara. Este cambio de sistema coordenado comprende una transformación rígida que consiste en seis parámetros (tres para las rotaciones y tres para las traslaciones). Estos parámetros son la posición y orientación que tiene la cámara con respecto a la escena, y son llamados parámetros extrínsecos.

2. *Transformación proyectiva cámara-imagen.* Una vez que surge el cambio del sistema coordenado, los puntos tridimensionales expresados en el sistema coordenado de la cámara son proyectados sobre el plano de la imagen. Las nuevas coordenadas reciben el nombre de coordenadas normalizadas.

3. *Transformación afín cámara-imagen.* Finalmente, se necesita obtener las coordenadas en el sistema de la imagen expresado en pixeles. Para ello, las coordenadas normalizadas se someten a una transformación afín del plano. Esta transformación, en este caso particular está compuesta de cuatro parámetros intrínsecos.

III.5.1 Modelo de Cámara Obscura

El modelo de cámara obscura[2] está basado en que un punto en el espacio con coordenadas $M = (X, Y, Z)^t$ es mapeado al punto m donde la línea que une al punto M con el centro de proyección C pasa a través del plano de la imagen, ver Figura 8. Este modelo indica que cada punto en la escena pasa a través del centro óptico de la cámara para

[2]Conocido en inglés como Pinhole Model

proyectarse en el plano de la imagen. Por lo tanto, existen dos transformaciones principales que son llevadas a cabo en el proceso de formación de la imagen. La primera es, la transformación que existe del sistema coordenado del mundo al sistema coordenado de la cámara $M \rightarrow C$ y la segunda, la transformación que existe del sistema coordenado de la cámara al sistema coordenado de la imagen $C \rightarrow I$. Mediante estas dos transformaciones y una proyección se obtiene finalmente la imagen.

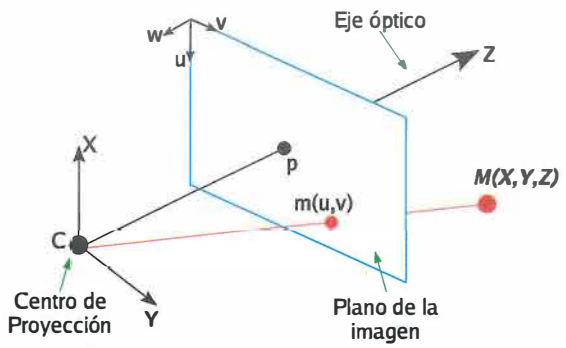


Figura 8: *Modelo Geométrico de una Cámara.* La figura muestra la geometría de un modelo de cámara obscura. C es el centro de proyección y p el punto principal. El plano de la imagen está situado al frente del centro de la cámara.

En la Figura 8 se observa el modelo geométrico de la cámara, que consiste en la proyección de perspectiva sobre el plano de la imagen. El punto p es proyectado perpendicularmente hacia el plano de la imagen formando una recta llamada eje óptico. El centro de proyección C se encuentra a cierta distancia del plano de la imagen, esta distancia es la distancia focal f. El punto M se proyecta en la imagen a lo largo de una recta que pasa por M y C. Al elegir un sistema coordenado fijo en la cámara, el plano

x-y de este sistema coordenado es paralelo al plano de la imagen, y el eje z se confunde con el eje óptico. El origen del sistema coordenado de la cámara se encuentra en C, siendo X, Y y Z las coordenadas del punto M. En ese mismo sistema, las coordenadas (x', y', z') de la proyección de M sobre el plano de la imagen, en el sistema coordenado de la cámara, están dadas por las siguientes ecuaciones:

$$x' = \frac{fx}{z}, \qquad y' = \frac{fy}{z}, \qquad z' = f \tag{7}$$

Al utilizar coordenadas homogéneas tenemos que $M = (X, Y, Z, 1)$ y el cálculo de su proyección, en este caso $m = (x', y', z')$ esta dado por las siguientes ecuaciones:

$$\begin{bmatrix} sx' \\ sy' \\ sz' \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}}_{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{8}$$

donde, P representa la matriz de proyección, la cual ayuda en el proceso de pasar las coordenadas del mundo a coordenadas de la imagen. s representa el factor de escala y f la distancia focal que existe del centro de proyección de la cámara C al plano de la imagen. Tenemos que:

$$x' = \frac{X}{s}, \qquad y' = \frac{Y}{s}, \qquad z' = \frac{Z}{s}, \qquad s = \frac{Z}{f} \tag{9}$$

Es importante mencionar que el modelo geométrico de una cámara se caracteriza por dos transformaciones: Una *proyección*, la cual transforma un punto del espacio en un punto de la imagen. La segunda, una *transformación*, de un sistema coordenado métrico ligado a la cámara, a un sistema coordenado en unidades pixeles ligado a la imagen.

III.5.2 Transformación del Sistema Coordenado del Mundo al Sistema Coordenado de la Cámara (Mundo-Cámara)

Parámetros Extrínsecos

El punto M de la escena es proyectado hacia el plano de la imagen, donde M mantiene una posición en la escena con respecto a la cámara. La orientación y la posición que tiene la cámara son llamados parámetros extrínsecos de la cámara o bien, parámetros de orientación exterior en fotogrametría. Para determinar estos parámetros se coloca un objeto de referencia en la escena, en este caso, una mira de calibración. La mira de calibración se compone por diversos puntos que son perfectamente conocidos y medidos en el sistema coordenado del mundo (X, Y, Z). Cada uno de estos puntos son proyectados en la imagen a través de la cámara, y de esta forma pueden ser medidos en el sistema coordenado de la imagen, obteniendo así, las coordenadas de los puntos de la mira en el sistema coordenado de la imagen.

La transformación mundo-cámara se compone de tres rotaciones y tres traslaciones. Esta transformación rígida puede ser introducida en coordenadas homogéneas como sigue:

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathbf{t}} & 1 \end{bmatrix} \tag{10}$$

La matriz A representa una transformación rígida de seis grados de libertad, tres grados son para rotaciones y tres para traslaciones. A sus elementos se les llama parámetros extrínsecos.

III.5.3 Transformación del Sistema Coordenado de la Cámara al Sistema Coordenado de la Imagen (Cámara-Imagen)

Parámetros Intrínsecos

Los puntos de la imagen son medidos en pixeles en un sistema coordenado bidimensional u-v asociado a la imagen, ver Figura 8. Con el propósito de poder escribir la matriz de transformación del sistema coordenado de la cámara al sistema coordenado de la imagen, debemos introducir los siguientes parámetros: u_0, v_0 y w_0, que son las coordenadas del centro de proyección C en el sistema coordenado de la imagen, k_u es el factor de escala horizontal y k_v el factor de escala horizontal. Las unidades de ambos factores de escala esta dada en pixeles/mm. Se manejan dos factores de escala debido a que los pixeles de una cámara rara vez son cuadrados. La transformación del sistema coordenado de la cámara al sistema coordenado de la imagen se escribe de la siguiente manera para el punto m que es proyectado en el plano de la imagen.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \\ w_0 \end{bmatrix}, \tag{11}$$

donde u, v, w son las coordenadas del punto m en el sistema coordenado de la imagen. La transformación que se presenta, es una transformación afín que representa un cambio de escala, una rotación y una traslación. En el primer y tercer renglón de la segunda matriz de la ecuación 11 se tiene un valor de -1, esto es debido a que los ejes que se corresponden, u con X y w con Z, están en sentidos opuestos, ver Figura 8. La componente w es nula ya que el sistema coordenado de la imagen solo maneja dos dimensiones, por lo tanto, se ignora el tercer el renglón y se escribe la transformación como una matriz K de tamaño 3×4. La transformación que representa el cambio de

coordenadas del espacio proyectivo al plano de la imagen es:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = K \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}, \tag{12}$$

donde, s es el factor de escala vertical y

$$K = \begin{bmatrix} -k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

El modelo geométrico de la cámara puede representarse a través de ecuaciones una vez que K y P sean multiplicadas. Estas ecuaciones indican la relación entre las coordenadas de la cámara (x, y, z) en el punto M y las coordenadas de la imagen (u, v) en el punto m.

$$u = -k_u \frac{fx}{z} + u_0 \qquad v = k_v \frac{fy}{z} + v_0$$

De esta forma obtenemos el producto de KP:

$$KP = \begin{bmatrix} -k_u & 0 & 0 & u_0 \\ 0 & k_v & 0 & v_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} = \begin{bmatrix} -k_u & 0 & \frac{u_0}{f} & 0 \\ 0 & k_v & \frac{v_0}{f} & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \tag{14}$$

Los elementos de la matriz están definidas con un factor multiplicativo indeterminado, lo que indica que al multiplicar los coeficientes por f, este no afecta el resultado.

La matriz I_c es el resultado de la multiplicación de una transformación afín cámara-imagen (matriz C) y una transformación proyectiva, ver Ecuación 16.

$$C = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$I_c = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{16}$$

Esta matriz, representa la aplicación lineal del espacio proyectivo hacia el plano proyectivo, por lo que podemos expresar la transformación perspectiva como sigue:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = I_c \begin{bmatrix} x \\ y \\ z \\ s \end{bmatrix} \tag{17}$$

La matriz I_c está compuesta por los parámetros intrínsecos estimados al efectuar la calibración, α_u, α_v, u_0 y v_0, donde, $\alpha_u = -k_u f$ y $\alpha_v = k_v f$. Al introducir las coordenadas de la cámara sin una dimensión, x_c, y_c y z_c, tenemos que:

$$x_c = x/z \qquad y_c = y/z \qquad z_c = 1$$

La relación entre las coordenadas de la imagen y las coordenadas de la cámara está dada por las siguientes ecuaciones:

$$u = \alpha_u x_c + u_0 \qquad con \qquad \alpha_u < 0$$
$$v = \alpha_v y_c + v_0$$

En forma matricial tenemos:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \tag{18}$$

La matriz I_c contiene los parámetros intrínsecos (Cámara-Imagen, $C \rightarrow I$) y la matriz A, ver Ecuación 10, contiene los parámetros extrínsecos (Mundo-Cámara, $P \rightarrow C$). Una vez que fueron descritas, podemos describir la transformación directa Mundo-Imagen ($P \rightarrow I$) expresado en el sistema coordenado de la imagen. En este proceso intervienen los tres tipos de transformaciones, Afín, Proyectiva y Euclidiana, respectivamente.

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{Afin} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{Proyectiva} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{Euclidiana} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{19}$$

Esta transformación es englobada en una matriz P de tamaño 3×4, llamada *matriz de proyección perspectiva*:

$$P = I_c A$$

$$P = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$$= \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_u r_{11} + r_{31}u_0 & \alpha_u r_{12} + r_{32}u_0 & \alpha_u r_{13} + r_{33}u_0 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_{21} + r_{31}v_0 & \alpha_v r_{22} + r_{32}v_0 & \alpha_v r_{23} + r_{33}v_0 & \alpha_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

$$P = \begin{bmatrix} \alpha_u \mathbf{r_1} + u_0 \mathbf{r_3} & \alpha_u t_x + u_0 t_z \\ \alpha_v \mathbf{r_2} + v_0 \mathbf{r_3} & \alpha_v t_y + v_0 t_z \\ \mathbf{r_3} & t_z \end{bmatrix} \tag{21}$$

El resultado de $P = I_c A$ está escrito de forma compacta al utilizar la notación $\mathbf{r_i} = (\mathbf{r_{i1}}\ \mathbf{r_{i2}}\ \mathbf{r_{i3}})$. Entonces, tenemos que la matriz A puede representarse de la siguiente forma:

$$A = \begin{bmatrix} \mathbf{r_1} & t_x \\ \mathbf{r_2} & t_y \\ \mathbf{r_3} & t_z \\ \mathbf{0} & 1 \end{bmatrix} \tag{22}$$

La matriz de proyección perspectiva P puede ser escrita en forma compacta de la siguiente forma:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}}_{Matriz P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{23}$$

Las coordenadas X,Y y Z representan las coordenadas de un punto M tridimensional en el sistema coordenado de ésta. La matriz de proyección todavía puede expresarse de manera más compacta:

$$P = \begin{bmatrix} \mathbf{m_1} & m_{14} \\ \mathbf{m_2} & m_{24} \\ \mathbf{m_3} & m_{34} \end{bmatrix} \tag{24}$$

Al calcular la matriz P con $I_c A$, teniendo en cuenta propiedades de ortonormalidad de la rotación y resaltando que se debe obtener un valor negativo para α_u, obtenemos un conjunto de ecuaciones que permiten calcular los parámetros intrínsecos y extrínsecos en función de los coeficientes de P (Horaud y Monga, 1995). De esta forma, tenemos:

$$
\begin{aligned}
u_0 &= \mathbf{m_1}.\mathbf{m_3} \\
v_0 &= \mathbf{m_2}.\mathbf{m_3} \\
t_x &= 1/\alpha_u(m_{14} - u_0 m_{34}) \\
t_y &= 1/\alpha_v(m_{24} - u_0 m_{34}) \\
t_z &= m_{34} \\
\mathbf{r_1} &= 1/\alpha_u(\mathbf{m_1} - u_0\mathbf{m_3}) \\
\mathbf{r_2} &= 1/\alpha_v(\mathbf{m_2} - v_0\mathbf{m_3}) \\
\mathbf{r_3} &= \mathbf{m_3}
\end{aligned}
\tag{25}
$$

donde (.) denota el producto punto. Para conocer los parámetros intrínsecos se debe primero estimar los coeficientes de la matriz de proyección P y después extraer los parámetros de la cámara a partir de los coeficientes dados por las ecuaciones anteriores.

III.5.4 Calibración de una cámara

Las coordenadas en la imagen de un punto de la escena pueden escribirse usando la matriz P y las coordenadas tridimensionales (X, Y, Z).

$$u = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \tag{26}$$

$$v = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \tag{27}$$

Estas dos ecuaciones describen la línea que pasa por el centro de proyección y el punto de la imagen (u, v), en el sistema coordenado del mundo. Esta línea es la recta que pasa por C y el punto en la imagen m en la Figura 8, también llamada línea de visión.

Con el fin de calcular los coeficientes de la matriz P es necesario escribir un sistema de ecuaciones a partir de los puntos de la mira de calibración y su proyección en la imagen. Cada punto (X_i, Y_i, Z_i) se proyecta en (u_i, v_i) generándose dos ecuaciones (26) y (27). Estas ecuaciones son linealmente independientes con respecto a los coeficientes

de la matriz. Las ecuaciones (26) y (27) pueden ser reescritas como una combinación lineal de los m_{ij} parámetros:

$$X_i m_{11} + Y_i m_{12} + Z_i m_{13} + m_{14} - u_i X_i m_{31} - u_i Y_i m_{32} - u_i Z_i m_{33} = u_i m_{34} \tag{28}$$

$$X_i m_{21} + Y_i m_{22} + Z_i m_{23} + m_{24} - u_i X_i m_{31} - u_i Y_i m_{32} - u_i Z_i m_{33} = u_i m_{34} \tag{29}$$

De esta forma, se obtienen $2n$ ecuaciones para n puntos. Las ecuaciones pueden ser reescritas en forma matricial de la siguiente forma:

$$\mathbf{Qx} = \mathbf{u} \tag{30}$$

donde $\mathbf{Q}$ es una matriz de tamaño $2n \times 11$, el vector x de 11×1 y el vector u de $2n \times 1$. De forma más detallada tenemos que:

$$\mathbf{Q} = \begin{bmatrix} & & & & & & \vdots & & & & \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i X_i & -u_i Y_i & -u_i Z_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i X_i & -v_i Y_i & -v_i Z_i \\ & & & & & & \vdots & & & & \end{bmatrix} \tag{31}$$

$$\mathbf{x} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{21} & m_{22} & m_{23} & m_{24} & m_{31} & m_{32} & m_{33} \end{bmatrix}^t \tag{32}$$

$$\mathbf{u} = \begin{bmatrix} \vdots \\ u_i m_{34} \\ v_i m_{34} \\ \vdots \end{bmatrix} \tag{33}$$

Por lo tanto, tenemos que:

$$\begin{bmatrix} & & & & & \vdots & & & & & \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i \\ & & & & & \vdots & & & & & \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} \vdots \\ u_i m_{34} \\ v_i m_{34} \\ \vdots \end{bmatrix} \tag{34}$$

Los valores reales de los doce elementos que componen la matriz de proyección P, permiten conocer los parámetros intrínsecos y extrínsecos a partir de la Ecuación (26). Para ello, se hace uso del método de Faugeras y Toscani (1986), el cual da una solución robusta al problema que se describe en la siguiente sección.

Método Faugeras-Toscani

Al calcular los coeficientes de P en función de los coeficientes que componen las matrices I_c y A, encontramos que:

$$m_{31} = r_{31}, \qquad m_{32} = r_{32}, \qquad m_{33} = r_{33}$$

Aplicando identidades trigonómetricas se puede verificar que: $r_{31}^2 + r_{32}^2 + r_{33}^2 = 1$, la cual es utilizada como restricción. Obteniendo así:

$$\| \mathbf{m_3} \|^2 = m_{31}^2 + m_{32}^2 + m_{33}^2 = 1$$

A continuación se muestra cómo calcular P tomando en cuenta la restricción anterior. La ecuación 30 se puede desglosar de la siguiente manera:

$$\mathbf{Bx_9} + \mathbf{Cx_3} = \mathbf{0} \tag{35}$$

donde $\mathbf{B}$ es una matriz de tamaño $2n \times 9$ y $\mathbf{C}$ una matriz de $2n \times 3$:

$$\mathbf{B} = \begin{bmatrix} & & & & \vdots & & & & \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i \\ & & & & \vdots & & & & \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} & \vdots & \\ -u_iX_i & -u_iY_i & -u_iZ_i \\ -v_iX_i & -v_iY_i & -v_iZ_i \\ & \vdots & \end{bmatrix} \tag{36}$$

Como puede observarse en la Ecuación (35), es necesario descomponer el vector $\mathbf{x_{11}}$ en dos incógnitas $\mathbf{x_9}$ y $\mathbf{x_3}$, así tenemos:

$$\mathbf{x_9} = \begin{bmatrix} \mathbf{m_1} & m_{14} & \mathbf{m_2} & m_{24} & m_{34} \end{bmatrix}^t \tag{37}$$

y

$$\mathbf{x_3} = (\mathbf{m_3}) \tag{38}$$

El vector $\mathbf{x_3}$ representa a las variables necesarias para aplicar la restricción.

El criterio óptimo a minimizar para poder solucionar el sistema de ecuaciones es el siguiente:

$$\| \mathbf{Bx_9} + \mathbf{Cx_3} \|^2, \tag{39}$$

con la restricción:

$$\| \mathbf{x_3} \|^2 = \mathbf{1}$$

Utilizando el criterio de Lagrange, la función criterio puede ser escrita de la siguiente manera:

$$\mathbf{Q} = \| \mathbf{Bx_9} + \mathbf{Cx_3} \|^2 + \lambda(\mathbf{1} - \| \mathbf{x_3} \|^2) \tag{40}$$

desarrollando esta expresión tenemos:

$$Q = x_9^t B^t B x_9 + x_3^t C^t C x_3 + bf x_9^t B^t C x_3 + x_3^t C^t B x_9 + \lambda(1 - x_3^t x_3) \tag{41}$$

Igualando a cero las derivadas parciales de $\mathbf{x_9}$ y $\mathbf{x_3}$, resultan las siguientes dos ecuaciones:

$$B^t B x_9 + B^t C x_3 = 0$$

$$C^C x_3 + C^t B x_9 - \lambda x_3 = 0,$$

de las cuales obtenemos:

$$x_9 = -(B^t B)^{-1} B^t C x_3 \tag{42}$$

$$D x_3 = \lambda x_3 \tag{43}$$

$$D = C^t C - C^t B (B^t B)^{-1} B^t C \tag{44}$$

Finalmente, sustituyendo estas ecuaciones en (41) tenemos:

$$Q = x_3^t D x_3 = \lambda x_3^t x_3 = \lambda \tag{45}$$

D es una matriz simétrica, y definida positiva de tamaño 3×3. Ésta tiene sus valores propios reales y positivos. $\mathbf{x_3}$ es un vector propio de **D** asociado al valor propio de la variable λ.

Para aplicar el criterio es necesario seguir los siguientes pasos:

1. Formar la matriz D
2. Calcular los valores propios de la matriz **D**
3. Escoger el valor propio más pequeño (este valor es el que minimiza Q)
4. Calcular el vector propio ($\mathbf{x_3}$) asociado al valor propio seleccionado.

5. Normalizar el vector propio ($\mathbf{x_3}$)

6. Finalmente, calcular el vector x_9

Los elementos de P están dados por los vectores $\mathbf{x_3}$ y $\mathbf{x_9}$. El signo del vector propio, $\mathbf{x_3}$, no está definido, teniendo así dos soluciones: P y $-P$. Podemos escoger una de estas dos soluciones tomando en cuenta que la mira a calibrar se encuentra enfrente de la cámara y no atrás de ésta. Así pues $m_{34} = t_z > 0$.

III.5.5 Geometría Epipolar

La geometría epipolar proporciona la relación geométrica entre dos imágenes, ver Figura 9. Es independiente de la estructura de la escena y solo depende de los parámetros internos de la cámara y la posición relativa de la misma.

Epipolo. Es el punto de intersección de la línea que une los centros de la cámara (línea base) con el plano de la imagen.

Plano Epipolar. Es el plano que contiene la línea base, esto es, que es la unión de los centros de proyección con el punto fisico 3D. Es formado por la unión de los centros de proyección O y O' y el punto físico M.

Línea Epipolar. Es la intersección de un plano epipolar con el plano de la imagen. Todas las líneas epipolares intersectan en el epipolo. Al intersectarse el plano epipolar con cada una de las imágenes forma las líneas epipolares definiendo la correspondencia entre las líneas, ver Figura 10.

La geometría epipolar es representada por una matriz de tamaño 3×3 llamada matriz fundamental F. Si un punto M en el espacio es representado como m en la imagen izquierda, y m' en la derecha, entonces los puntos de la imagen satisfacen la relación $m'^t F m = 0$ (Hartley y Zisserman, 2001).

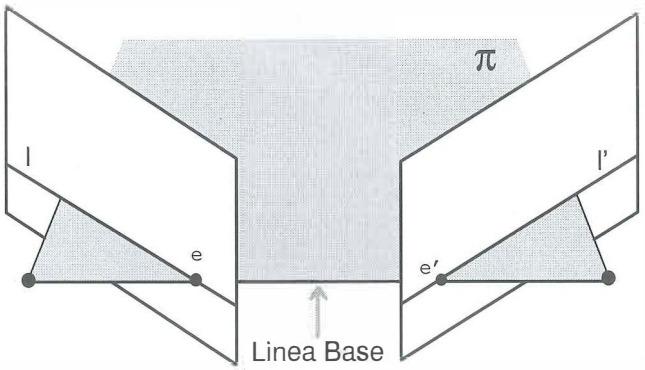


Figura 9: *Geometría Epipolar.* Cualquier plano π que intersecte el plano de la imagen en sus correspondientes líneas epipolares l y l' es un plano epipolar. La línea base intersecta cada plano de la imagen en los epipolos e y e'. Todas las líneas epipolares se intersectan en el epipolo.

Cada una de las imágenes tiene una matriz de la cámara asociada, C y C', donde C representa la matriz asociada a la imagen izquierda y C' a la imagen derecha. Un punto M en el espacio 3D es representado en la imagen como $m = Cm$ y $m' = C'm$. Los puntos m y m' corresponden entre si porque representan al mismo punto físico en el espacio 3D.

Calibración estereoscópica

En un sistema estereoscópico se utilizan dos cámaras que observan la misma escena, o bien, una sola cámara que adquiera dos fotografías desde distintos puntos de vista que implica un movimiento de la cámara. De esta forma, se obtienen dos proyecciones de la escena en el plano de las imágenes y cada cámara está asociada a un sistema coordenado. Sea M un punto en la escena y m y m' las dos proyecciones en la cámara izquierda y derecha respectivamente. Se puede escribir la recta que pasa por el centro focal C de la cámara izquierda y el punto m, así como también la recta que pasa por C' y m'. El punto M de la escena se encuentra en la intersección de estas dos rectas. A fin de calcular esta intersección y poder determinar la posición de M, es necesario

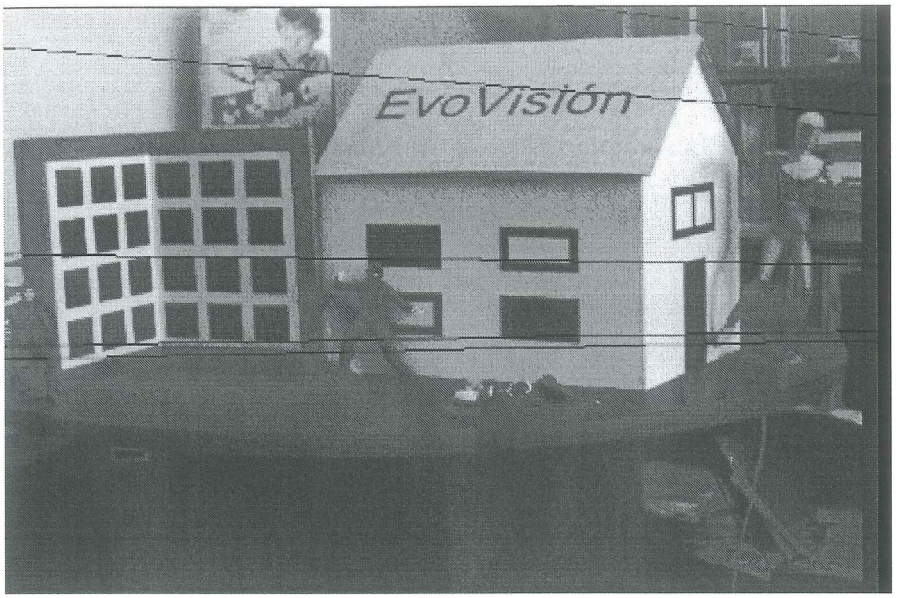


Figura 10: *línea Epipolar*. Se muestran 5 líneas epipolares en la imagen derecha para 5 puntos de la imagen izquierda. Note que el epipolo se encuentra fuera de la imagen.

expresar las dos ecuaciones de las rectas Cm y $C'm'$ en el mismo sistema coordenado.

Se mencionó anteriormente que el método Faugeras-Toscani permite determinar los elementos de la matriz de proyección P, llevando a cabo la calibración de una cámara. La calibración estereoscópica consiste en determinar la matriz de transformación A_s entre el sistema coordenado de la cámara derecha y el sistema coordenado de la cámara izquierda. La matriz A_s es del mismo tipo que la matriz A, ver Ecuación 10, la cual se compone de una rotación y una traslación. Para calibrar el sistema estereoscópico se sugiere seguir los siguientes pasos:

1. Calibrar cada cámara a partir de un sistema coordenado único, con el fin de obtener la matriz de proyección P que corresponde a la cámara izquierda y P' a la cámara derecha.

2. Extraer los parámetros intrínsecos y extrínsecos de cada una de las cámaras.

3. Calcular las matrices A y A'. La matriz A representa la transformación del sistema coordenado del mundo al sistema coordenado de la cámara izquierda (mundo-cámara). La matriz A' representa la transformación del sistema coordenado del mundo al sistema coordenado de la cámara derecha.

4. Una vez que se obtienen los valores de A y A' se calcula A_s:

$$A_s = A' A^{-1} \tag{46}$$

En forma matricial tenemos:

$$A_s = \begin{bmatrix} r_{11} & r_{12} & r_{13} & b_x \\ r_{21} & r_{22} & r_{23} & b_y \\ r_{31} & r_{32} & r_{33} & b_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r_1} & b_x \\ \mathbf{r_2} & b_y \\ \mathbf{r_3} & b_z \\ \mathbf{0} & 1 \end{bmatrix} \tag{47}$$

El vector $\mathbf{b} = (b_x b_y b_z)$ es el vector que va desde el centro de proyección C de la cámara izquierda al centro de proyección C' de la cámara derecha.

Punto de Correspondencia Geométrica

Existe una relación simple entre un punto de la imagen de la izquierda y un punto de la imagen de la derecha que puede describirse geométricamente.

Definición 2. Punto de Correspondencia Geométrica: *Sea C y C' los centros ópticos (proyección) de las cámaras izquierda y derecha. Un punto M en 3D es proyectado en los planos de ambas imágenes a los puntos m y m'. Dado un punto m_1 en el plano de la imagen izquierda, su punto correspondiente en la imagen derecha está restringido a pertenecer a un punto de la línea epipolar en la imagen derecha. De igual forma, el punto m_1' debe encontrarse en la línea epipolar de la imagen izquierda.*

Para describir la relación entre un pixel de una imagen y un pixel en la otra imagen, ver Figura 11, primero se trabaja sobre el sistema coordenado de la cámara y no en el sistema coordenado de la imagen. A partir de un pixel con coordenadas u y v,

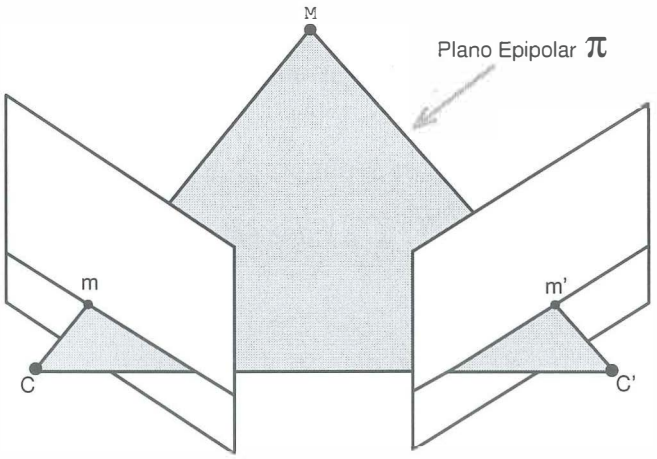


Figura 11: *Correspondencia Geométrica.* Las dos cámaras son indicadas por los centros de proyección C y C' y los planos de las dos imágenes. Los centros de proyección, el punto M, y sus correspondientes puntos proyectados en las dos imágenes m y m' están sobre un mismo plano π.

podemos obtener sus coordenadas en la cámara izquierda. La matriz C se encuentra en la Ecuación 15.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = C^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{48}$$

Para la cámara derecha se tiene una expresión similar:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = C'^{-1} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \tag{49}$$

La relación entre los dos sistemas coordenados se expresa de la siguiente manera:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = A_s \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{50}$$

donde, (X, Y, Z) representan las coordenadas en el sistema de la cámara izquierda y (X', Y', Z') las coordenadas en el sistema de la cámara derecha.

Las coordenadas de m son $(x, y, 1)$ donde $x = X/Z$ y $y = Y/Z$. De igual manera, tenemos que para m' sus coordenadas son $(x', y', 1)$. x' y y' pueden ser escritas como sigue:

$$x' = \frac{X'}{Z'} = \frac{r_{11}X + r_{12}Y + r_{13}Z + b_x}{r_{31}X + r_{32}Y + r_{33}Z + b_z} \tag{51}$$

$$y' = \frac{Y'}{Z'} = \frac{r_{21}X + r_{22}Y + r_{23}Z + b_y}{r_{31}X + r_{32}Y + r_{33}Z + b_z} \tag{52}$$

Tomando en cuenta que $X = xZ$ y que $Y = yZ$, y con la notación $\mathbf{p} = (x\ y\ 1)^t$ se puede simplificar estas fórmulas:

$$x' = \frac{Z\mathbf{r_1}.\mathbf{p} + b_x}{Z\mathbf{r_3}.\mathbf{p} + b_z} \tag{53}$$

$$y' = \frac{Z\mathbf{r_2}.\mathbf{p} + b_y}{Z\mathbf{r_3}.\mathbf{p} + b_z} \tag{54}$$

Se puede expresar entonces la posición de un punto en la imagen de la derecha en función de su punto correspondiente en la imagen de la izquierda, a partir de los parámetros del sistema y de la profundidad Z del punto M. Con estas ecuaciones se puede reducir la búsqueda del punto que corresponde en la imagen derecha haciendo uso de la restricción epipolar.

III.6 Restricciones Geométricas y Físicas

Debido a que el problema de correspondencia es un problema de búsqueda muy difícil, es necesario agregar información adicional para disminuir correspondencias falsas y reducir el tiempo de búsqueda. Para ello, algunas restricciones tanto físicas como geométricas pueden ser tomadas en cuenta.

III.6.1 Epipolar

Una de las restricciones geométricas más importantes es la restricción epipolar ya que reduce la búsqueda bidimensional a una sola dimensión. La Figura 12 muestra un ejemplo de la restricción epipolar.

Definición 3. Restricción epipolar. *Dado un punto m en la imagen izquierda, el punto correpondiente m' debe de estar sobre la línea epipolar correspondiente.*

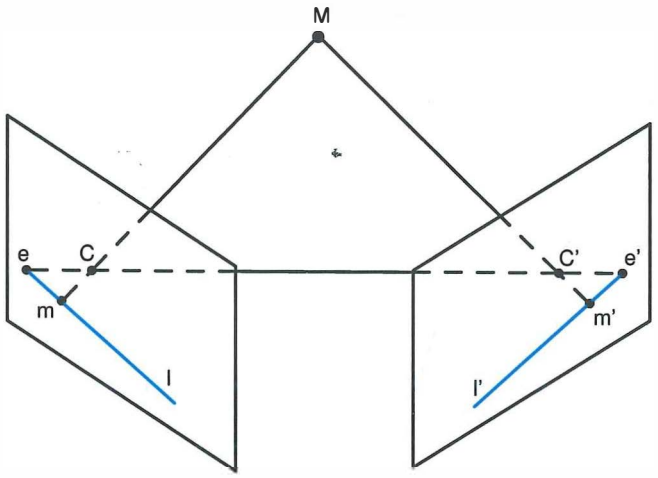


Figura 12: *Restricción Epipolar.* Esta restricción asegura que un punto 3D que es proyectado en ambos planos de la imagen como m y m' se localizan en una línea epipolar en cada imagen. El punto m' está localizado en la línea epipolar l de la imagen izquierda y viceversa.

La *línea epipolar* es descrita mediante una ecuación que describe el lugar de los puntos de la imagen derecha que pueden corresponder a un punto m de la imagen izquierda. Para cada punto de la imagen izquierda, existe una línea epipolar derecha y, a su vez, para cada punto de la imagen derecha existe una línea epipolar izquierda. La ecuación de la línea epipolar puede ser obtenida eliminando Z de las ecuaciones (53) y (54) quedando de la siguiente manera:

Para la Ecuación 53, tenemos:

$$(Z\mathbf{r_3}.\mathbf{p} + b_z)x' = Z\mathbf{r_1}.\mathbf{p} + b_x$$

$$Z\mathbf{r_3}.\mathbf{p}.x' + x'b_z = Z\mathbf{r_1}.\mathbf{p} + b_x$$

$$Z\mathbf{r_3}.\mathbf{p}.x' + x'b_z - Z\mathbf{r_1}.\mathbf{p} = b_x$$

$$Z\mathbf{r_3}.\mathbf{p}.x' - Z\mathbf{r_1}.\mathbf{p} = b_x - x'b_z$$

$$Z(\mathbf{r_3}.\mathbf{p}.x' - \mathbf{r_1}.\mathbf{p} = b_x - x'b_z)$$

$$Z = \frac{b_x - x'b_z}{\mathbf{r_3}.\mathbf{p}.x' - \mathbf{r_1}.\mathbf{p}} \tag{55}$$

Para la Ecuación 54, tenemos:

$$(Z\mathbf{r_3}.\mathbf{p} + b_z)y' = Z\mathbf{r_2}.\mathbf{p} + b_y$$

$$Z\mathbf{r_3}.\mathbf{p}.y' + y'b_z = Z\mathbf{r_2}.\mathbf{p} + b_y$$

$$Z\mathbf{r_3}.\mathbf{p}.y' + y'b_z - Z\mathbf{r_2}.\mathbf{p} = b_y$$

$$Z\mathbf{r_3}.\mathbf{p}.y' - Z\mathbf{r_2}.\mathbf{p} = b_y - y'b_z$$

$$Z(\mathbf{r_3}.\mathbf{p}.y' - \mathbf{r_2}.\mathbf{p}) = b_y - y'b_z$$

$$Z = \frac{b_y - y'b_z}{\mathbf{r_3}.\mathbf{p}.y' - \mathbf{r_2}.\mathbf{p}} \tag{56}$$

Igualando las ecuaciones (55) y (56), tenemos:

$$\frac{b_x - x^{'} b_z}{\mathbf{r_3.p}.x^{'} - \mathbf{r_1.p}} = \frac{b_y - y^{'} b_z}{\mathbf{r_3.p}.y^{'} - \mathbf{r_2.p}}$$

$$b_x - x^{'} b_z(\mathbf{r_3.p}.y^{'} - \mathbf{r_2.p}) = b_y - y^{'} b_z(\mathbf{r_3.p}.x^{'} - \mathbf{r_1.p})$$

Se continúa desarrollando, hasta llegar a la ecuación de la línea epipolar:

$$(b_z\mathbf{r_2} \cdot \mathbf{p} - b_y\mathbf{r_3} \cdot \mathbf{p})x^{'} + (b_x\mathbf{r_3} \cdot \mathbf{p} - b_z\mathbf{r_1} \cdot \mathbf{p})y^{'} = b_x\mathbf{r_2} \cdot \mathbf{p} - \mathbf{b_y r_1} \cdot \mathbf{p} \tag{57}$$

Se puede observar que la ecuación (57) es la ecuación de una recta de la forma

$$a^{'} x^{'} + b^{'} y^{'} + c^{'} = 0 \tag{58}$$

Es la ecuación de la recta expresada en el sistema coordenado de la cámara derecha. Puede ser expresada matricialmente de la siguiente forma:

$$\begin{bmatrix} x^{'} & y^{'} & 1 \end{bmatrix} \begin{bmatrix} a^{'} \\ b^{'} \\ c^{'} \end{bmatrix} = 0 \tag{59}$$

Los parámetros de esta recta $a^{'}, b^{'}, c^{'}$ están en función de los parámetros de la otra imagen en el sistema coordenado de la cámara. Las expresiones son las siguientes:

$$\begin{aligned} a^{'} &= (b_z\mathbf{r_2p} - b_y\mathbf{r_3p}) \\ b^{'} &= (b_x\mathbf{r_3p} - b_z\mathbf{r_1p}) \\ c^{'} &= (b_x\mathbf{r_2p} - b_y\mathbf{r_1p}) \end{aligned} \tag{60}$$

donde,

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{61}$$

Desarrollando lo anterior, tenemos:

$$\begin{aligned} a' &= (b_y r_{31} - b_z r_{21})x + (b_y r_{32} - b_z r_{22})y + (b_y r_{33} - b_z r_{23}) \\ b' &= (b_z r_{11} - b_z r_{31})x + (b_z r_{12} - b_x r_{32})y + (b_z r_{13} - b_x r_{33}) \\ c' &= (b_x r_{21} - b_y r_{11})x + (b_x r_{22} - b_y r_{12})y + (b_x r_{23} - b_y r_{13}) \end{aligned} \tag{62}$$

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{MatrizEsencial} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{63}$$

La relación geométrica entre dos imágenes se obtiene primero en el sistema coordenado de las cámaras (Matriz Esencial) y después en el sistema coordenado de la imagen (Matriz Fundamental).

Matriz Esencial

La matriz escencial E describe la transformación epipolar izquierda-derecha, la cual permite calcular la ecuación de una línea epipolar que pasa por la imagen derecha asociada a un punto de la imagen izquierda.

$$\mathbf{E} = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{64}$$

El producto de las dos matrices mostradas en la ecuación (64) genera la matriz escencial. Esta matriz puede ser calculada a partir de los parámetros b_x, b_y, b_z y $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}$. Estos parámetros son obtenidos durante el proceso de calibración del sistema estereoscópico.

Los parámetros a', b', c' pueden ser representados en forma matricial como sigue:

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{65}$$

La Ecuación (65) puede ser expresada de forma compacta de la siguiente manera:

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \mathbf{E} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{66}$$

La Ecuación (66) representa la transformación epipolar en la cual un punto de la imagen izquierda $(x\ y\ 1)^t$, le corresponde una recta sobre la imagen derecha descrita por los parámetros $(a'b'c')$. Sustituyendo (66) en (58) tenemos:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \mathbf{E} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

siendo,

$$\mathbf{p}'^{\mathbf{t}}\mathbf{E}\mathbf{p} = \mathbf{0} \tag{67}$$

Las ecuaciones que describen la relación geométrica entre un par de imágenes dependen del proceso de búsqueda si es de izquierda-derecha o de derecha-izquierda. La matriz **E** describe la transformación epipolar izquierda-derecha, ver Ecuación 67, la cual permite calcular la ecuación de la línea epipolar que pasa por la imagen derecha asociada a un punto de la imagen izquierda. Los puntos p y p' se encuentran en el sistema coordenado de la cámara. La transformación epipolar de derecha-izquierda esta dada por su transpuesta:

$$\mathbf{p}^{\mathbf{t}}\mathbf{E}^{\mathbf{t}}\mathbf{p}' = \mathbf{0} \tag{68}$$

BIBLIOTECA
CICESE

Matriz Fundamental

La matriz fundamental F es la representación algebraica de la geometría epipolar (Hartley y Zisserman, 2001) en el sistema coordenado de las imágenes. Un punto m es mapeado en la línea epipolar l' definido por F. La matriz fundamental satisface la condición que para cualquier par de puntos correspondientes $m \leftrightarrow m'$ en las dos imágenes:

$$\mathbf{m'^t F m = 0} \tag{69}$$

Esto es cierto, porque si los puntos m y m' corresponden, entonces m' se encuentra sobre la línea epipolar $l' = Fm$ correspondiendo al punto m. En otras palabras, $0 = m'^t l' = m'^t F m$. Si los puntos de la imagen satisfacen la relación $m'^t F m = 0$. entonces los rayos definidos por estos puntos son coplanares[3]. Ésta es una condición necesaria para los puntos que corresponden en las imágenes.

Los parámetros de la matriz fundamental pueden ser obtenidos a través de la calibración, obteniendo las matrices de proyección, o bien, puede ser calculada por lo menos con 7 puntos que correspondan. En este caso, hay un algoritmo muy conocido para calcular la matriz fundamental llamado el algoritmo de ocho puntos de Hartley (1997).

En este trabajo se usa un sistema calibrado, por lo tanto, se obtendrá F a partir de los parámetros intrínsecos y extrínsecos. A continuación, se obtiene F.

Las matrices C y C' describen los parámetros intrínsecos y extrínsecos de las dos cámaras. Al sustituir en la Ecuación (67) tenemos:

$$m'^t (C'^{-1})^t E C^{-1} m = 0$$

[3]Que se encuentran en el mismo plano

Necesitamos pasar (67) a coordenadas de las imágenes. Para ello, usamos (48) en la forma $m = Cp$ y $m' = C'p'$ que se introduce en (67). Tenemos como resultado, la matriz fundamental F definida como sigue:

$$F = (C'^{-1})^t E C^{-1} \tag{70}$$

III.6.2 Orientación

La línea epipolar además de su posición en la imagen mantiene una orientación dentro de la misma. La línea epipolar, en este caso puede ser vista como un vector director asociado al punto en la imagen (primitiva). Generalmente, se utiliza en los contornos de los objetos en la imagen, ver Figura 13. La restricción de orientación es utilizada en este trabajo de investigación usando la inclinación de la linea epipolar de un punto en la imagen. Conocemos la ecuación de la recta definida en (58), por lo tanto, es posible obtener la pendiente de la recta $pr = tan\theta = -a'/b'$ y posteriormente obtener sus grados despejando θ.

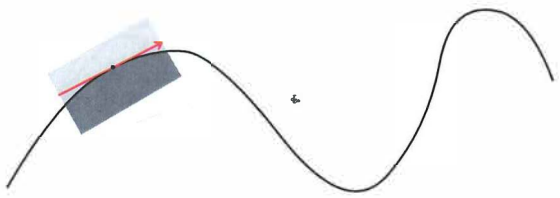


Figura 13: *Restricción de Orientación.* Un punto del contorno está definido por su posición en la imagen, y la dirección de la tangente al contorno en este punto.

III.6.3 Orden

Sea M_1 un punto y sean m_1 y m_1' sus proyecciones en las dos imágenes. Las direcciones de proyección de este punto dividen el plano epipolar en dos regiones, una región interior

y una región exterior, ver Figura 14. Sean M_2 y M_3 dos puntos en el sistema coordenado del mundo y sean m_2, m_3 y $m_2^{'}, m_3^{'}$ sus proyecciones respectivas en la imagen izquierda y derecha respectivamente. El punto M_2 se encuentra dentro de la región interior y el punto M_3 se encuentra en la región exterior. Observemos el orden de proyección de estos puntos en las dos imágenes.

- En la imagen izquierda el orden es m_3, m_1, m_2, ver Figura 14.
- En la imagen derecha el orden es $m_3^{'}, m_2^{'}, m_1^{'}$, ver Figura 14.

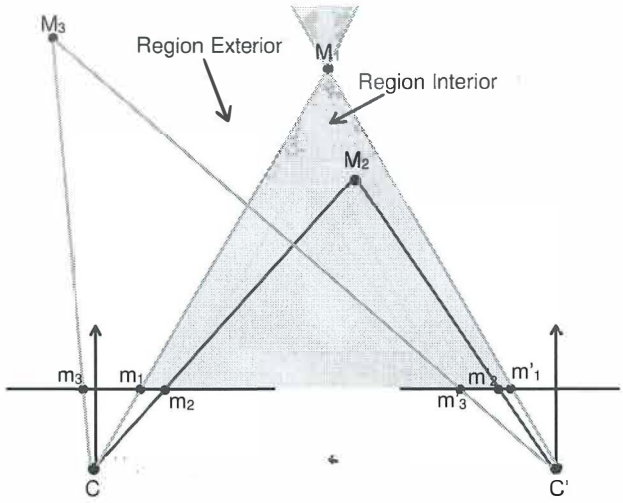


Figura 14: *Restricción de Orden.* M_1 y M_3 se proyectan en el mismo orden en las dos imágenes mientras que el orden de las proyecciones M_1 y M_2 es inverso, dado que el punto M_2 se encuentra dentro de la zona prohibida (zona interior)

Si se toma M_1 como referencia, las proyecciones de M_3 aparecen en el mismo orden en las dos imágenes mientras que el orden de proyección de M_2 se invierte, ver Figura 14.

Supongamos ahora que M_1 y M_3 se encuentran sobre la superficie de un objeto, ver Figura 15. El hecho que M_3 se encuentre en la región exterior de M_1 es debido a que el objeto está mínimamente inclinado en relación al plano de las imágenes.

Sea ahora el objeto 2 de la misma Figura 15 sobre el cual se encuentran los puntos M_1 y M_2. El punto M_2 se encuentra en la región interior de M_1 lo que significa que el objeto 2 está muy inclinado en relación con el plano de la imágen. Si este objeto es opaco, las proyecciones de dos puntos no pueden verse en las dos imágenes. Si el objeto es transparente, las proyecciones se verán en las dos imágenes pero en un orden inverso.

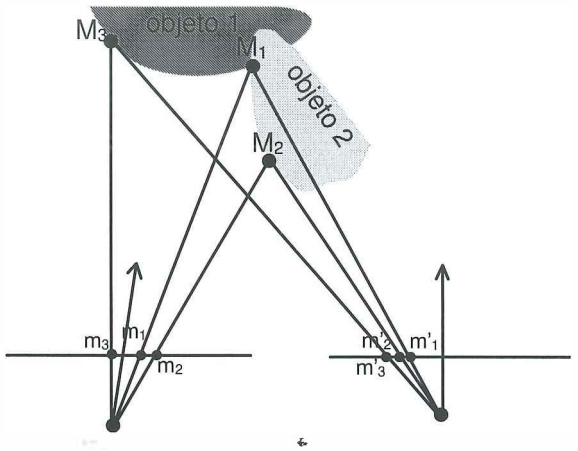


Figura 15: *Restricción de Orden.* Cuando dos puntos se encuentran sobre la superficie de un objeto que está poco inclinado como el caso del objeto 1, el orden de sus proyecciones es el mismo en las dos imágenes. Si el objeto se encuentra muy inclinado como el objeto 2, las proyecciones no se ven simultáneamente en las dos imágenes si el objeto es opaco, pero si el objeto es transparente, el orden se invierte.

III.6.4 Unicidad

Consideremos ahora, un caso extremo de la restricción de orden. El punto M_2 se encuentra en la frontera de las regiones exterior e interior del punto M_1, ver Figura

16. Los dos puntos se proyectan en un mismo punto en la imagen izquierda y en dos puntos diferentes en la imagen derecha. Si se designa por m la proyección común de M_1 y de M_2 en la imagen izquierda y por $m_1^{'}$ y por $m_2^{'}$ sus proyecciones en la imagen derecha, se deben tener al mismo tiempo los apareamientos $(m, m_1^{'})$ y $(m, m_2^{'})$. Aceptar que estos dos apareamientos sean compatibles equivale a aumentar la dificultad de la correspondencia ya que, un punto de una imagen debería tener un único punto que le corresponda, ver Figura 16a .

Si consideramos objetos opacos en la escena, entonces un punto en la imagen izquierda debe tener a lo más un correspondiente en la imagen derecha, ver Figura 16b. Es decir, la correspondencia debe ser única entre dos imágenes. Esta restricción falla si existen objetos transparentes en la escena, en tal caso, se produciría una oclusión.

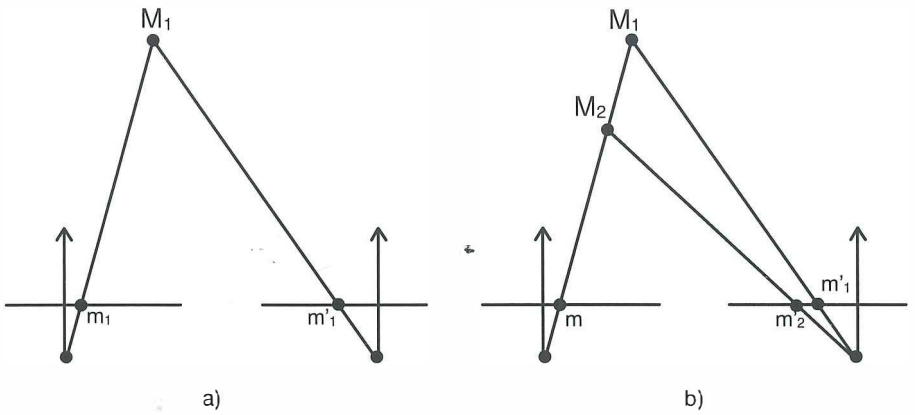


Figura 16: *Restricción de unicidad.* *a*) Para el punto m_1 existe solo un punto $m_1^{'}$ que corresponde en la otra imagen. En este caso, la restricción de unicidad se cumple. *b*) La restricción de unicidad no se respeta cuando dos puntos se proyectan en un mismo punto en una imagen y en dos puntos diferentes en la otra imagen. En este caso, M_1 y M_2 se proyectan en la imagen izquierda en un único punto m mientras que en la imagen derecha se proyecta en los dos puntos que corresponden $m_1^{'}$ y $m_2^{'}$.

III.6.5 Continuidad de la Figura

La restricción de continuidad de la figura es una restricción sencilla y eficaz cuando los puntos que van a ponerse en correspondencia se encuentran a lo largo de contornos. La Figura 17 muestra dos imágenes y puntos a lo largo de un contorno que pertenecen a estas imágenes. Las líneas epipolares son tambien representadas sobre esta figura, las cuales están rectificadas (líneas horizontales). Los puntos que pertenecen a un contorno de la imagen de la imagen izquierda se ponen en correspondencia con los puntos de la imagen derecha que no necesariamente pertenecen a un mismo contorno. Este tipo de error puede corregirse fácilmente asignando los puntos que faltan del contorno de la imagen izquierda al contorno que contenga la mayoría de los puntos en la imagen derecha. Esta restricción debe considerarse en un contexto global ya que puede haber casos que no se cumplan, por ejemplo en la Figura 17.

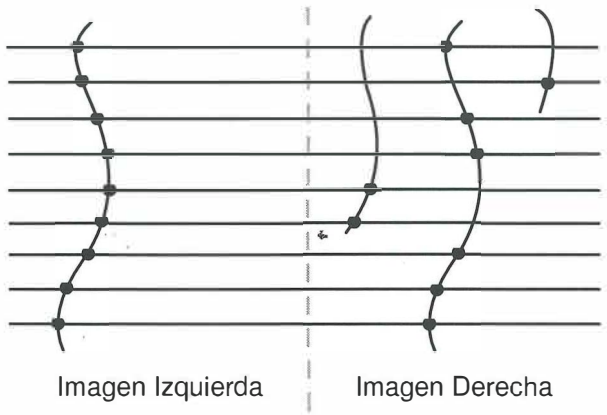


Figura 17: *Restricción de Continuidad de la figura.* Los puntos de la imagen izquierda no son bien correspondidos en la imagen derecha debido a la existencia de contornos parecidos en una sola región.

III.6.6 Similaridad

La restricción de similaridad toma en cuenta los valores de intensidad de los pixeles. Al usar un enfoque basado en intensidad o en áreas, los pixeles que correspondan en ambas imágenes tendrán valores de intensidad muy similares. Así mismo, cuando se usa la función de correlación, las ventanas tendrán que ser altamente correlacionadas. Para el enfoque basado en características, las características que correspondencia deben tener valores de atributos similares.

III.6.7 Relajación

La relajación es una restricción usada globalmente en la correspondencia para eliminar apareamientos falsos. El proceso de eliminar apareamientos falsos consiste en asignar un valor de peso (probabilístico) a cada pixel que se va a corresponder. El valor de peso cambia iterativamente conforme la correspondencia se va llevando a cabo, y los pixeles que tengan asignado un valor de probabilidad por debajo de un umbral se eliminan.

III.6.8 Suavidad

La restricción de suavidad es usada de manera global y está muy relacionada al flujo óptico. Ha sido usada en trabajos de investigación donde se pretende calcular la velocidad del movimiento de superficies de una secuencia de mapas de disparidad[4] . Las regiones locales del mapa de disparidad deben ser relativamente lisas independientemente de las regiones con oclusión o de la discontinuidad (Sun, 2002).

[4]Imagen que muestra la profundidad de los objetos a través de niveles de gris

III.7 Conclusiones

En este capítulo se detallan aspectos geométricos relacionados al problema de correspondencia, así como también sobre el proceso de formación de las imágenes ya que de ahí se deriva cierta información que es de vital importancia para llevar a cabo la correspondencia en un sistema calibrado.

Es importante mencionar, que al trabajar sobre el par estereoscópico se involucra el uso de tres sistemas coordenados. El primero, es el sistema coordenado del mundo o de la escena, ya que primero ocurre una transformación mundo-cámara en el proceso de adquisición de la imagen. El segundo, es el sistema coordenado de la cámara ya que ocurre una transformación cuando la escena se proyecta a través de la cámara. El tercero y último, es el sistema coordenado de la imagen, en el cual se pierde una dimensión cuando ocurre la transformación cámara-imagen dado que la imagen es bidimensional con las coordenadas u y v.

Un concepto geométrico de mayor relevancia es el de la geometría epipolar, ya que nos permite conocer la relación geométrica entre dos imágenes. El uso de la línea epipolar reduce el problema de búsqueda a una sola dimensión. Por ello, es ampliamente usada en este tipo de problemas, ya que la búsqueda del punto se hace solamente a lo largo de la línea en la imagen derecha. El proceso de calibración, las transformaciones entre un sistema coordenado y otro se realizan con el fin de conocer la matriz fundamental y poder utilizar la geometría epipolar.

El uso de la geometría epipolar ayuda en gran medida en el problema. Sin embargo, no es suficiente debido a que el problema tiene ambigüedades inherentes, por lo que es necesario el uso de información adicional como la restricción epipolar, orientación, orden, unicidad, continuidad de la figura, similaridad, relajación, suavidad, entre otras.

Nos referimos a problemas inherentes cuando las imágenes cuentan con oclusiones, patrones repetitivos, baja textura, falta de información por el campo de visión, entre otros. El uso de las restricciones geométricas y físicas auxilian en el proceso para identificar este tipo de problemas dentro de las imágenes. Las restricciones que se usan en este trabajo es la restricción epipolar, de orientación y similaridad.

Capíulo IV

El Algoritmo de Infección: Un Nuevo Enfoque

IV.1 Conceptualización sobre Vida Artificial

La vida artificial o *A-life*[1], estudia la vida haciendo uso de medios teóricos y computacionales para crear sistemas complejos los cuales simulan el comportamiento de seres vivos o de algún fenómeno natural. La estructura y comportamiento de estos sistemas no proviene de la inteligencia de cada una de sus partes, sino de su interacción.

El comportamiento de un sistema inspirado en la naturaleza puede incluir nacimiento y muerte, crecimiento, selección natural, evolución, aprendizaje e inteligencia.

Christopher Lagton, pionero y fundador de la primera conferencia de *A-life* Steven (1993) define a la vida artificial como sigue: *La vida artificial se suele definir como la ciencia que trata de situar la vida "tal como es" dentro del contexto de la vida "tal como podría ser".*

Los científicos de *A-life* estudian el significado de cómo los sistemas vivientes actuales pueden ser generados, evolucionados y observados. El juego de la vida es uno de los ejemplos más ilustrativos sobre vida artificial. Este juego trata de simular organismos vivientes, cómo se van reproduciendo y muriendo en el transcurso del tiempo. Este juego llegó a ser el ejemplo más famoso del uso del autómata celular inventado por Von

[1] *En inglés, Artificial Life*

Neumann. El juego de la vida fue creado por John Conway (Steven, 1993) y fue la inspiración para una generación de investigadores de *A-life*.

IV.2 Herramientas

Las herramientas informáticas usadas en vida artificial son principalmente:

- **Redes Neuronales**. Las Redes Neuronales Artificiales[2] fueron originalmente una simulación del sistema nervioso biológico, el cual está formado por un conjunto de unidades llamadas *neuronas* o *nodos* conectadas unas con otras.

  McCulloch y Pitts (1943) propusieron el primer modelo de red neuronal en términos de un modelo computacional. Este primer modelo sirvió de ejemplo para modelos posteriores. Las redes neuronales tienen distintas aplicaciones ya que son utilizadas en la minería de datos, reconocimiento de patrones y los sistemas de control, por mencionar algunos ejemplos. Las redes neuronales constituyen una parte muy importante en el estudio y desarrollo de la inteligencia artificial y ahora en la vida artificial.

- **Algoritmos Genéticos (AGs)**. Los AGs tratan de resolver problemas de optimización imitando la selección natural mediante el uso de operadores genéticos como la mutación, cruzamiento y selección. Fueron descritos por John Holland en los años 60s, conocido como el padre de los algoritmos genéticos. El algoritmo trabaja a través de cromosomas que representan organismos o soluciones candidatas al problema. Cada cromosoma consiste de un cojunto de genes representado por una cadena de bits. Un ejemplo de cómo trabaja un AG es:

[2]En inglés Artificial Neural Networks

1. Se comienza generando aleatoriamente la población de cromosomas, por ejemplo, las soluciones candidatas al problema.

2. Calcular la función objetivo para cada cromosoma en la población.

3. Después, aplicar los operadores genéticos a la población y seleccionar los mejores para crear una nueva población. Las peores soluciones se van desechando en el proceso.

4. Se repite el proceso desde el punto 2.

Los GAs pueden ser aplicados a distintos problemas, tanto científicos como de ingeniería. Por ejemplo: en la optimización, programación dinámica, aprendizaje de robots o máquinas, modelos económicos, modelos de sistemas inmunes, modelos ecológicos (Mitchell y Forrest, 1995), entre otros.

- **Autómatas Celulares**. Debido a su importancia en esta tesis, los autómatas celulares se explican a detalle en la sección IV.3.

- **Lógica Borrosa o Difusa**. Ordena un razonamiento basado en reglas imprecisas y con datos incompletos. Es una rama de la inteligencia artificial que se basa fundamentalmente en los grados de verdad ya que existen hechos en el mundo que no necesariamente son completamente ciertos o falsos.

  Los esquemas de razonamiento aproximado, usados en la lógica difusa intentan reproducir el proceso de razonamiento del cerebro humano. Estos esquemas consisten en una generalización de los esquemas básicos de inferencia en Lógica Binaria (cierto-falso). Algunas aplicaciones de la lógica borrosa o difusa pueden encontrarse en los verificadores de ortografía, en el funcionamiento de electrodomésticos, en la automatización, en sistemas expertos, entre otros.

IV.3 Autómatas Celulares

Los autómatas celulares (AC) modelan situaciones físicas y obtienen la solución de un problema. Pueden ser usados para simular problemas físicos como un incendio forestal. Los AC's consisten de un enrejado uniforme n-dimensional, un vecindario y un conjunto de reglas, usados en un tiempo t, ver Figura 18 y IV.4.3, (Chaudhuri *et al.*, 1997).

Cada celda del enrejado o rejilla es llamada **célula**. El valor de la célula representa esencialmente un estado finito. El estado de la celda representa un valor numérico entero y va cambiando en relación a un conjunto de reglas. La configuración de la rejilla cambia en pasos de tiempo discreto.

Cada célula almacena información que comúnmente se conoce como su **estado**, y en cada paso de tiempo verifica las celdas alrededor de ésta que se encuentran en el vecindario y posteriormente se consulta la tabla de reglas para determinar su estado en el próximo paso de tiempo.

El autómata celular fue introducido por Von Neumann y Stanislaw Ulam alrededor de los años 40s. La idea de Von Neumann se consideró como un organismo viviente utilizando un autómata.

Los autómatas celulares son considerados sistemas dinámicos discretos que involucran reglas simples determinísticas que rigen su comportamiento en cada paso de tiempo; de esta forma se **autoreproduce**. Las propiedades principales que tiene un AC es la rejilla n-dimensional y un comportamiento dinámico, descrito por las reglas. Estas reglas describen el estado de una célula en el próximo paso de tiempo dependiendo de los estados de las células vecinas. Una colección de células podría ser vista como un **organismo**.

A finales de los 60s fue cuando John Conway creó el juego de la vida (Steven, 1993) con el fin de confirmar si era verdad que el autómata de Neumann se calificaba como una computadora universal, esto es, que podría emular cualquier función descifrable de cualquier otra computadora por el uso de un conjunto de reglas lógicas. El organismo por si mismo fué frustrantemente complejo, con sus doscientas mil celdas y veintinueve estados.

Conway tenía la idea que un autómata celular con capacidades de computación universales podría ser mucho más sencillo, asi que simplificó grandemente el trabajo del maestro Neumann. La clave fue el uso de reglas que dictaban la sobrevivencia, nacimiento y muerte. Las reglas en este caso, generaban todo en este mundo artificial, utilizando solamente dos estados, prendido-apagado, uno-cero ó vivo-muerto.

El autómata celular se eligió como herramienta global para el proceso de correspondencia porque permite que la información de la imagen sea fácilmente manipulada a través de la rejilla. De esa forma, se puede llevar un control de los pixeles que pueden ser asignados automáticamente, sin necesidad de realizar los cálculos correspondientes.

IV.3.1 Tipos de Vecindario

Cada una de las células por sí solas representan un estado numérico. Para introducir la parte dinámica en el sistema, se agrega el conjunto de reglas que actuarán en relación a los estados de las células vecinas y la célula central del vecindario.

El vecindario tiene distintas estructuras y es usado de acuerdo a la dimensión del autómata. Por ejemplo, para un autómata 1-D se utiliza un vecindario de tres celdas, ver Figura 20a. Para un autómata 2-D los vecindarios comúnmente utilizados son el vecindario de Von Neumann (Figura 20b), Moore (Figura 20c) y Moore Extendido (Figura 20d).

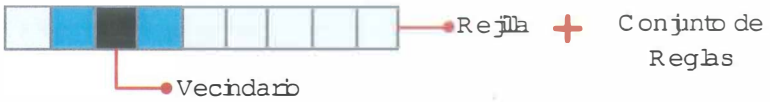


Figura 18: Autómata unidimensional. La estructura general de un autómata unidimensional consiste de una rejilla de tamaño $1 \times n$.

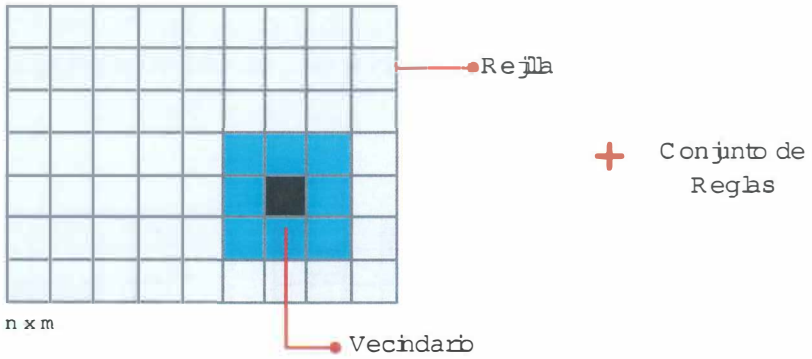


Figura 19: Autómata bidimensional. La estructura general de un autómata bidimensional consiste de una rejilla de tamaño $2 \times n$.

El tipo de vecindario usado en esta tesis mantiene una estructura diferente a la existente bibliográficamente. Ésta se divide en 9 vecinos cercanos y 16 vecinos externos. El vecindario cercano, ver Figura 21a es un vecindario de Moore y el vecindario externo, ver Figura 21b se alinea a las células del vecindario de Moore. La razón de crear una estructura nueva fue principalmente para tomar en cuenta las direcciones de la línea epipolar, donde se hace la búsqueda de la correspondencia. Lo anterior es debido a que la búsqueda de los pixeles de la imagen izquierda se realiza sobre la línea epipolar de la imagen derecha.

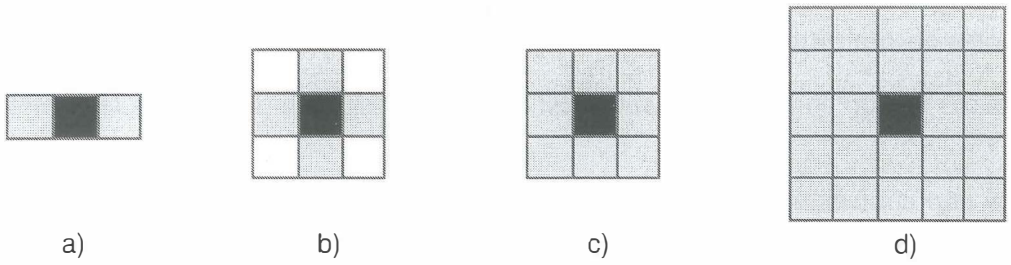


Figura 20: *Estructuras de Vecindarios. a)* Vecindario de tres células para un autómata unidimensional. Para un autómata bidimensional: *b)* Vecindario de Von Neumann con 4 células, *c)* Vecindario de Moore con 9 células y *d)* Vecindario de Moore Extendido con 25 células.

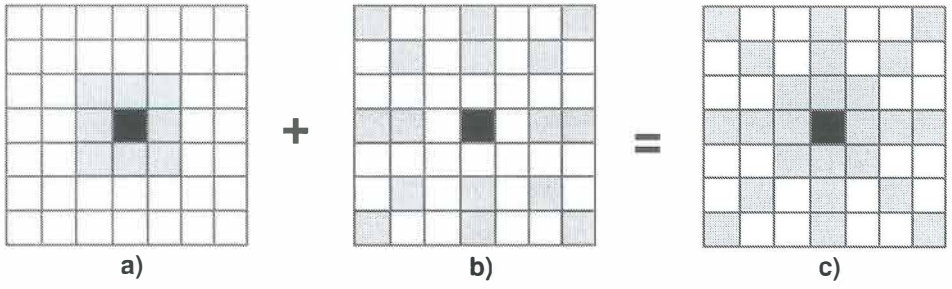


Figura 21: *Vecindario usado en esta tesis. a)* Las nueve celdas centrales forman un vecindario de Moore, *b)* El resto de celdas se alinean desde la célula central para formar el vecindario completo. *c)* Estructura del vecindario usado en la tesis con un total de 25 vecinos.

IV.4 Complejidad de las reglas

Un conjunto de reglas permite que el comportamiento del autómata sea dinámico. Lo que hace una regla en particular, es modificar el estado de la célula central a otro estado. Estas posibles reglas son extraídas de un conjunto muy grande que es determinado mediante el número de vecinos que son tomados en cuenta en el vecindario (Sipper, 1997; Chaudhuri *et al.*, 1997). Establecer el conjunto de reglas más adecuadas para un problema en específico es algo muy complejo ya que a medida que el número de estados y el número de vecinos aumenta, el conjunto de reglas crece exponencialmente, tal

como se muestra más adelante. Esta complejidad depende del número de estados que se utilice en el autómata, así como también del número de vecinos que se consideren.

Sea R el número de estados que se utilizan y k el número de vecinos que se toman en cuenta, entonces se tiene que el conjunto de configuraciones distintas del vecindario es de tamaño R^k y número de mapeos distintos para todas esas configuraciones está dado por R^{R^k}.

A continuación se muestra el ejemplo más sencillo de la complejidad del número de reglas posibles a usar para un autómata 1-D, 2-D y finalmente, nuestro caso.

IV.4.1 Caso más sencillo: Autómata 1-D

La Figura 22 ejemplifica un autómata unidimensional con estados binarios (0,1) y un vecindario de tres celdas (Izquierda-Central-Derecha).

Figura 22: Autómata unidimensional.

Número de estados $R = 2$.

Vecindario de tres celdas $k = 3$.

Por lo tanto, se tendrían $2^3 = 8$ configuraciones posibles para un vecindario de tres celdas utilizando únicamente dos estados. En la Figura 23 se muestran las 8 configuraciones distintas para este caso particular. Estas 8 configuraciones indican todas las posibles combinaciones que puede tener la estructura de la regla considerando en este caso, el vecino izquierdo, el vecino derecho y la celda central. En ocasiones,

el estado de la celda central no se toma en cuenta, eso dependerá de la estructura del vecindario y cómo esté diseñada la regla.

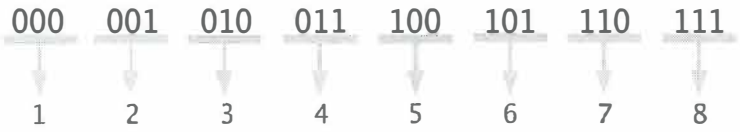


Figura 23: $2^3 = 8$ Configuraciones distintas de reglas.

Por otro lado, se tienen $2^{2^3} = 256$ mapeos distintos para esas configuraciones de reglas. Los mapeos indican la acción que va a tomar la regla para una configuración en particular. En este caso, se están manejando estados binarios, lo cual indica que cada una de las reglas puede tomar dos acciones, 1 ó 0. Es por eso que tenemos 2^{2^3} mapeos dado que existen 2^3 configuraciones distintas y cada una de ellas puede cambiar el estado de la celda central a 0 ó 1 según sea el caso. El orden de las reglas es importante en su aplicación porque determinan el comportamiento del autómata.

En la Figura 24 se muestra el origen de los 256 mapeos distintos que se tienen para las 8 configuraciones. Los mapeos se muestran organizados por conjuntos para que pueda apreciarse la diferencia.

IV.4.2 Caso del Autómata 2-D

Número de Estados $R = 2$.

Vecindario de nueve celdas $k = 9$.

Figura 24: $2^{2^3} = 256$ Mapeos distintos para 8 configuraciones.

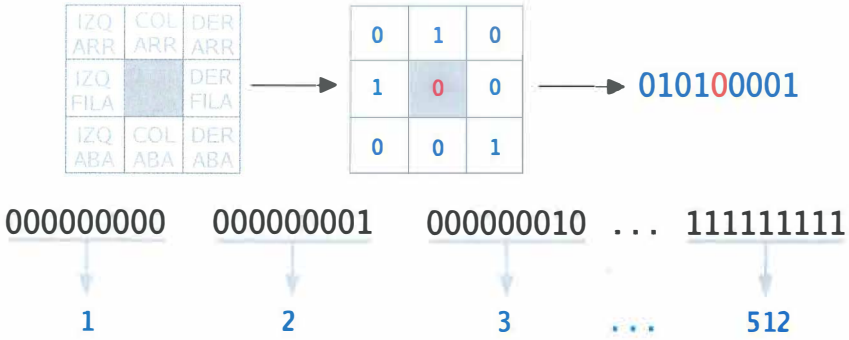


Figura 25: Configuración de las reglas utilizando un vecindario de Moore con estados binarios ($2^9 = 512$).

En el caso de un autómata 2-D utilizando estados binarios y un vecindario de Moore, se tienen $2^9 = 512$ configuraciones de reglas distintas. En la Figura 25 se aprecian las configuraciones de las reglas y las posiciones de los estados en el vecindario de Moore. Los mapeos para tales configuraciones está dado por $2^{2^9} = 1.3408 \times 10^{154}$. En la Figura 26 se muestran los mapeos para las 512 reglas.

#	REGLA	ACCIÓN	#	REGLA	ACCIÓN		#	REGLA	ACCIÓN
1-	000000000	----> 0	1-	000000000	----> 1		1-	000000000	----> 1
2-	000000001	----> 0	2-	000000001	----> 0		2-	000000001	----> 1
3-	000000010	----> 0	3-	000000010	----> 0		3-	000000010	----> 1
4-	000000011	----> 0	4-	000000011	----> 0		4-	000000011	----> 1
5-	000000100	----> 0	5-	000000100	----> 0		5-	000000100	----> 1
6-	000000101	----> 0	6-	000000101	----> 0	. . .	6-	000000101	----> 1
	.			.				.	
	.			.				.	
	.			.				.	
512-	111111111	----> 0	512-	111111111	----> 0		512-	111111111	----> 1
	1			2				2.6187e+151	

Figura 26: Ejemplo de los mapeos de las reglas con $k = 9$ y $R = 2$. Se tienen 512 configuraciones y $2^{2^9} = 1.3408 \times 10^{154}$ mapeos distintos para esas configuraciones.

IV.4.3 Caso del Autómata 2-D: Nuestro Caso

Número de Estados $R = 4$.

Vecindario de veinte y cinco celdas $k = 25$.

Nuestro trabajo de investigación hace uso de 4 estados distintos acorde al problema que se está tratando. El vecindario que se ha diseñado es de 25 vecinos tal como se puede ver en la Figura 21. Entre mayor sea el número de estados y el número de vecinos que se usen en el autómata, mayor dificultad habrá para identificar el conjunto de reglas que mejor se adapten al problema. A medida que k y R aumentan, la cantidad de mapeos aumenta exponencialmente.

En nuestro caso, tenemos que el número de configuraciones posibles de reglas está dado por: $4^{25} = 1.1259 \times 10^{15}$ y el número de mapeos para esas configuraciones está dado por: $4^{4^{25}}$.

El número de configuraciones es exponencial y es muy grande el número de mapeos

que pueden hacerse con ese número de configuraciones. Debido a que el número de configuraciones es muy grande, se ideó el uso de un identificador que pudiera usarse como comodín en el diseño de las reglas. De esa forma, los estados de la configuración pueden omitirse cuando en ese momento la información del estado no sea relevante. Por tal motivo, el comodín representa cualquier valor de estado que pudiera tener la célula dentro de la rejilla.

Los estados en esta tesis se describen numéricamente del 0 al 3. La Figura 27 muestra la representación de cada uno de los estados y la variable usada como comodín. El comodín es representado numéricamente con el número 5 y puede ser identificado con el color gris o un carácter al igual que los cuatro estados anteriores. Las diferentes representaciones de los estados es con el fin de comprender mejor el funcionamiento de las reglas y la formalización del algoritmo.

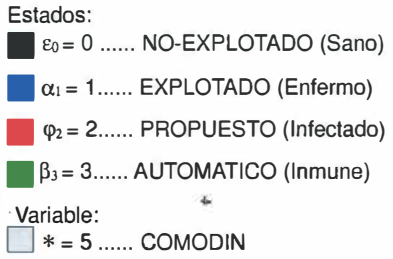


Figura 27: Estados utilizados en el autómata

El estado NO-EXPLOTADO (*Sano*) representa el pixel que no ha sido evaluado, asignado automáticamente, ni propuesto.

El estado EXPLOTADO (*Enfermo*) corresponde al caso en que un pixel de la imagen izquierda ha efectuado los cálculos necesarios para identificar su correspondiente pixel en la imagen derecha.

El estado PROPUESTO (*Infectado*) es usado para marcar el pixel cuando no se tiene la seguridad de poder asignar automáticamente su correspondiente en la imagen derecha. Una vez que es marcado se podrá decidir posteriormente si habrá evaluación, o bien, asignación automática.

El estado AUTOMÁTICO (*Inmune*) representa el caso que a un pixel se le ha sido asignado automáticamente su correspondiente en la imagen derecha sin efectuar ningún cálculo.

IV.5 La Correlación como Función Objetivo Local

La función de correlación lineal puede ser usada como un criterio de similitud entre ventanas de imágenes de tamaño fijo. El par estéreo de imágenes es usado como entrada de datos. El proceso de correlación puede verse como un proceso de búsqueda con el cual se puede identificar los pixeles correspondientes en ambas imágenes. Este proceso pretende maximizar el criterio de similitud dentro de una región de búsqueda. Sean m y m' los pixeles en la imagen izquierda y la imagen derecha, $2n+1$ el ancho de la ventana de correlación medido en pixeles, $R(m')$ la región de búsqueda en la imagen derecha asociada con m, y $\phi(I_l, I_r)$ una función de las ventanas de ambas imágenes. La función ϕ es llamada Correlación Cruzada Normalizada de Media Cero[3] para empatar el contenido de ambas imágenes, ver Figura 28. Los valores de intensidad de los pixeles de $I_r(x'+i, y'+j)$ y $I_r(x', y')$ son obtenidos en $R(m')$. Los valores de $I_l(x+i, y+j)$ y $I_l(x, y)$ son obtenidos de la ventana de correlación asociada al punto m en la imagen izquierda.

[3]Del inglés Zero-Mean Normalized Cross Correlation(ZNCC)

Si $\phi = 0$ no hay correlación lineal.

Si $\phi = 1$ se tiene correlación perfecta positiva.

Si ϕ =-1 se tiene correlación perfecta negativa.

$$\phi(I_l, I_r) = \frac{\sum_{i=-n}^{n}\sum_{j=-n}^{n}[(I_l(x+i,y+j) - \overline{I_l(x,y)})(I_r(x'+i,y'+j) - \overline{I_r(x',y')})]}{\sqrt{\sum\sum(I_l(x+i,y+j) - \overline{I_l(x,y)})^2 \sum\sum(I_r(x'+i,y'+j) - \overline{I_r(x',y')})^2}} \tag{71}$$

donde:

$I_l(x+i, y+j)$	Representa el valor de intensidad del pixel en la posición $(x+i, y+j)$ de la imagen izquierda.
$\overline{I_l(x,y)}$	Representa el valor de intensidad promedio de los pixeles que se encuentran en la ventana de correlación de la imagen izquierda.
$I_r(x'+i, y'+j)$	Representa el valor de intensidad del pixel en la posición $(x'+i, y'+j)$ de la imagen derecha.
$\overline{I_r(x',y')}$	Representa el valor de intensidad promedio de los pixeles que se encuentran en la ventana de correlación de la imagen derecha.

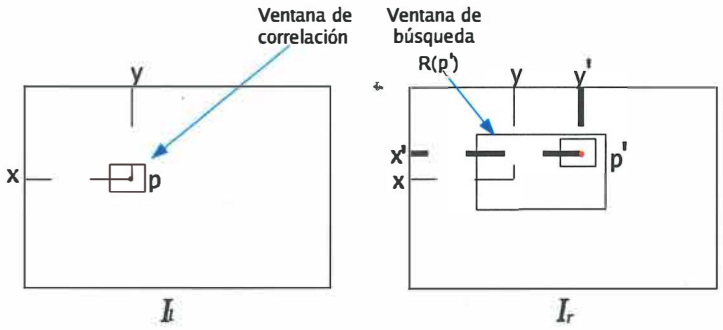


Figura 28: Coeficiente de correlación del punto m de la imagen izquierda con el punto m' de la imagen derecha. Esquema que representa el método para un algoritmo de búsqueda exhaustiva utilizando la correlación como función objetivo.

La ventana de correlación está centrada en (x, y) y (x', y') de la imagen izquierda y derecha respectivamente. A diferencia de los primeros algoritmos de búsqueda exhaustiva tradicionales, la correlación se lleva a cabo solamente a lo largo de la línea epipolar asociada a m' que se encuentra dentro de la ventana de búsqueda de la imagen derecha y no sobre toda la ventana de búsqueda.

IV.6 Epidemia artificial para el problema de correspondencia estéreo

La motivación principal para usar lo que llamamos el algoritmo de infección se basa en el hecho de que cuando observamos una escena enfocamos nuestra atención a ciertas partes que de algún modo mantienen nuestro interés en la escena. No es necesario analizar cada parte de la escena a detalle, por lo que la mayoría del proceso de atención se realiza *"adivinando"* partes de la escena, mientras que estamos descubriendo nuestras actividades.

Este proceso de "*adivinar*" información basada en regiones locales puede ser implementada a través del proceso de propagación de la correspondencia. Nuestra idea consiste en realizar el modelado del problema como una enfermedad.

La invasión de una enfermedad en una población es un fenómeno presente en la naturaleza. Este fenómeno puede ser visto en poblaciones pertenecientes a cualquier especie, y ha sido una preocupación para los humanos durante siglos. Este problema ha sido estudiado durante los últimos 50 años, con la finalidad de obtener ideas acerca de cómo se esparce una enfermedad por la población, y además, decidir cómo vacunar a la gente para detener la propagación de dicha enfermedad.

Aunque ha habido varios enfoques en el estudio de la propagación de enfermedades, los investigadores afirman que el primer intento para formular un modelo matemático es

el de Lowell Reed y Wade Hapton en los años 20's (Abbey, 1952). El modelo fue llamado SIR (Susceptible/Infective/Recovery). En este modelo una población es dividida en tres clases de acuerdo a su estatus en relación a la enfermedad de interés: *Susceptible* que significa que ellos están libres de la enfermedad pero que pueden llegar a tenerla. *Infectiva*, significa que tienen la enfermedad y pueden pasarla a otros. Finalmente, *Recuperada* significa que se han recuperado de la enfermedad y ya no pueden trasmitirla. Algunos valores de probabilidad por unidad de tiempo son empleados para calcular las transiciones de un estado dado a otro.

Los algoritmos epidémicos también pueden ser vistos desde otro punto de vista. Dado que el modelo emplea reglas de transición de diferentes estados que cada individuo de la poblacion puede presentar durante el proceso epidémico, otros autores han estudiado y modelado la propagación de enfermedad en una población por medio de autómatas celulares (Maniatty *et al.*, 1998).

Nosotros presentamos un nuevo algoritmo, el *algoritmo de infección*. Éste posee algunas propiedades de los modelos epidémicos, una de ellas es que la información es transmitida entre los individuos de una población empleando reglas de transición. Sin embargo, la diferencia radica en que el número de estados que nosotros empleamos es mayor que el empleado en los modelos epidémicos. Como se vió anteriormente los modelos epidémicos emplean tres estados: susceptible, infectivo, recuperado. En nuestro caso usamos cuatro estados: EXPLOTADO, NO-EXPLOTADO,PROPUESTO y AUTOMÁTICO, que traduciéndolos en este contexto serían: *Enfermos, Sanos, Infectados e Inmunes*, respectivamente.

El estado EXPLOTADO (*Enfermo* representa las células que han sido infectadas por el virus, sin solución alguna. Se refiere a los pixeles que han sido calculados para encontrar las correspondencias. El estado NO-EXPLOTADO (*Sano*) se refiere a las

células que no han sido infectadas por el virus, o bien, las células que permanecen en el estado inicial. El estado AUTOMÁTICO (*Inmune*) representa las células que no pueden ser infectadas por el virus, son inmunes a la enfermedad. Finalmente, el estado PROPUESTO (*Infectado*) representa las células que han adquirido el virus con posibilidad de recuperarse.

IV.7 Descripción del algoritmo de infección

El algoritmo de infección usa el concepto de virus natural en la búsqueda de correspondencias con imágenes estéreo reales. El propósito es encontrar todas los puntos que correspondan en ambas imágenes ahorrando el máximo número de cálculos posibles para lograr obtener una imagen virtual intermedia. El objetivo principal es ahorrar el mayor número de operaciones conservando la calidad de la imagen en relación a una imagen generada con una búsqueda exhaustiva.

Usamos un proceso de búsqueda basado en reglas de transición, las cuales son similares a las de un autómata celular. Nuestro autómata puede ser visto como una secuencia de pasos en tiempo discreto seguidos por un conjunto de reglas definidas. El algoritmo de infección usa cuatro clases de rejillas (matrices):

1. Una rejilla que corresponde a la imagen izquierda. Almacena los valores de intensidad de la imagen izquierda. Los valores de la rejilla no cambian.

2. Una segunda rejilla que corresponde a la imagen derecha. Almacena los valores de intensidad de la imagen derecha. Los valores de la rejilla no cambian.

3. Una tercera rejilla que corresponde al autómata. Almacena los estados que permiten llevar el control de los pixeles correspondientes en la imagen derecha. En esta rejilla es donde el conjunto de reglas se aplica para producir el proceso de correspondencia basándose en la información de la imagen y el estado de cada pixel. Los valores de la rejilla cambian dinámicamente.

4. La cuarta rejilla corresponde a la imagen virtual intermedia que se genera a partir del proceso de correspondencia. Almacena los valores de intensidad de la nueva imagen. En esta rejilla se visualiza la correspondencia de la imagen izquierda y la imagen derecha.

La Figura 29 ilustra las relaciones entre cada una de las rejillas. El autómata indica cuándo evaluar, proponer o asignar automáticamente un pixel. Además lleva el control de los pixeles que se generan en la imagen virtual.

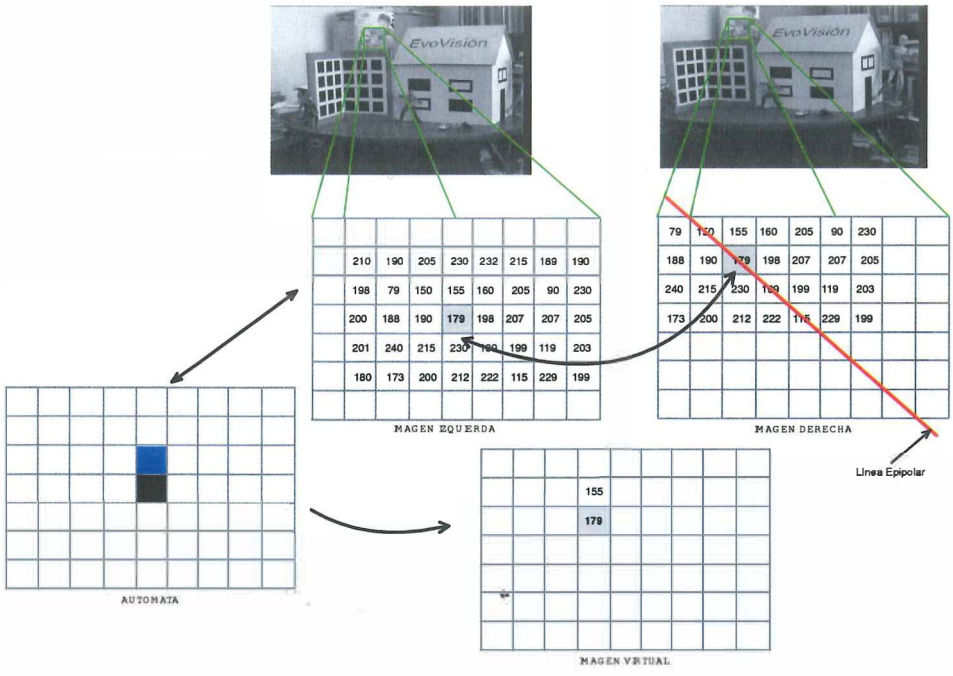


Figura 29: Relación entre cada una de las rejillas usadas en el algoritmo de infección. Los valores numéricos representan el valor del nivel de intensidad de una región de la imagen. El cuadro azul representa el estado *enfermo* y el color negro el estado *sano*

La calibración nos permite obtener las matrices de proyección (P y P') de cada imagen y la matriz Fundamental (F). Una vez que se conoce la geometría epipolar se procede a realizar la búsqueda de correspondencias entre los puntos utilizando el autómata celular como herramienta de búsqueda global.

Las reglas se codifican en una matriz de tamaño 14×15 que corresponde a las 14 reglas representadas en los renglones de la tabla X en la Sección V.3.1 y las 15 columnas que se usan para cada regla. La rejilla que representa el autómata se inicializa en estado NO-EXPLOTADO (*Sano*). Se utiliza un detector de esquinas sobre la rejilla de la imagen izquierda para obtener los puntos más prominentes de la imagen. Estos puntos sirven como núcleo de infección una vez que se conoce su correspondiente punto en la imagen derecha.

Una vez que se tienen los núcleos de infección en la rejilla del autómata, se procede a recorrer la rejilla para que las reglas puedan irse aplicando e ir realizando las inferencias y cálculos correspondientes. La rejilla se recorre repetidas veces hasta que el número de estados AUTOMÁTICOS (*Inmunes*) y estados EXPLOTADOS (*Enfermos*) generados en el paso de tiempo t y $t+1$ sea el mismo. Lo anterior indica que al no haber cambios entre los estados, se terminó de explorar la imagen.

Cada una de las rejillas es del tamaño del par estéreo, (484×768) células. La mayoría del tiempo se trabaja sobre la rejilla del autómata porque nos indica cuándo evaluar, proponer o asignar automáticamente un pixel. Cuando el estado del automáta indica evaluación, se toma información del pixel de la imagen izquierda, se calcula su linea epipolar en la imagen derecha y se realiza la búsqueda a lo largo de la línea. Una vez que se conoce su correspondiente a través de la correlación, el punto se reproyecta en la rejilla de la nueva imagen (virtual). El control de las evaluaciones que se realizan durante el proceso se lleva a través de las reglas aplicadas en el autómata.

El criterio de terminación del algoritmo depende del número de evaluaciones y asignaciones automáticas que se realicen en el proceso. Si el número de estados EXPLOTADOS (*Enfermos*) se mantiene constante en dos ciclos continuos y a su vez, el número de estados AUTOMÁTICOS (*Inmunes*), entonces, el algoritmo finaliza la

búsqueda de correspondencias. Lo anterior, significa que al finalizar la búsqueda, solo obtendremos pixeles que hayan sido evaluados o asignados automáticamente, por lo tanto, el autómata solo tendrá almacenado los estados EXPLOTADOS (*Enfermos*) y AUTOMÁTICOS (*Inmunes*). Para corroborar que realmente el autómata finaliza con esos estados, se podrá constatar con la simulación de reglas en la Sección V.2. Se observará que la imagen final de la simulación, ver Figura 40, solo contendrá el color azul y verde que corresponde a los estados mencionados anteriormente.

Finalmente, cuando el criterio de terminación se cumple, la correspondencia se visualiza en la imagen virtual intermedia que se genera.

A continuación se muestra el pseudocódigo del algoritmo de infección.

```
INICIO
Cálculo de la matriz Fundamental (F) de izquierda-derecha
Cálculo de las matrices de Proyección MM1 y MM2 de la imagen
izquierda y derecha respectivamente.
Codificación de la matriz de reglas.
Inicialización de la rejilla del autómata al estado SANO (No-Explotado).
Detección de esquinas con el detector KyR Kitchen y Rosenfeld (1982)
en la imagen izquierda.
Mientras que el número de estados AUTOMÁTICOS (Inmunes) y
EXPLOTADOS (Enfermos) sea diferente en el paso de tiempo t y t + 1.
DO
    FOR col=6 to N
        FOR ren=6 to M
            vecinos=cuentaClaseVecinos(ren,col)
            pos=edosPosicion(ren,col)
            edoCelda=rejillaAut[ren][col].celda
```

```
14.            accion=busquedaRegla(edoCelda,vecinos,pos)
15.            SWITCH(accion)
16.               1: EXPLOTAR
17.               2: PROPONER
18.               3: Analizar la inclinación de la línea epipolar
19.                  IF la inclinación = a la inclinación de al menos
                     tres vecinos evaluados
20.                     Asignar AUTOMÁTICAMENTE
21.                  ELSE
22.                     EXPLOTAR
23.                  ENDIF
24.            ENDSWITCH
25.         ENDFOR
26.      ENDFOR
27.   ENDDO
28.   Imagen virtual generada
29.   FIN
```

Definición de Datos y Funciones

- *vecinos*: Es una estructura de datos que almacena el número total de cada uno de los estados que se repiten dentro del vecindario.

- *pos*: Es una estructura de datos que almacena el número entero del estado en las 8 posiciones centrales del vecindario alrededor de la célula central, ver Figura 21.

- $rejillaAut[ren][col].celda$: Es una matriz con la misma dimensión de la imagen, donde cada uno de sus elementos es una estructura de datos que almacena: *a*) el estado de la célula central, *b*) la dirección en grados de la línea epipolar, *c*) las coordenadas del pixel correspondiente en la imagen derecha y *d*) el índice de la regla que fue aplicada por última vez en la célula central.

- *accion*: Es una variable que almacena el número entero del estado cuando una regla es verdadera.

- $cuentaClaseVecinos(ren, col)$: Función que regresa la cantidad de cada uno de los estados que son iguales dentro del vecindario.

- $edosPosicion(ren, col)$: Función que regresa el valor del estado en cada una de las nueve posiciones centrales.

- $busquedaRegla(edoCelda, vecinos, pos)$: Función que regresa el valor del estado en el paso de tiempo $t + 1$ una vez que la regla se cumple.

- $EXPLOTAR$: Esta expresión indica que el pixel de la imagen izquierda que está relacionado con un estado en la rejilla del autómata debe de evaluarse. Se manda llamar una función que calcule la linea epipolar para ese punto y calcule la correlación para encontrar el punto correspondiente en la imagen derecha. El estado de la célula central cambia por el valor del estado que se encuentra en la variable *accion*.

- $PROPONER$: Esta expresión indica que la célula central debe ser marcada para después analizar la posibilidad de asignar automáticamente el pixel correspondiente en la otra imagen.

- $Asignar\ AUTOMATICAMENTE$: Esta expresión indica la asignación de las coordenadas del pixel en la imagen derecha sin realizar ningún cálculo cuando la regla sea verdadera. La asignación se basa en la información de los estados alrededor del pixel central.

IV.8 Formalización del Algoritmo

Sean dos imágenes (I_l, I_r) definidas como matrices de pixeles con una dimensión de $m \times n$ y valores de intensidad del pixel $I = \{0, 1, ..., 255\}$, donde $M = \{1, ..., m\}$ y $N = \{1, ..., n\}$.

El algoritmo de infección está compuesto de una parte local y una parte global. La parte local se encarga de hacer los cálculos para encontrar los pixeles correspondientes en la imagen derecha. En esta parte se hace uso de las reglas de manera individual y de las restricciones de visión como la línea epipolar y la restricción de orientación. La parte globlal, por el contrario, describe el comportamiento del algoritmo en general, es decir, cómo se propaga el apariamiento.

El algoritmo de correspondencia puede ser descrito formalmente de la siguiente manera:

Parte Global

La parte global puede ser descrita por la definición formal de un autómata celular:
Dado un conjunto finito S y una dimensión d podemos considerar una rejilla n-dimensional en la cual cada punto tiene una etiqueta del conjunto S. Formalmente, la rejilla es el conjunto $L = Z^d$.

Definición 4. Autómata celular: *Un autómata celular es determinado por un cuádruple $A = (S, d, Q, f)$, donde S es un conjunto finito, d es un entero positivo, $Q \subset Z^d$ es un conjunto finito, y f: $S^Q \rightarrow S$ es una función local arbitraria.*

Esta definición formal es interpretada de la siguiente forma: un estado puede tomar cualquier valor de Q y puede ser inicialmente q_0 (estado inicial de las células). El autómata lee una cadena de símbolos como entrada, uno a la vez, cada símbolo forma parte de S. Si un símbolo s es leído de la entrada del autómata que se encuentra en estado q, la función de transición de la regla f cambia el estado q a un estado s

y procede a leer el próximo símbolo de entrada. Al terminar de leer los símbolos de entrada, se tiene un conjunto Q_f que corresponde a los estados que quedan al final del proceso.

En nuestro caso, tenemos que A=(S,d,Q,f)

$S = \alpha_1, \varphi_2, \beta_3, \varepsilon_0, *, 0, 1, 2, 3$

$d = 2$

$Q = \alpha_1, \varphi_2, \beta_3, \varepsilon_0, *$

f: $S^Q \rightarrow S$.

$Qf = \{\alpha_1, \beta_3\}$

$q_0 = \{\varepsilon_0\}$

Lo anterior indica que el autómata usa un alfabeto de nueve símbolos, utiliza cuatro estados distintos de los cuales solo quedan α_1 y β_3 al final del proceso. La razón del por qué solo quedan los dos estados anteriores va en relación a la idea original del algoritmo que fué evitar calcular la mayoría de los pixeles de la imagen; por lo tanto, al final del proceso de correspondencia cada pixel debió ser evaluado o asignado automáticamente. La regla de transición se aplica para cada pixel de la imagen izquierda tomando en cuenta los estados de las células del vecindario modificando solamente el estado de la célula central.

Parte Local

La parte local corresponde a las reglas de transición que pueden ser formalizadas a través de Lógica de Predicados. Las transiciones de cada regla pueden ser visualizadas a través de Grafos.

Definición 5. Grafo:*Un grafo el cual es denotado por* $G = (X, \Gamma)$, *es el par que consiste del conjunto* X *y la función* Γ. *Donde,* $X = \varepsilon_0, \alpha_1, \varphi_2, \beta_3$ *y* $U = (\varepsilon_0, \varepsilon_0), (\alpha_1, \alpha_1), (\varepsilon_0, \beta_3), (\beta_3, \varphi_2), (\beta_3, \alpha_1)$

La Figura 30 representa las transiciones de las catorce reglas, donde se puede observar que se tienen solamente dos estados finales (α_1 y β_3). De un estado NO-EXPLOTADO (*Sano*) pasa a un estado EXPLOTADO (*Enfermo*) ó a un estado PROPUESTO (*Infectado*). De un estado PROPUESTO (*Infectado*) pasa a un estado AUTOMÁTICO (*Inmune*) ó a un estado EXPLOTADO (*Enfermo*).

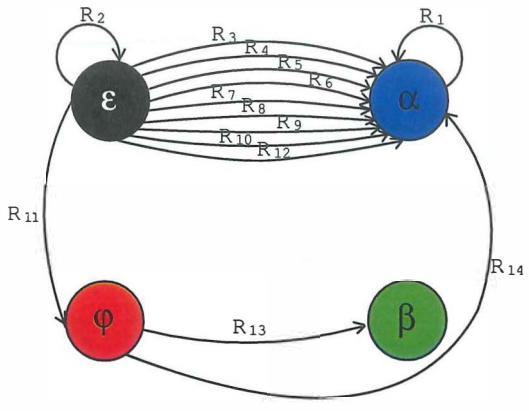


Figura 30: Grafo Global que representa la transición de las 14 reglas.

Las transiciones individuales pueden ser representadas a través de un subgrafo, ver Figura 31.

Un subgrafo de un grafo (X, Γ) es definido como un grafo de la forma (A, Γ_A), donde $A \subset X$, en la cual la función Γ_A esta definida por:

$\Gamma_{A\varepsilon_0} = \Gamma_{\varepsilon_0 \cap A}$.

De igual forma para $\Gamma_{B\alpha_1}$ tenemos:

$\Gamma_{B\alpha_1} = \Gamma_{\alpha_1 \cap B}$.

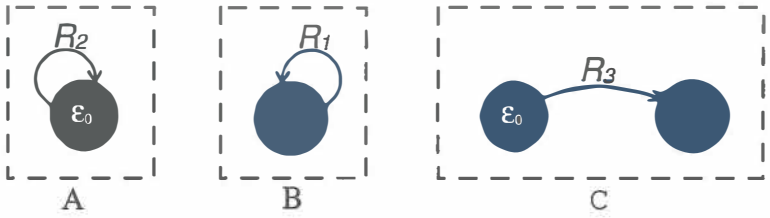


Figura 31: Subgrafo A, B, C de la regla 2 (R_2), regla 1 (R_1) y regla 3 (R_3).

IV.8.1 Descripción y Notación General de una Regla

Las células evolucionan en pasos de tiempo discreto de acuerdo al conjunto de reglas determinísticas usadas en el autómata. Cada regla depende de los estados del vecindario. El estado de la célula central cambiará, si y sólo si, los estados de sus vecinos cumplen con la regla. La función de transición del próximo estado se aplica al vecindario de 25 células que se usa en esta tesis.

La regla se expresa a través de lógica de predicados porque permite manejar las posiciones de las células individualmente y en conjunto. Es más adecuada de usar esta forma, ya que no solamente se toma en cuenta las posiciones de los estados, sino también la suma de cada uno de los estados que se repiten en el vecindario. Por ejemplo, la cantidad de estados que son AUTOMÁTICOS (*Inmunes*) dentro del vecindario en el paso de tiempo (t).

La estructura general de una regla es de la siguiente manera:

$$\begin{aligned} \text{R}(edo)= \quad & \text{IZQARR}(edo) \wedge \text{COLARR}(edo) \wedge \text{DERARR}(edo) \wedge \text{IZQFILA}(edo) \wedge \\ & \text{DERFILA}(edo) \wedge \text{IZQABA}(edo) \wedge \text{COLABA}(edo) \wedge \text{DERABA}(edo) \\ & \wedge \text{VECEXP}(n) \wedge \text{VECPROP}(n) \wedge \text{VECEXT}(n) \wedge \\ & \text{VECAUT}(n) \rightarrow \text{ACCION}(edo) \end{aligned}$$

donde:

edo: Representa el valor del estado como un número entero positivo (0,1,2,3,5). El valor

5 representa la variable comodín.

n: Representa un valor entero positivo. n puede ser un valor distinto en cada cláusula.

$R()$: Representa el estado de la célula central en el paso de tiempo t.

$IZQARR(), COLARR(), ..., DERABA()$: Representa el estado de las posiciones de la célula en el vecindario.

$VECEXP()$: Representa el número de edos. EXPLOTADOS(*Enfermos*) que se repiten en el vecindario, excepto la célula central.

$VECPROP()$: Representa el número de edos. PROPUESTOS(*Infectados*) que se repiten en el vecindario, excepto la célula central.

$VECEXT()$: Representa el número de edos. AUTOMÁTICOS (*Inmunes*) que no se encuentren dentro de las 9 células centrales.

$VECAUT()$: Representa el número de edos. AUTOMÁTICOS (*Inmunes*) que se repiten en el vecindario, excepto la célula central.

$ACCION()$: Representa el número del estado a cambiar en la célula central cuando la regla es verdadera.

La regla se aplica basándose en la información de los estados de las células vecinas. El estado de la célula central cambia cuando la regla es verdadera (cuando cada una de las cláusulas del predicado es verdadera). Para ello, se toma en cuenta los estados de las posiciones del vecindario interno, ver Figura ??a y un número n de estados que se repiten dentro del vecindario. Las posiciones del vecindario interno corresponde a las cláusulas $IZQARR(edo)$, $COLARR(edo)$, $DERARR(edo)$, $IZQFILA(edo)$, $DERFILA(edo)$, $IZQABA(edo)$, $COLABA(edo)$ y $DERABA(edo)$. Las cláusulas que toman en cuenta el número de estados que se repiten dentro del vecindario son, $VECEXP(n)$, $VECPROP(n)$, $VECEXT(n)$, $VECAUT(n)$.

Tabla II: Estructura de la Tabla de reglas

C	#	IZQ ARR	COL ARR	DER ARR	IZQ FILA	DER FILA	IZQ ABA	COL ABA	DER ABA	VEC EXP	VEC PROP	VEC EXT	VEC AUT	A
⋮						⋮								⋮
⋮						⋮								⋮

Cada una de las reglas están agrupadas en una tabla, ver Tabla II, donde los renglones representan cada una de las reglas que están numeradas en la segunda columna. La información que se almacena en la tabla son los valores numéricos (0,1,2,3,5) que representan cada uno de los 4 estados (NO-EXPLOTADO (*Sano*), EXPLOTADO (*Enfermo*), PROPUESTO (*Infectado*), AUTOMÁTICO (*Inmune*)) y la variable comodín que especificamos anteriormente. La Figura 27 muestra la representación de los estados de manera numérica, gráfica (color) y de una letra griega.

La primera columna (C) de la tabla de regleas representa el estado de la célula central, la segunda columna (#) el número de regla, de la columna 3 a la 10 representan los estados en una posición específica del vecindario, ver Figura 32. La columna 11 a la 14 almacena la cantidad de estados EXPLOTADOS (*Enfermos*), AUTOMÁTICOS (*Inmunes*) en el vecindario externo, ver Figura 33 y vecindario interno. La última columna (A) almacena el valor del estado que va a tener la célula central cuando la regla sea verdadera. La descripción detallada del contenido de la tabla se muestra en el capítulo V.

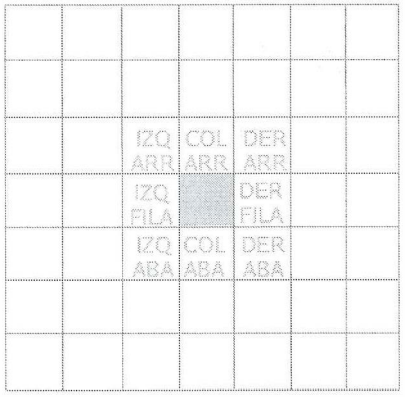


Figura 32: Relación de la posición de 8 estados centrales

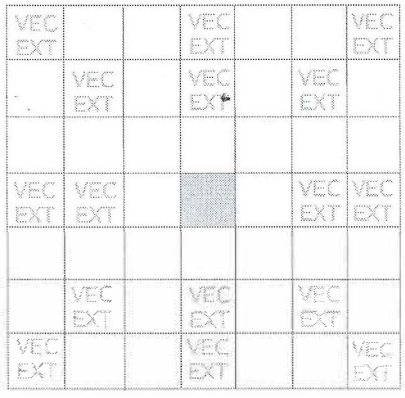


Figura 33: Relación de la posición de los estados que se encuentran en la parte exterior del vecindario

IV.9 Conclusiones

En el presente capítulo se detallaron los conceptos más importantes para entender el uso de la herramienta de trabajo, así como también el nivel de complejidad que representa el espacio de las reglas. Se dió a conocer el funcionamiento del algoritmo de infección y el uso de conceptos de visión por computadora en el proceso de correspondencia. Se mostró el diseño de cada una de las reglas en sus distintas representaciones y cómo son codificadas a la tabla de reglas. Lo anterior, con el fin de entender cómo trabaja de manera global el autómata celular y cómo se esta usando en la tesis.

Es importante entender cada uno de los componentes que conforman el conjunto de reglas. Entre mayor sea el número de estados y el número de vecinos que se usen en el autómata, mayor complejidad habrá para identificar el conjunto de reglas que mejor se adapten al problema.

A medida que estos parámetros aumentan la cantidad de mapeos aumenta exponencialmente, tal como se pudo apreciar en la explicación del caso de un autómata unidimensional y en el bidimensional. Se observó que la diferencia entre ambos autómatas fue de solo 6 vecinos, sin embargo, el aumento de mapeos posibles aumentó exorbitadamente.

El conjunto de reglas fue propuesto basándonos en la idea original de cómo hacer la propagación de la correspondencia a través de la imagen. Esta idea consistió principalmente en *adivinar* correspondencias de algunos vecinos utilizando la información de los estados en las posiciones de los pixeles. Para llegar al diseño de ese conjunto de reglas se basó en el hecho de que si existe un pixel altamente correlacionado en la imagen derecha, esto es, si se han realizado los cálculos para obtener la correlación del pixel, entonces los pixeles alrededor del pixel evaluado tienen mucha probabilidad de estar

también altamente correlacionados. Lo anterior se comprobó con experimentaciones sobre ciertas regiones de la imagen izquierda.

Fue de vital importancia el uso de una variable que sirviera como comodín que permitiera reemplazar estados en el conjunto de reglas. Esto nos permitió considerar solo la información del vecindario que fuera necesaria y reducir el número de reglas.

En la correspondencia, el autómata celular representa la información de la imagen izquierda la cual es manipulada a través de cuatro estados. Los estados van cambiando por medio de las reglas y éstas a su vez permiten el comportamiento de la propagación de la correspondencia. La expansión se genera como si fuera un virus en un entorno de células vivas. El uso del autómata celular en la búsqueda de correspondencias ayuda en gran medida a evitar recalcular pixeles que ya fueron evaluados anteriormente.

Capíulo V

Experimentación

Este capítulo introduce las experimentaciones realizadas sobre el problema de correspondencia. En un inicio se empezó a trabajar en dos direcciones: por un lado, se estudió sobre el problema de correspondencia y por otro lado, la herramienta global refiriéndonos al autómata celular. Al analizar el problema de correspondencia se reproyectaron los puntos 2D del par estéreo en 3D (Sección V.1) y fueron visualizados en VRML[1]. Se planteó una meta a corto plazo, diseñar el conjunto de reglas que permitieran obtener un ahorro alrededor de un 50% de evaluaciones. Para ello, se realizó una simulación (ver Sección V.2) y posteriormente la implementación del algoritmo con el conjunto de reglas utilizando información del par estéreo.

La implementación del algoritmo de infección consistió en analizar 5 casos distintos de ahorro en evaluaciones; los casos se encuentran en la Sección V.3.

V.1 Reconstrucción 3D de la imagen

La Figura 34 muestra la reconstrucción 3D de los puntos de la imagen. Esta reconstrucción se visualizó en el Lenguaje de Modelado de Realidad Virtual, VRML [2], se puede apreciar claramente que existen dos conjuntos de puntos los cuales están dispersos en el espacio. La explicación del por qué los puntos se encuentran divididos en dos conjuntos es debido a que los puntos se proyectan por detrás de la cámara como se

[1] Virtual Reality Modeling Language
[2] En inglés, Virtual Reality Modeling Language

muestra en la Figura 35. La dispersidad de los puntos en el espacio se debe principalmente a que los puntos de la imagen se encuentran en un plano y al calcular su tercera dimensión quedan espacios entre los puntos por la posición de los objetos en el mundo real que le dan el sentido de profundidad. Por ello, el modelo 3D se ve mejor desde un punto de vista cuando los puntos son reproyectados en un plano, tal como se muestra en la Figura 1 en la Sección I.1.

Figura 34: Reconstrucción 3D de todos los puntos de la imagen.

V.2 Simulación de las reglas

Se diseñó un programa en C++ donde se visualiza el comportamiento de las reglas mediante el uso de un autómata celular. La visualización del comportamiento consistió en representar las transiciones de las reglas a través de los colores que representa a cada uno de los estados (negro,rojo,verde y azul). El autómata tiene un total de 3920 células,

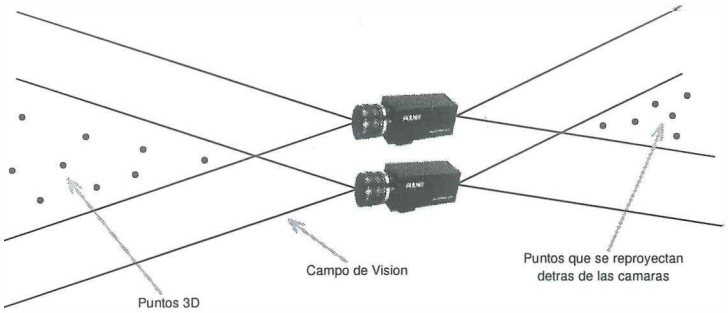


Figura 35: Proyección de los puntos a partir de un par estéreo.

utiliza 4 estados y 14 reglas (Tabla X). Durante el proceso de evolución de los estados se lleva el control del número de células EXPLOTADAS (*Enfermas*) y AUTOMÁTICAS (*Inmunes*). Este control permite conocer el porcentaje aproximado de asignaciones automáticas que se esperaría obtener al utilizar las reglas en la implementación real.

El algoritmo comienza seleccionando al azar un número n ($1 \leq n \leq 3920$) de células como núcleos de infección. A estas células se les asigna el estado EXPLOTADO (*Enfermo*) y a partir de la inicialización se aplica cada una de las reglas alrededor del núcleo de infección, ver Figura 36.

La Tabla III muestra la evolución de las reglas en cinco etapas del proceso. Se observa que el número de estados AUTOMÁTICOS (*Inmunes*) aumenta conforme la propagación crece alrededor del núcleo de infección. Entre menor número de núcleos de infección mayor es el número de iteraciones que se requieren para terminar el proceso. Por ello, el último ciclo de las Tablas III , IV , V , VI y VII es variable.

La Figura 36 muestra la pantalla de la simulación donde se forma la estructura inicial del algoritmo. Los colores representan cada uno de los estados que se vieron anteriormente en la Figura 27 de la Sección IV.4.3.

Tabla III: Tabla de datos de la Simulación de Reglas

Figura	CICLO	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)
36	2	48	0
37	9	375	329
38	14	931	807
39	19	1311	1232
40	61	2021	1899
Número de núcleos de infección: 10			
Porcentaje de Ahorro: $(1899 \times 100) \div 3920 = 48.44\%$			

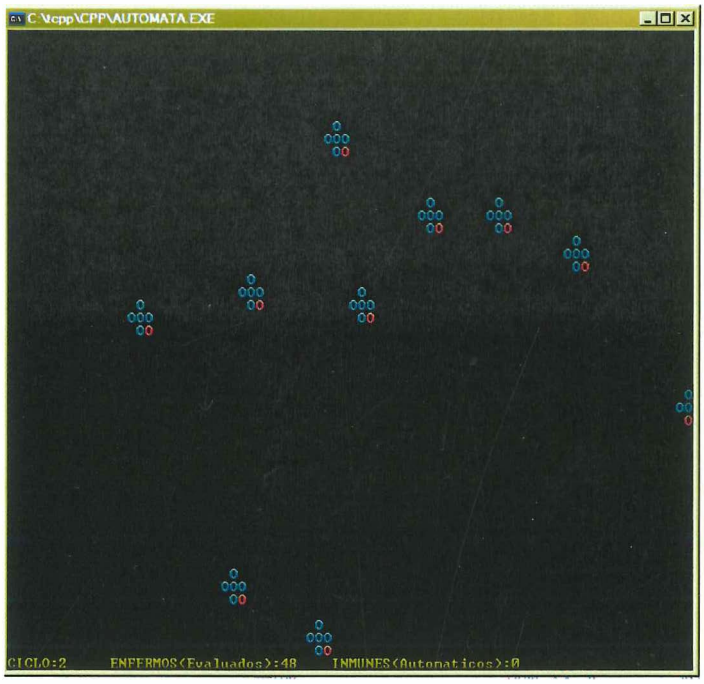


Figura 36: Estructura inicial de las reglas alrededor del núcleo de infección.

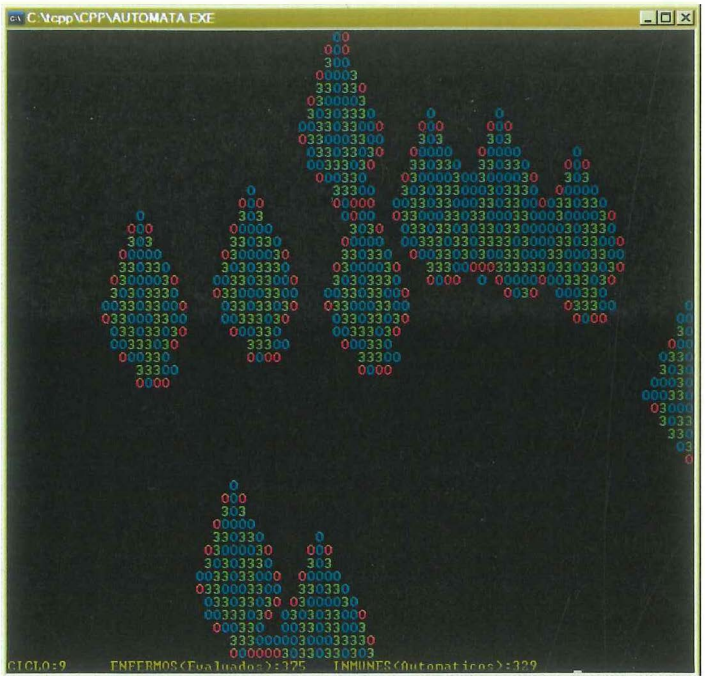


Figura 37: Propagación de las reglas de transición en el ciclo 9.

La Figura 37 muestra la pantalla de la simulación en el ciclo 9. La imagen muestra muestra el crecimiento de la propagación alrededor del núcleo de infección.

La Figura 38 muestra la pantalla de la simulación en el ciclo 14, se puede observar que las regiones se agrupan entre sí, formando cada vez regiones más grandes. Las asignaciones automáticas representadas por el color verde y las evaluaciones representadas por el color azul, son distribuidas en el autómata. La propagación va creciendo en cada ciclo hasta completar toda la ventana como se muestra en la Figura 40.

El comportamiento del autómata es diferente cada vez que el programa es ejecutado (ver Tabla IV , V , VI , VII). Lo anterior, depende de la ubicación de los núcleos de infección ya que las reglas interactúan de acuerdo a la información local de sus vecinos para producir un comportamiento global. En este caso, el comportamiento global está

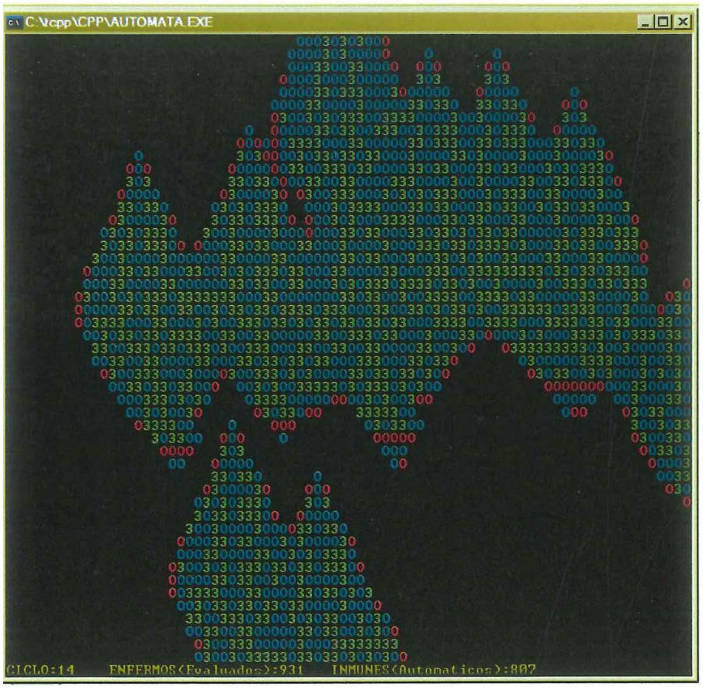


Figura 38: Propagación de las reglas de transición en el ciclo 14.

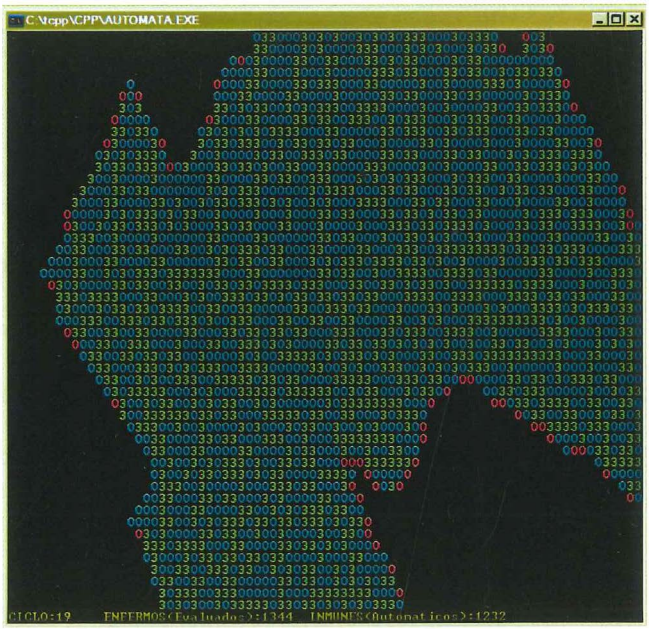


Figura 39: Propagación de las reglas de transición en el ciclo 19.

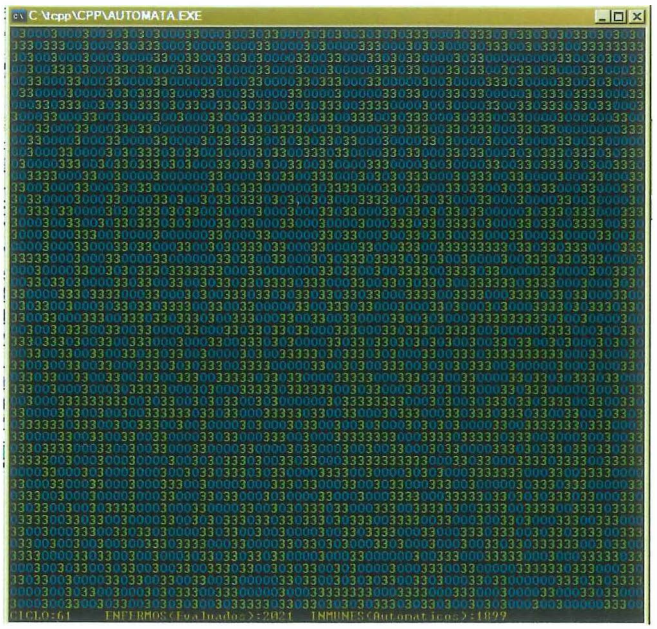


Figura 40: Imagen final de la propagación de las reglas en el ciclo 61.

relacionado con el porcentaje de ahorro que se obtiene independientemente de la ubicación de cada núcleo de infección. Puede observarse que en todos los experimentos se ahorra entre un 45% y 50% de operaciones usando como entrada 10 núcleos de infección (Ver Tabla VIII).

Se realizaron experimentos utilizando 1, 5, 10, 20 y 25 núcleos de infección para corroborar que independientemente de la ubicación de los núcleos y del número de núcleos que se utilicen, el porcentaje de ahorro siempre se mantiene dentro de un rango. En la Tabla IX se observa que el porcentaje de ahorro se mantiene dentro del rango 45-50%. El porcentaje de ahorro promedio de acuerdo a las experimentaciones fue de un 47.48%.

Tabla IV: Ejecución 1. Tabla de datos.

CICLO	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)
1	10	0
2	50	0
4	69	41
9	380	331
11	569	500
14	845	746
16	1052	940
19	1276	1141
22	1472	1267
50	2114	1806
Número de núcleos de infección: 10		
Porcentaje de Ahorro: $(1806 \times 100) \div 3920 = 46.07\%$		

Tabla V: Ejecución 2. Tabla de datos.

CICLO	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)
1	10	0
2	49	0
4	78	47
9	360	342
11	542	499
14	796	722
16	968	898
19	1204	1136
22	1406	1321
45	1969	1951
Número de núcleos de infección: 10		
Porcentaje de Ahorro: $(1951 \times 100) \div 3920 = 49.77\%$		

Tabla VI: Ejecución 3. Tabla de datos.

CICLO	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)
1	10	0
2	50	0
4	71	40
9	385	343
11	595	540
14	916	805
16	1078	980
19	1315	1183
22	1476	1335
58	2050	1870
Número de núcleos de infección: 10		
Porcentaje de Ahorro: $(1870 \times 100) \div 3920 = 47.70\%$		

Tabla VII: Ejecución 4. Tabla de datos.

CICLO	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)
1	10	0
2	50	0
4	69	41
9	360	336
11	590	530
14	901	823
16	1155	1055
19	1485	1368
22	1689	1562
40	2033	1887
Número de núcleos de infección: 10		
Porcentaje de Ahorro: $(1887 \times 100) \div 3920 = 48.13\%$		

Tabla VIII: Resumen de las Tablas III , IV , V , VI y VII

TABLA	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)	PORCENTAJE
III	2021	1899	48.44%
IV	2114	1806	46.07%
V	1969	1951	49.77%
VI	2050	1870	47.70%
VII	2033	1887	48.13%
Número de núcleos de infección: 10			

Tabla IX: Porcentaje de Ahorro Promedio

# NÚCLEOS	EXPLOTADOS (*Enfermos*)	AUTOMÁTICOS (*Inmunes*)	PORCENTAJE
1	2094	1820	46.42%
5	2049	1871	47.72%
10	2033	1887	48.13%
15	2051	1869	47.67%
20	2049	1871	47.72%
25	2068	1852	47.24%
Porcentaje de Ahorro Promedio: 47.48%			

V.2.1 Objetivo de la Simulación

El objetivo principal de la simulación fue diseñar un conjunto de reglas que permitiera un ahorro significativo de operaciones para el algoritmo de correspondencias y obtener un porcentaje de ahorro similar en la implementación con las imágenes reales. El diseño debía tener la flexibilidad necesaria para agregar, borrar o modificar las reglas dentro del conjunto de reglas.

La información del vecindario permite tomar decisiones sobre cuál regla es la que mejor se adapta a la configuración del vecindario en el paso de tiempo t. El orden de las reglas dentro de la tabla es importante en su aplicación, ya que si las reglas son las mismas pero el orden se cambia, el comportamiento puede ser totalmente distinto. Lo anterior, es debido a que cada regla individualmente es determinística, pero al combinarlas en conjunto el comportamiento es dinámico.

V.3 Implementación del algoritmo

La implementación consistió en el desarrollo del algoritmo de correspondencias entre un par de imágenes estéreo calibradas usando un autómata celular y un conjunto de reglas que dirigen el comportamiento y funcionamiento del algoritmo. La implementación fue

desarrollada bajo el sistema operativo Linux en una computadora portátil Dell Inspiron 8500, Pentium 4 con 2.0GHz y 256 MB[3] de memoria RAM[4].

El *algortimo de infección* se agregó a un módulo del sistema de reconstrucción 3D que pertenece al grupo de investigación EvoVisión del CICESE[5]. Todos los módulos fueron reprogramados usando unas librerías programadas en C++, diseñadas especialmente para visión por computadora llamadas VXL[6]. Las librerías fueron creadas extrayendo el núcleo de funcionalidades de IUE[7] y Tarjet Jr, con la finalidad de usarlas sobre cualquier plataforma pretendiendo que el sistema fuera portable, rápido y eficiente. Las librerías se encuentran disponibles en la dirección http://vxl.sourceforge.net

El motivo principal de utilizar estas librerías en el sistema de reconstrucción es que permiten que el software sea portable ya que anteriormente el software solo podía estar en una sola computadora. Además, que fueron diseñadas especialmente para la investigación en visión por lo que el proceso es mucho más eficiente que programar paso a paso funciones básicas, como por ejemplo, la asignación de memoria y operaciones con matrices, funciones del numerical recipes (Press *et al.*, 1994), funciones de procesamiento de imágenes, entre otras funcionalidades.

Los módulos incluidos actualmente en el sistema de reconstrucción son los siguientes:

1. Adquisición de Fotos.

2. Calibración de la cámara izquierda.

3. Calibración de la cámara derecha.

[3]En inglés, MegaBytes
[4]En inglés, Random Access Memory
[5]Centro de Investigación Científica y de Educación Superior de Ensenada
[6]En inglés, Vision x Libraries
[7]En inglés, Image Understanding Enviroment

4. Generación de puntos 3D.

El algoritmo de infección se encuentra dentro del módulo de Generación de puntos 3D. La descripción del funcionamiento del sistema, así como también, de cada uno de sus módulos se encuentran en el Apéndice A.

Los resultados del algoritmo de infección están basados en cinco casos de experimentación que fueron desarrollados a partir de los resultados de la simulación. Los cinco casos consisten en la obtención de diferentes porcentajes de ahorro en operaciones utilizando el mismo par estéreo.

En primera instancia, se usó el conjunto de reglas diseñadas en la simulación obteniendo como resultado un 47% de ahorro en operaciones con la intención de conservar la calidad en la imagen, ver la subsección V.3.1. Una vez que se obtuvieron los primeros resultados de la implementación, se modificaron las reglas para generar cuatro conjuntos de reglas adicionales y observar el comportamiento del algoritmo.

Finalmente, con las modificaciones realizadas a las reglas se obtuvieron los 5 casos de porcentajes de ahorro (0%,47%,70%,78%,99%) que se encuentran en las siguientes secciones V.3.2 , V.3.1 , V.3.3 , V.3.4 y V.3.5.

La Figura 41 muestra las dos imágenes que fueron usadas como datos de entrada en cada uno de los 5 experimentos. Cada una de las imágenes tiene una dimensión de 768×484 pixeles. Las imágenes de entrada cuentan con oclusión, regiones de baja textura, patrones repetitivos y campo de visión por la traslación.

La subsección V.3.1 detalla el proceso del algoritmo, la descripción de cada una de las 14 reglas y la evolución de los estados a través de imágenes. Para los demás casos solo se muestra los resultados principales del algoritmo.

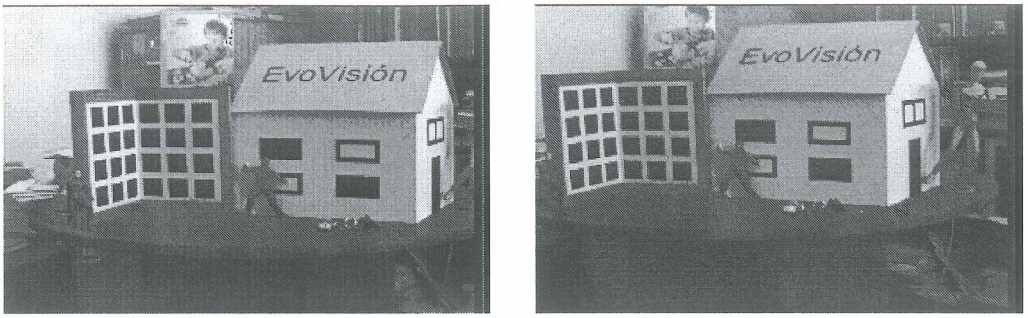


Imagen Izquierda Imagen Derecha

Figura 41: Par estéreo utilizado como datos de entrada en cada uno de los experimentos.

V.3.1 Experimento 1: Caso del 47% de ahorro

Este experimento se realizó con el conjunto de reglas mostradas en la simulación pero ahora aplicadas con las imágenes reales. Se comprobó que las reglas para este caso se mantienen en el rango visto en la simulación 45-50%.

La Figura 42 es el resultado final del algoritmo de infección ahorrando un 47% de operaciones. La Figura 43 y 44 muestran la evolución del algoritmo de infección en seis imágenes.

La Figura 45 muestra el resultado de aplicar el algoritmo de infección con el conjunto de reglas que ahorran un 47% de operaciones. Se señalan algunos de los problemas inherentes de la imagen donde no es posible hacer correspondencia. Mientras que en las partes donde se cuenta con textura (el muñeco de juguete) la correspondencia se efectúa bastante bien.

Descripción de las reglas

A continuación se explica cada una de las reglas y cómo se pasan a la tabla de reglas para el caso de ahorro del 47%. La visualización de las reglas de los demás casos se muestran en el apéndice B.

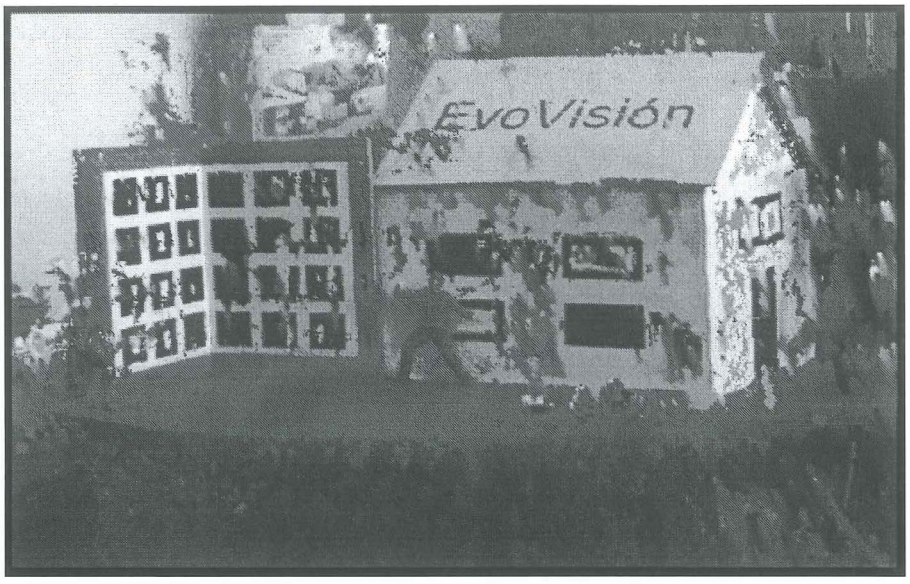
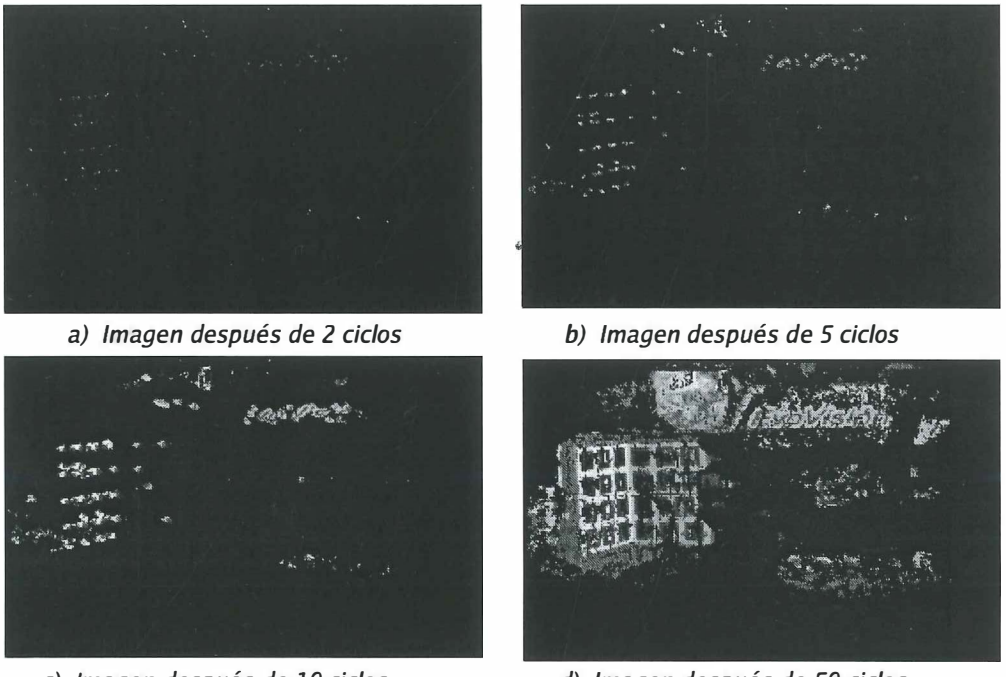


Figura 42: Experimento 47%. Resultado final del algoritmo de infección aplicando un filtro de mediana.



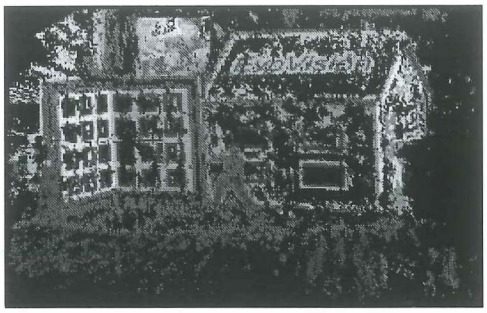
a) Imagen después de 2 ciclos

b) Imagen después de 5 ciclos

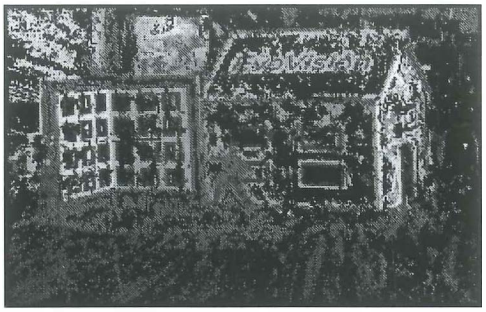
c) Imagen después de 10 ciclos

d) Imagen después de 50 ciclos

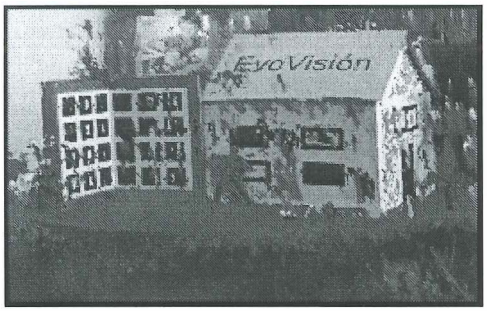
Figura 43: Cuatro etapas inciales de la evolución del algoritmo de infección.



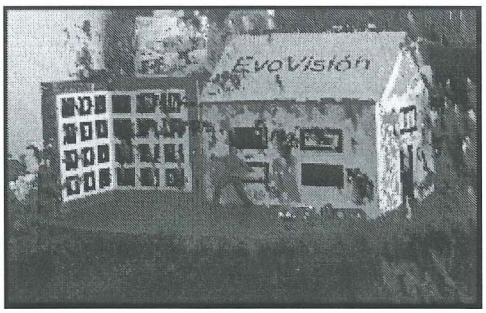
e) Imagen después de 150 ciclos



f) Imagen después de 250 ciclos



g) Imagen Final, ciclo 300. Se usó un Filtro de mediana. Resultado de nuestro algoritmo



h) Imagen Final, se usó un Filtro de mediana Resultado con un algoritmo de búsqueda exhaustiva.

Figura 44: Etapas Finales del algoritmo de infección y la imagen final del algoritmo de búsqueda exhaustiva.

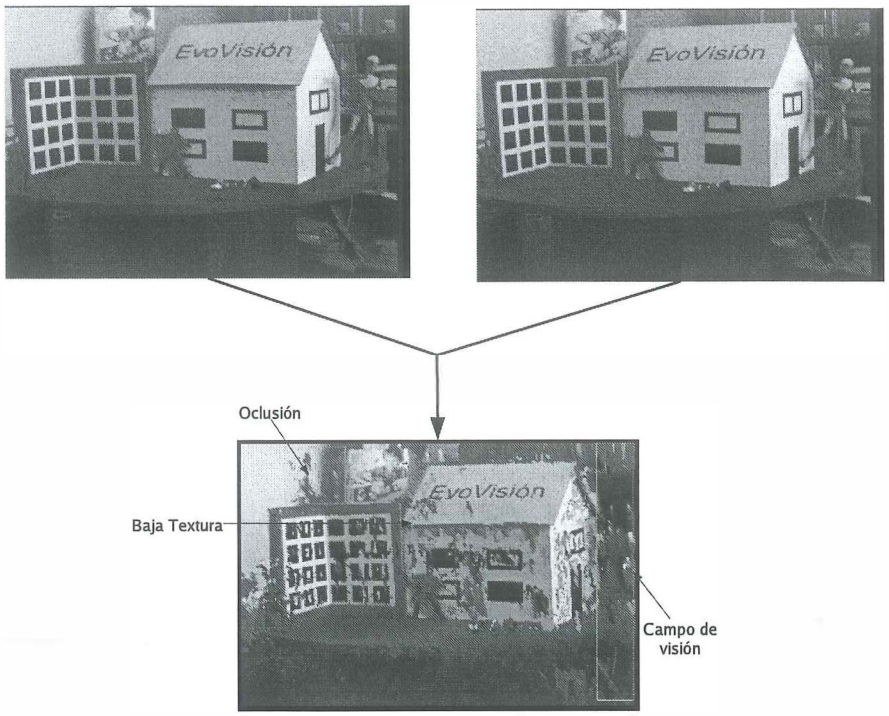


Figura 45: Imagen que resulta después de aplicar el algoritmo de infección.

- **Regla 1**. El propósito de esta regla es no realizar un cambio de estado cuando se visite una célula que haya sido EXPLOTADA (*Enferma*). El motivo es que el pixel de la imagen relacionado con la célula central fue evaluado anteriormente para encontrar su correspondiente en la imagen derecha, por lo que no es necesario hacer algún cambio. Para codificar la regla 1 en la tabla de reglas tomamos en cuenta el vecindario de la regla 1, ver Figura 46. Los estados de los vecinos que no son tomados en cuenta se usa la variable comodín en cada una de sus posiciones. De acuerdo al vecindario se acomodan los números correspondientes de los estados del vecindario en la tabla, tomando en cuenta que el valor numérico del comodín es 5 y del estado EXPLOTADO (*Enfermo*) es 1, ver la regla 1 en la Tabla X.

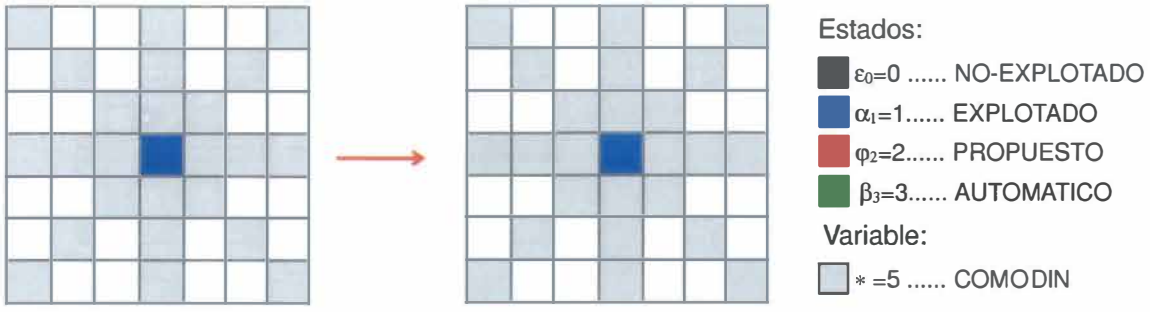


Figura 46: Transición de la Regla 1. El estado de la célula central no cambia dado que el pixel ya fue evaluado.

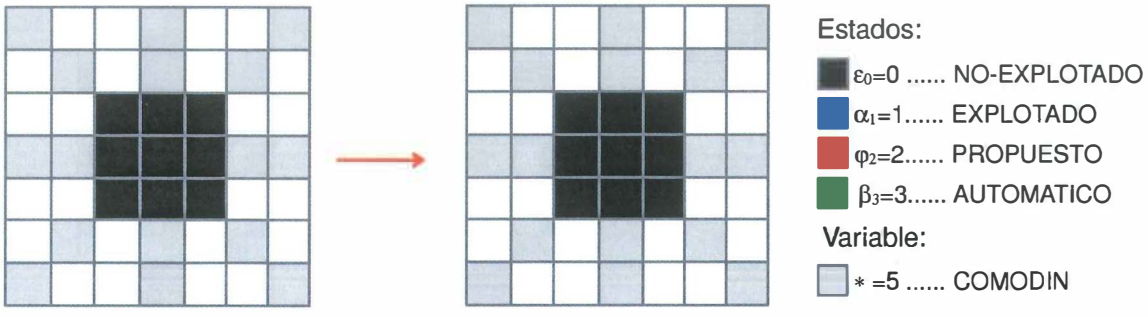


Figura 47: Transición de la Regla 2. El estado de la célula central no cambia, debido a que no hay información en el vecindario interno.

- **Regla 2**. La regla 2 indica que cuando el vecindario se encuentre vacío, esto es, que ninguna célula vecina haya sido evaluada, no se cambia el estado de la célula central, ver Figura 47. Para codificar la regla 2 en la tabla de reglas, se toma en cuenta el valor numérico que pertenece a cada estado en el vecindario y se codifica en la tabla.

- **Regla 3, 4, 5 y 6**. El propósito de estas cuatro reglas es crear la estructura inicial de la propagación. Las cuatro reglas crean una estructura en forma de cruz, para que la propagación empiece a efectuarse en todas direcciones. Estas reglas solo se usan en la inicialización de la propagación, el resto de las reglas son

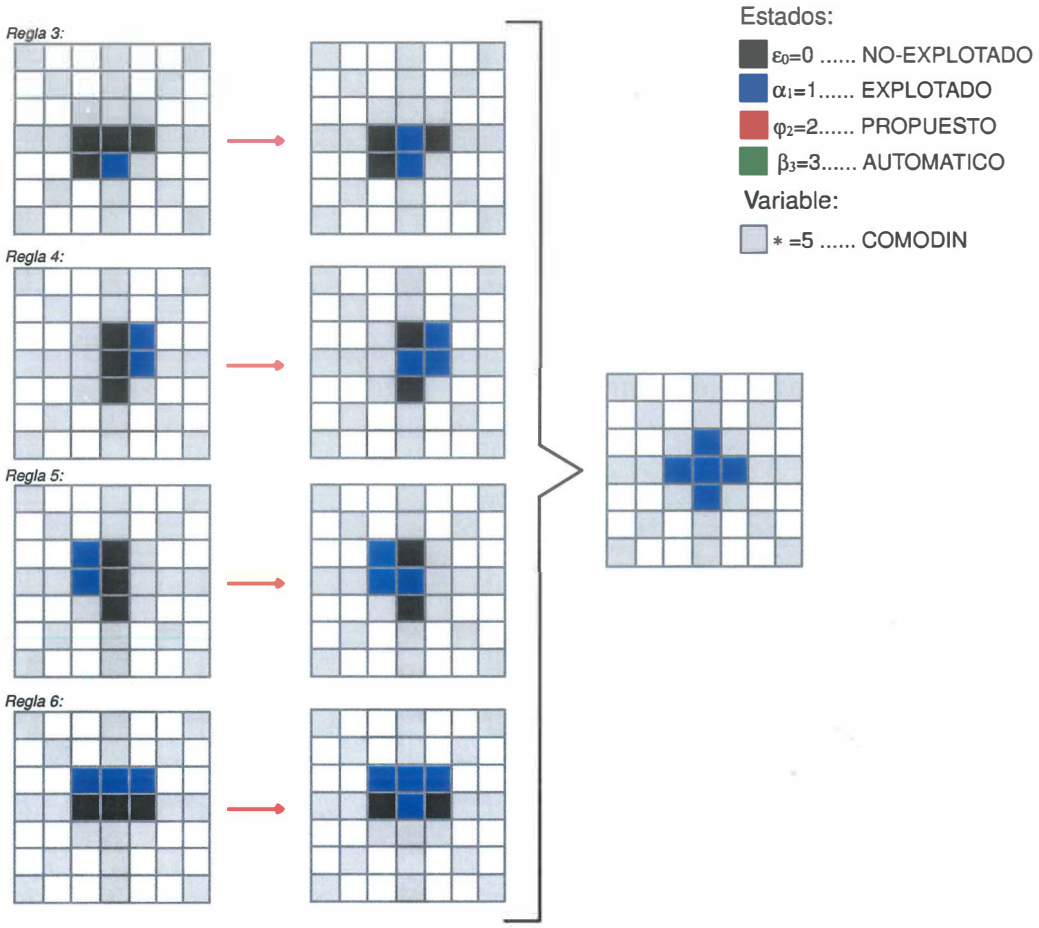


Figura 48: Transición de la Regla 3, 4, 5 y 6. Estas reglas en conjunto forman la estructura inicial de la propagación.

las que efectúan realmente el comportamiento del autómata. La codificación se realiza en base a los estados de cada célula en su representación numérica, ver Figura 48.

- **Regla 7 y 8**. La regla 7 y 8 aseguran evaluar pixeles en una región donde existan pixeles asignados automáticamente. La codificación a la tabla de reglas es de la misma manera que las demás reglas. La visualización de la regla 7 y 8 a través del vecindario se muestra en la Figura 49 y 50 respectivamente.

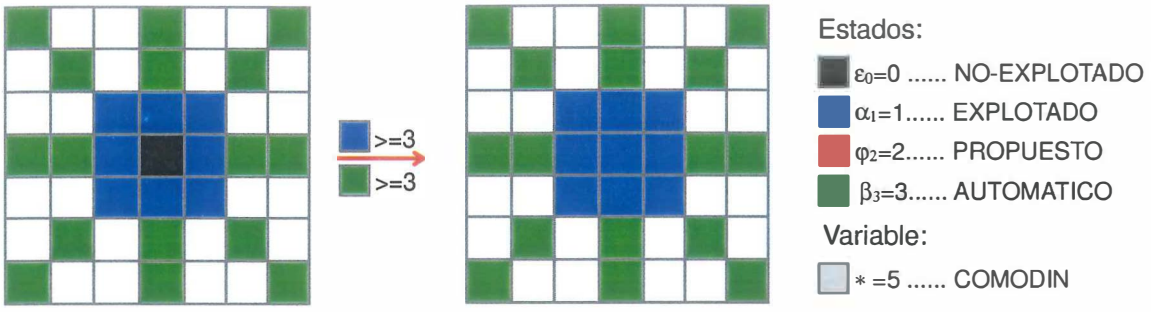


Figura 49: Transición de la Regla 7. Esta regla indica que se requieren mínimamente tres estados EXPLOTADOS (*Enfermos*)en el vecindario interno y tres estados AUTOMÁTICOS (*Inmunes*) en el vecindario externo.

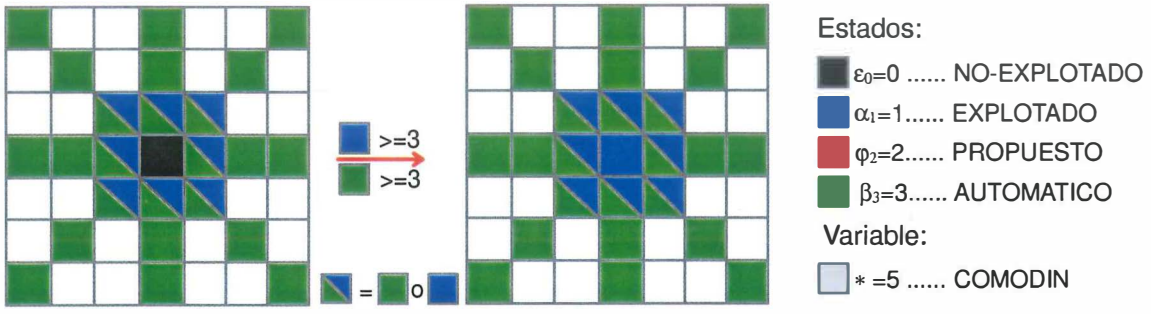


Figura 50: Transición de la Regla 8. Esta regla indica que se requieren mínimamente tres estados EXPLOTADOS (*Enfermos*)en el vecindario interno y tres estados AUTOMÁTICOS (*Inmunes*) en todo el vecindario.

- **Regla 9 y 10**. Estas dos reglas evitan que los estados PROPUESTOS (*Infectados*) se generen de manera lineal, que la propagación se realice en una sola dirección. La codificación de estas dos reglas se hace con los estados que están en las ocho posiciones alrededor de la célula central. La Figura 51 muestra la visualización de ambas reglas.

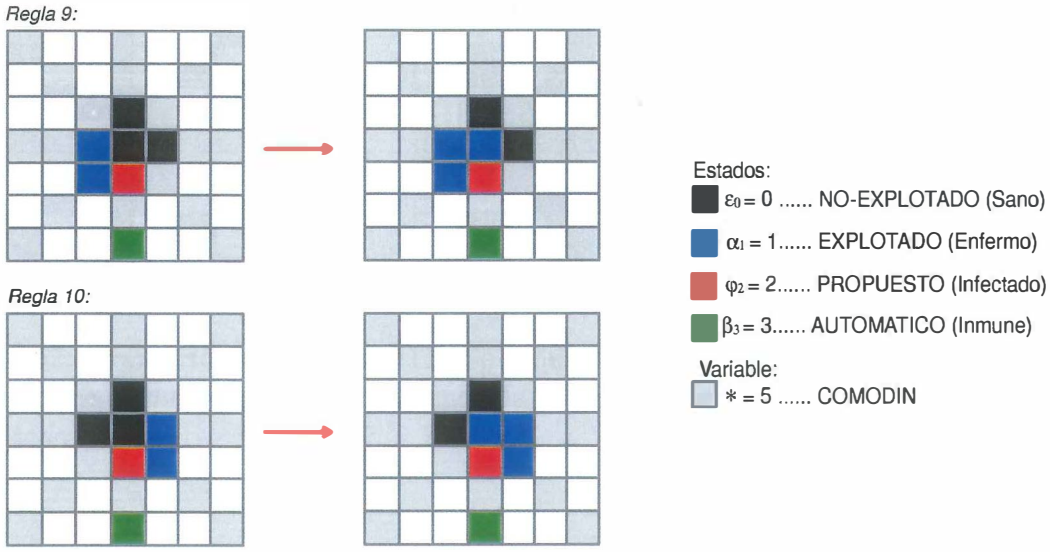


Figura 51: Transición de la Regla 9 y 10. Estas reglas evitan la propagación lineal de la asignación automática.

- **Regla 11**. La regla 11 indica que si la célula central está en estado NO-EXPLOTADO (*Sano*) y alrededor de esta célula se tiene al menos tres estados EXPLOTADOS (*Enfermos*) entonces, el estado de la célula central cambia a un estado PROPUESTO (*Infectado*), es decir, el pixel que está relacionado con la célula central se deja marcado para analizar posteriormente se realiza la asignación automática. La codificación de la regla 11 a la tabla de reglas se realiza agregando la cantidad estados EXPLOTADOS (*Enfermos*) en la columna VECEXP. En este caso, se utilizan tres estados EXPLOTADOS (*Enfermos*) como mínimo, ver Figura 52.

- **Regla 12**. La regla 12 indica que si la célula central se encuentra en un estado NO-EXPLOTADO (*Sano*) y alrededor de esta célula se tiene al menos tres estados PROPUESTOS (*Infectados*) entonces, el estado de la célula central cambia a un estado EXPLOTADO (*Enfermo*). Es decir, que el pixel relacionado con la célula

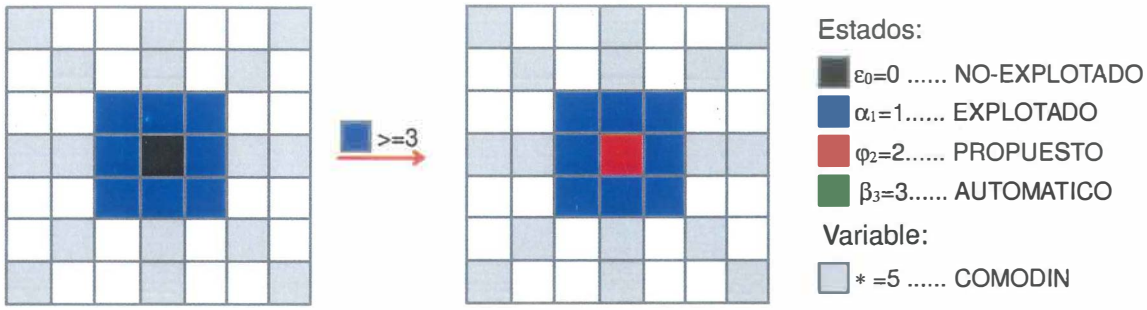


Figura 52: Transición de la Regla 11. Se requiere mínimamente tres estados EXPLOTADOS (*Enfermos*) para dejar marcada la célula central como estado PROPUESTO (*Infectado*).

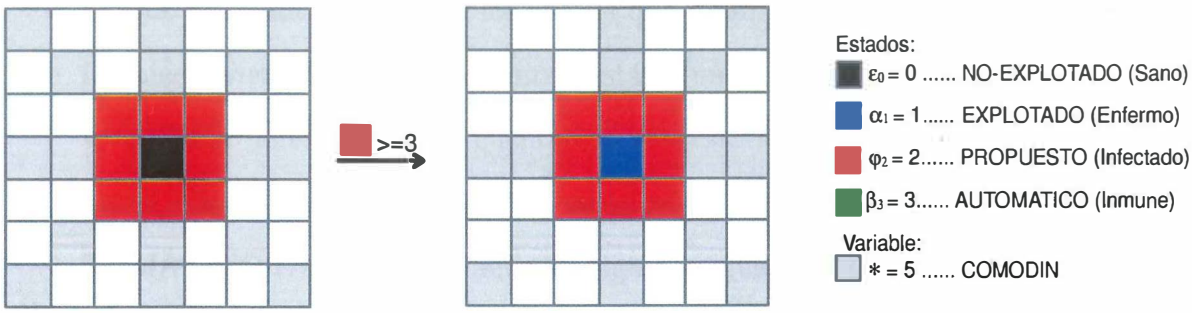


Figura 53: Transición de la Regla 12. Se requiere mínimamente tres estados PROPUESTOS (*Infectados*) para que el estado de la célula central cambie a un estado EXPLOTADO (*Enfermo*).

central, debe ser evaluado. La codificación de la regla 12 a la tabla de reglas se realiza agregando la cantidad de estados PROPUESTOS (*Infectados*) en la columna VECPROP. En este caso, se consideran tres estados PROPUESTOS (*Infectados*) como mínimo. Esta regla puede ser visualizada en la Figura 53 y ver la relación con la Tabla de reglas (X).

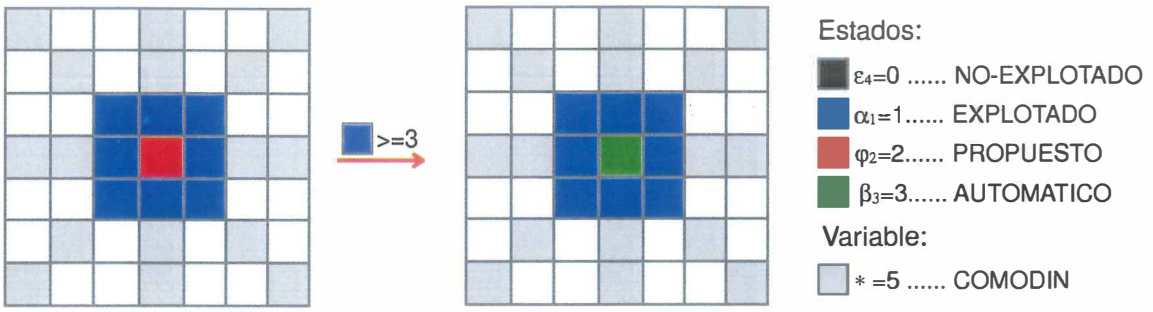


Figura 54: Regla 13. Se requiere mínimamente tres estados EXPLOTADOS (*Enfermos*) para que la célula central cambie a un estado AUTOMÁTICO (*Inmune*).

- **Regla 13**. Esta regla indica los pixeles que serán asignados automáticamente en la imagen virtual. Si la célula central está en un estado PROPUESTO (*Infectado*) y alrededor de esta célula se cuenta con tres estados EXPLOTADOS (*Enfermos*) como mínimo, entonces, el estado de la célula central cambia a un estado AUTOMÁTICO (*Inmune*). Lo anterior significa, que si se cuenta por lo menos con tres pixeles evaluados alrededor del pixel central que haya sido marcado, se asigna automáticamente su pixel correspondiente de la imagen derecha. La codificación de esta regla consiste en agregar en la columna VECEXP la cantidad de estados EXPLOTADOS (*Enfermos*) que se considerarán como mínimo. Cuando la regla 13 es verdadera, se condiciona a asignar automáticamente siempre y cuando la restricción de orientación se cumple. La restricción de orientación indica que si la linea epipolar de cada uno de los pixeles que han sido evaluados tienen la misma dirección, entonces, se procede a asignar automáticamente. De lo contrario, el pixel se evalúa. La Figura 54 muestra la regla 13.

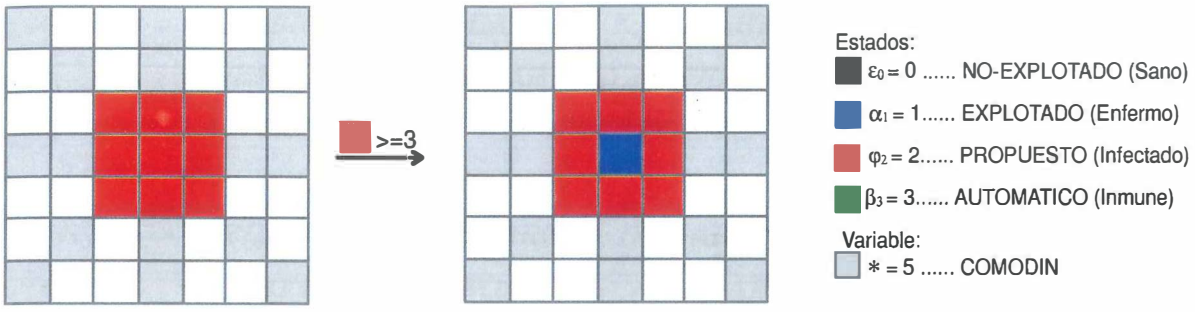


Figura 55: Regla 14. Se requiere mínimamente tres estados PROPUESTOS (*Infectados*) para que la célula central cambie a un estado EXPLOTADO (*Enfermo*).

- **Regla 14**. La regla 14 indica que si la célula central está en un estado PROPUESTO (*Infectado*) y alrededor de esta célula se tiene por lo menos tres estados PROPUESTOS (*Infectados*), entonces, la célula central cambia al estado ENFERMO (Explotado). El motivo de esta regla es que no es posible tener regiones con una gran cantidad de pixeles marcados sin haber hecho ninguna evaluación de alguno de ellos. La regla se muestra en la Figura 55.

Notación para las reglas que generan un 47% de ahorro en operaciones.

Cada una de las reglas es definida mediante Lógica de Predicados (Sección IV.8.1). A continuación, se muestran las catorce reglas para este caso en particular, aclarando que las cláusulas que contengan (*) siempre serán verdaderas dado que el estado de la célula en esa posición no nos interesa para esa regla.

$$\begin{aligned} R_1(\alpha_1) = \; & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\ & DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\ & \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\ & VECAUT(*) \rightarrow ACCION(\alpha_1) \end{aligned}$$

$$
\begin{aligned}
R_2(\varepsilon_0) = \; & IZQARR(\varepsilon_0) \wedge COLARR(\varepsilon_0) \wedge DERARR(\varepsilon_0) \wedge IZQFILA(\varepsilon_0) \wedge \\
& DERFILA(\varepsilon_0) \wedge IZQABA(\varepsilon_0) \wedge COLABA(\varepsilon_0) \wedge DERABA(\varepsilon_0) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\varepsilon_0) \\
R_3(\varepsilon_0) = \; & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(\varepsilon_0) \wedge \\
& DERFILA(\varepsilon_0) \wedge IZQABA(\varepsilon_0) \wedge COLABA(\varepsilon_0) \wedge DERABA(\alpha_1) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_4(\varepsilon_0) = \; & IZQARR(*) \wedge COLARR(\varepsilon_0) \wedge DERARR(\alpha_1) \wedge IZQFILA(*) \wedge \\
& DERFILA(\alpha_1) \wedge IZQABA(*) \wedge COLABA(\varepsilon_0) \wedge DERABA(*) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_5(\varepsilon_0) = \; & IZQARR(\alpha_1) \wedge COLARR(\varepsilon_0) \wedge DERARR(*) \wedge IZQFILA(\alpha_1) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(\varepsilon_0) \wedge DERABA(*) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_6(\varepsilon_0) = \; & IZQARR(\alpha_1) \wedge COLARR(\alpha_1) \wedge DERARR(\alpha_1) \wedge IZQFILA(\varepsilon_0) \wedge \\
& DERFILA(\varepsilon_0) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_7(\varepsilon_0) = \; & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(3) \wedge VECPROP(*) \wedge VECEXT(3) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_8(\varepsilon_0) = \; & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(3) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(3) \rightarrow ACCION(\alpha_1)
\end{aligned}
$$

$$
\begin{aligned}
R_9(\varepsilon_0) = \quad & IZQARR(*) \wedge COLARR(\varepsilon_0) \wedge DERARR(*) \wedge IZQFILA(\alpha_1) \wedge \\
& DERFILA(\varepsilon_0) \wedge IZQABA(\alpha_1) \wedge COLABA(\varphi_2) \wedge DERABA(*) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_{10}(\varepsilon_0) = \quad & IZQARR(*) \wedge COLARR(\varepsilon_0) \wedge DERARR(*) \wedge IZQFILA(\varepsilon_0) \wedge \\
& DERFILA(\alpha_1) \wedge IZQABA(*) \wedge COLABA(\varphi_2) \wedge DERABA(\alpha_1) \\
& \wedge VECEXP(*) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_{11}(\varepsilon_0) = \quad & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(3) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\varphi_2) \\
R_{12}(\varepsilon_0) = \quad & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(*) \wedge VECPROP(3) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1) \\
R_{13}(\varphi_2) = \quad & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(3) \wedge VECPROP(*) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\beta_3) \\
R_{14}(\varphi_2) = \quad & IZQARR(*) \wedge COLARR(*) \wedge DERARR(*) \wedge IZQFILA(*) \wedge \\
& DERFILA(*) \wedge IZQABA(*) \wedge COLABA(*) \wedge DERABA(*) \\
& \wedge VECEXP(*) \wedge VECPROP(3) \wedge VECEXT(*) \wedge \\
& VECAUT(*) \rightarrow ACCION(\alpha_1)
\end{aligned}
$$

La tabla de reglas puede ser fácilmente modificada agregando o quitando reglas, o bien, modificando las mismas reglas. El orden de las reglas es importante, la primera regla que se cumpla es la primera que se aplica. Por ello, deben estar ordenadas de lo particular a lo general, es decir, las reglas más específicas van al principio y las reglas más generales van al final.

Tabla X: Tabla de reglas que genera un 47% de ahorro en operaciones.

C	#	IZQ ARR	COL ARR	DER ARR	IZQ FILA	DER FILA	IZQ ABA	COL ABA	DER ABA	VEC EXP	VEC PROP	VEC EXT	VEC AUT	A
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	1
0	8	5	5	5	5	5	5	5	5	3	5	5	3	1
0	9	5	0	5	1	0	1	2	5	5	5	5	5	1
0	10	5	0	5	0	1	5	2	1	5	5	5	5	1
0	11	5	5	5	5	5	5	5	5	3	5	5	5	2
0	12	5	5	5	5	5	5	5	5	5	3	5	5	1
2	13	5	5	5	5	5	5	5	5	3	5	5	5	3
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1

Columna (C) indica el valor del estado de la célula central.
Columna (#) indica el número de regla.
Columna 3-10, indican el valor del estado en una posición específica del vecindario.
Columna 11-14, indican la cantidad de estados que se repiten en el vecindario.
Columna (A), indica el valor del estado que obtendrá la célula central cuando la regla sea verdadera.

La Tabla X muestra el conjunto de reglas que permiten un ahorro en operaciones de un 47% con respecto al método exhaustivo. Al modificar las reglas 7-14 se obtuvo 4 tablas más, que permiten ahorrar 0%, 70%, 78%, y 99% en cálculo. Las reglas de la 1-6 no se modificaron porque las primeras dos son esenciales y las otras cuatro generan la estructura inicial.

La primera columna (C) de la Tabla X representa el estado de la célula central del vecindario en el paso de tiempo t. La segunda columna indica el número de regla. De la tercera a la décima columna se encuentran los estados de las ocho posiciones centrales

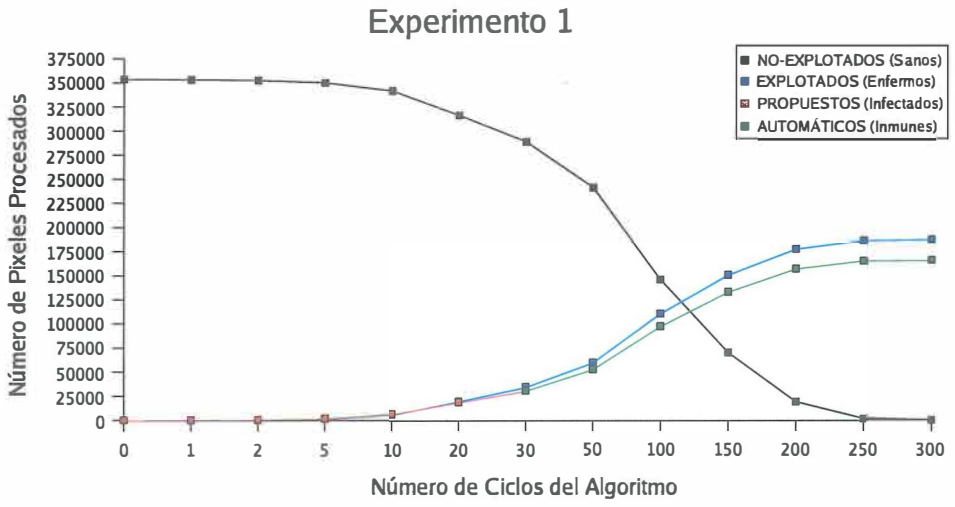


Figura 56: Experimento 47%. Gráfica de resultados de la evolución de cada uno de los estados.

del vecindario alrededor de la célula central, ver Figura 32.

La columna VECEXP representa la cantidad de células que tengan el estado EXPLOTADO (*Enfermo*) dentro del vecindario, excluyendo la célula central. La columna VECPROP contiene el número de estados PROPUESTOS (*Infectados*) que se encuentren dentro del vecindario. La columna VECEXT contiene el número de estados AUTOMÁTICOS (*Inmunes*) que se encuentren solamente en la parte exterior del vecindario, ver Figura 33.

La columna VECAUT representa la cantidad de estados AUTOMÁTICOS (*Inmunes*) que se encuentren solamente en las ocho posiciones centrales del vecindario alrededor de la célula central, ver Figura 32. La última columna (A) representa el nuevo valor del estado de la célula central cuando la regla sea verdadera.

Tabla XI: Experimento 47%. Datos generados por el algoritmo de infección

CICLO	COLUMNA-1	COLUMNA-2	COLUMNA-3	COLUMNA-4
0	353526	0	0	0
1	353471	452	161	0
2	352858	578	549	161
5	350545	1767	2255	1690
10	341881	6665	7250	5977
20	316448	20310	19465	18056
30	288861	34951	32375	31155
50	241116	60198	54279	53281
100	145904	110845	98433	97624
150	70456	150649	133660	133052
200	19533	177336	157304	156944
250	2112	186276	165156	165118
300	97	187354	166030	166021

Hora Inicial:Tue Mar 16 16:56:40 2004
Hora Final:Tue Mar 16 17:06:57 2004

Tiempo transcurrido: **00:10:17** Contra: 00:17:18 del método Exhaustivo

Porcentaje de Ahorro en operaciones: $(166021 \times 100) \div 353626 = 46.9\%$
Porcentaje de Ahorro en tiempo: 41%

COLUMNA-1:NO-EXPLOTADOS (*Sanos*)
COLUMNA-2:EXPLOTADOS (*Enfermos*)
COLUMNA-3:PROPUESTOS (*Infectados*)
COLUMNA-4:AUTOMÁTICOS (*Inmunes*)

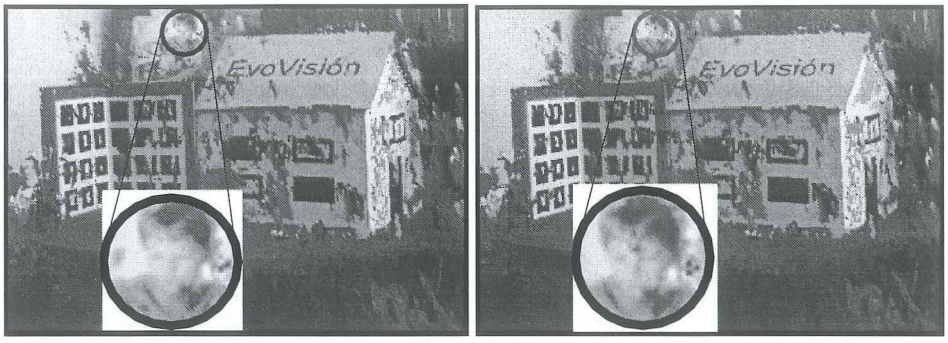


Imagen Final, resultado de nuestro algoritmo *Imagen Final, resultado de una búsqueda exahustiva*

Figura 57: La región del recuadro se mejora en relación con el resultado del método exhaustivo.

El resultado de la evolución del algoritmo de infección ahorrando un 47% de operaciones se muestra gráficamente en la Figura 56. La línea que representa al estado EXPLOTADO (*Enfermo*) se encuentra por encima de la línea del estado AUTOMÁTICO (*Inmune*) ya que el número de pixeles evaluados fue ligeramente mayor a los pixeles que fueron asignados automáticamente en la imagen derecha. La Tabla XI muestra la evolución numérica de cada uno de los estados y el tiempo que tardó el algoritmo en llevar a cabo el proceso de correspondencia. El algoritmo ahorró un 41% en tiempo en comparación al algoritmo de búsqueda exhaustiva.

Se puede observar en la Figura (57) y (58) que la calidad de los resultados se mejora ligeramente en comparación con los resultados producidos por el algoritmo de búsqueda exhaustiva, haciendo hincapié que nuestro algoritmo ahorró un 41% en tiempo y no empeoró la calidad de la imagen. Cuando se tiene objetos con buena textura dentro de la escena, la correlación es más eficiente, por lo tanto la correspondencia se facilita, ver los recuadros de la Figura 59.

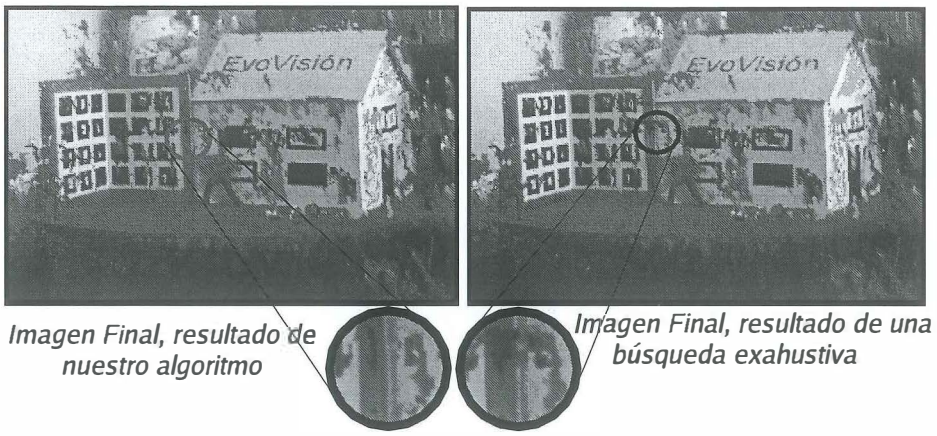


Figura 58: La región lateral a la mira de calibración se mejora ligeramente.

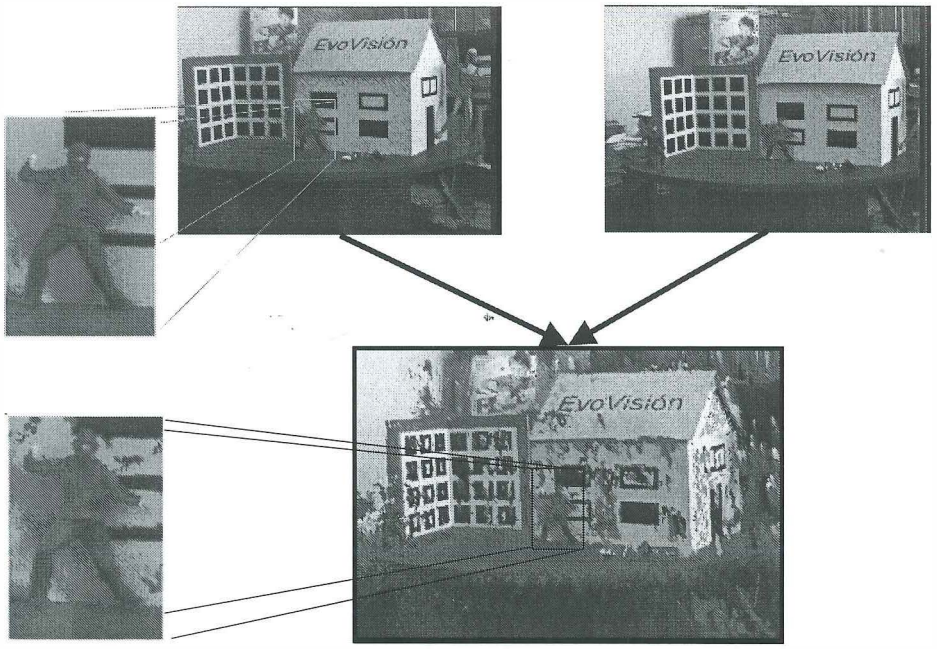


Figura 59: Se aprecia la comparación entre una región de la imagen original y una región de la imagen producida por el algoritmo cuando existe buena textura en el par estéreo.

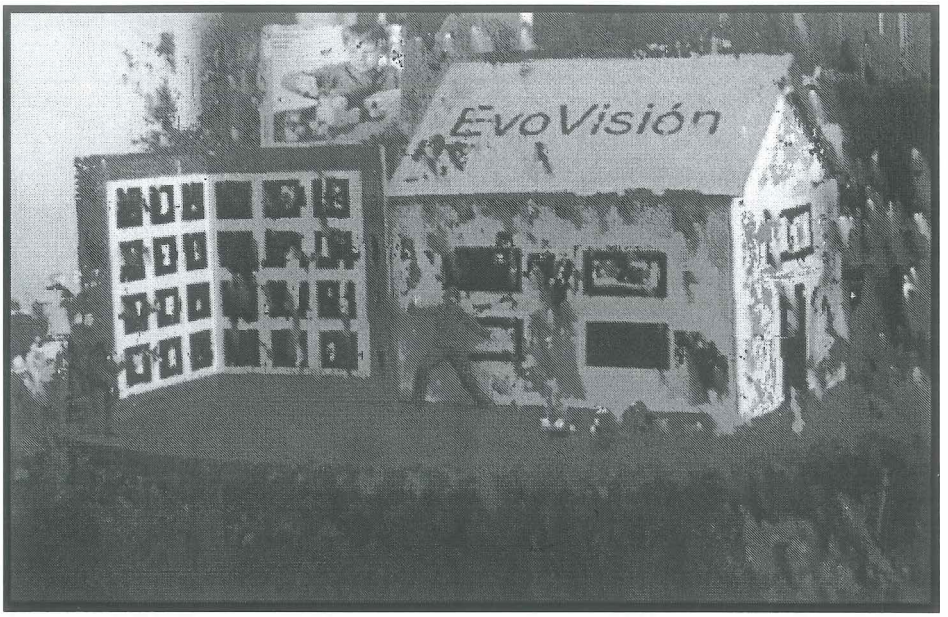


Figura 60: Experimento 0%. Imagen que resulta después de aplicar el algoritmo de infección a la cual se le aplicó un filtro de mediana.

V.3.2 Experimento 2: Caso del 0% de ahorro

Este experimento representa la búsqueda exhaustiva del algoritmo de correspondencias. Las reglas fueron modificadas para que la acción de cada regla indicara evaluación del pixel (ver Tabla XII) y de esta forma, no realizar ninguna asignación automática. La imagen generada por el algoritmo de infección y la imagen generada por el algoritmo de búsqueda exhaustiva son muy similares, ver Figura 60 y 61.

La Tabla XII genera un 0% de ahorro en operaciones lo que es equivalente a realizar la correspondencia con un algoritmo de búsqueda exhaustiva. Se modificaron las reglas 11, 13 y 14 indicando que se debe evaluar en todo momento y se redujo el número de estados EXPLOTADOS (*Enfermos*). En ningún momento se genera el estado PROPUESTO (*Infectado*) y el estado AUTOMÁTICO (*Inmune*), por tal motivo, las dos últimas reglas se quitaron de la tabla.

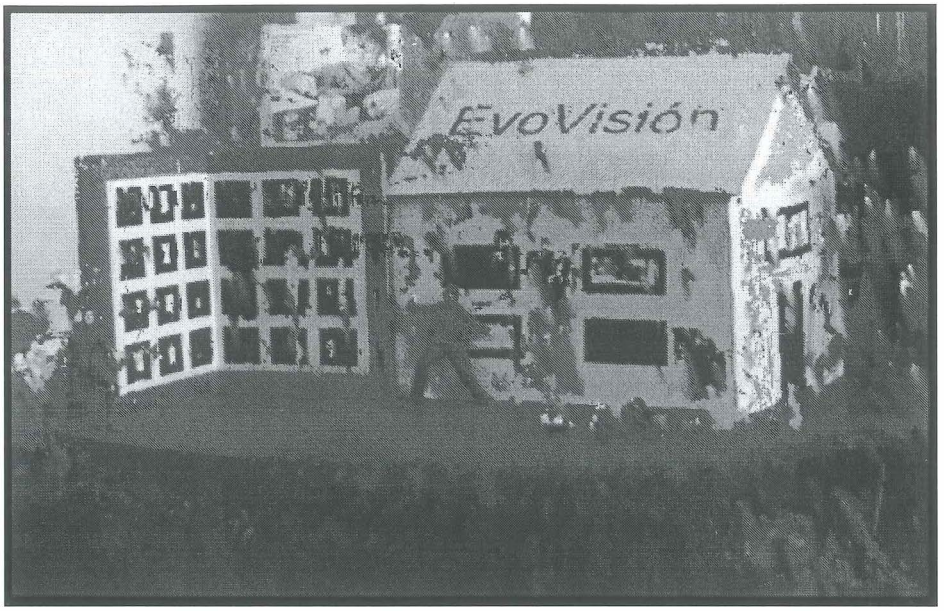


Figura 61: Imagen que resulta de aplicar del algoritmo de búsqueda exhaustiva a la cual se le aplicó un filtro de mediana.

Tabla XII: Tabla de reglas que genera un 0% de ahorro en operaciones

C	#	IZQ ARR	COL ARR	DER ARR	IZQ FILA	DER FILA	IZQ ABA	COL ABA	DER ABA	VEC EXP	VEC PROP	VEC EXT	VEC AUT	A
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	1
0	8	5	5	5	5	5	5	5	5	3	5	5	3	1
0	9	5	0	5	1	0	1	2	5	5	5	5	5	1
0	10	5	0	5	0	1	5	2	1	5	5	5	5	1
0	11	5	5	5	5	5	5	5	5	1	5	5	5	1
0	12	5	5	5	5	5	5	5	5	5	3	5	5	1

Columna (C) indica el valor del estado de la célula central.
Columna (#) indica el número de regla.
Columna 3-10, indican el valor del estado en una posición específica del vecindario.
Columna 11-14, indican la cantidad de estados que se repiten en el vecindario.
Columna (A), indica el valor del estado que obtendrá la célula central cuando la regla sea verdadera.

Tabla XIII: Experimento del 0%. Datos generados por el algoritmo de infección.

CICLO	COLUMNA-1	COLUMNA-2	COLUMNA-3	COLUMNA-4
0	0	0	0	0
1	353471	319278	0	0
2	34193	320004	0	0
5	32216	321777	0	0
10	29646	324310	0	0
20	24976	328940	0	0
30	20706	333170	0	0
50	13366	340430	0	0
100	2016	351580	0	0
150	0	353471	0	0
200	0	353471	0	0
250	0	353471	0	0
300	0	353471	0	0

Hora Inicial: Tue Mar 16 20:10:55 2004
Hora Final: Tue Mar 16 20:26:50 2004

Tiempo transcurrido: **00:15:55** Contra: 00:17:18 del método Exhaustivo

Porcentaje de Ahorro en operaciones: 0%
Porcentaje de Ahorro en tiempo: 8%
COLUMNA-1:NO-EXPLOTADOS (*Sanos*)
COLUMNA-2:EXPLOTADOS (*Enfermos*)
COLUMNA-3:PROPUESTOS (*Infectados*)
COLUMNA-4:AUTOMÁTICOS (*Inmunes*)

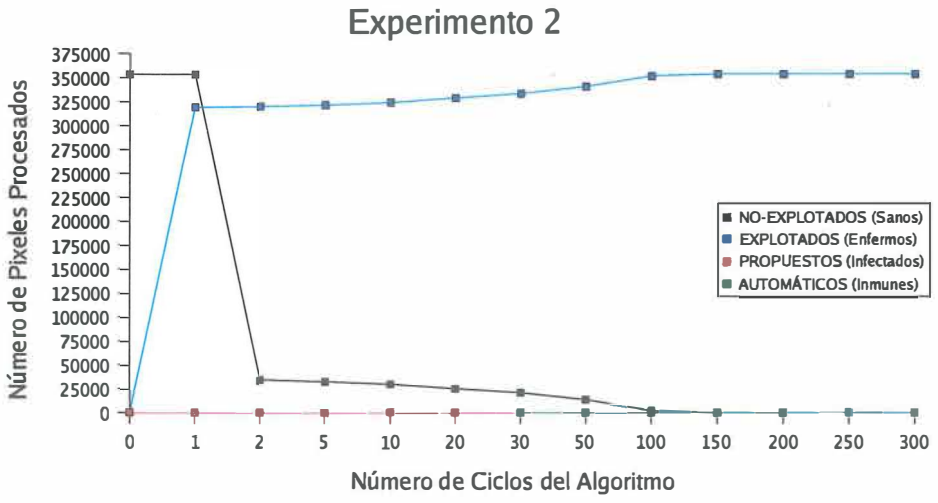


Figura 62: Experimento 0%. Gráfica de resultados de la evolución de cada uno de los estados.

La Figura 62 muestra la gráfica de la evolución de cada uno de los estados cuando se realiza la evaluación de todos los pixeles de la imagen.

El numero de pixeles NO-EXPLOTADOS (*Sanos*) decrementa considerablemente a partir del primer ciclo hasta llegar a cero después del ciclo número 100. Por el contrario, los pixeles EXPLOTADOS (*Enfermos*) aumentan considerablente durante el primer ciclo debido a que todas las reglas indican evaluación del pixel, ver gráfica en la Figura 62. La Tabla XIII muestra un ahorro de operaciones del 0%, mientras que en tiempo ahorra un 8%, esto es debido a que el método exhaustivo busca cada uno de los pixeles de la imagen izquierda su correspondiente en la imagen derecha. Cuando la correlación de un pixel es baja, el método exhaustivo realiza de nuevo la búsqueda, ahora de derecha a izquierda, por ello, se ahorra en tiempo y no en operaciones. Lo anterior indica que nuestro algoritmo es más rápido aún cuando se evalúa cada pixel.

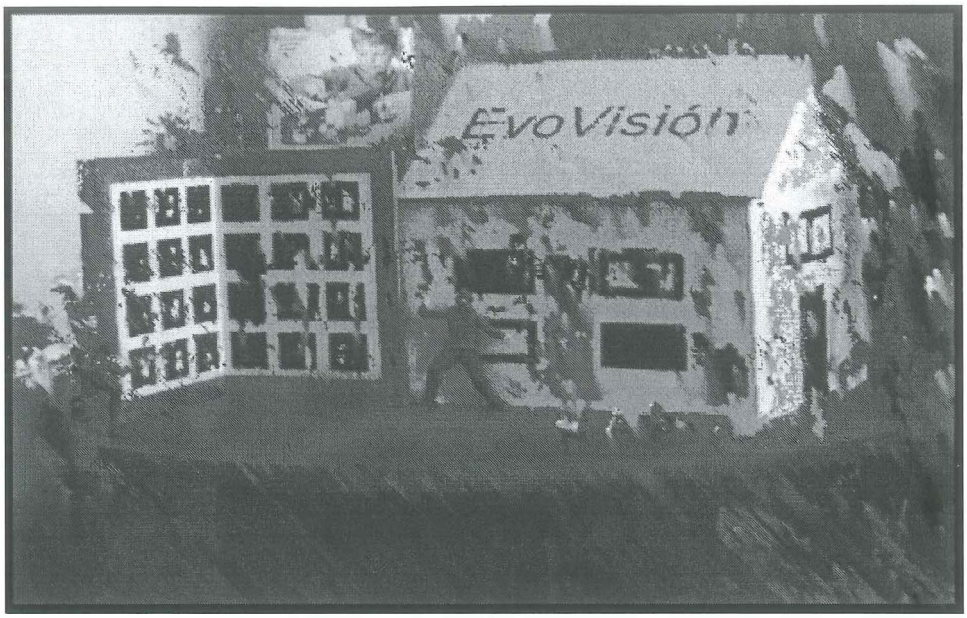


Figura 63: Experimento 70%. Imagen resultante del algoritmo de infección a la cual se le aplicó un filtro de mediana.

Las curvas que representan a los pixeles AUTOMÁTICOS (*Inmunes*) y PROPUESTOS (*Infectados*) siempre se encuentran en cero ya que ninguna regla permite proponer o asignar automáticamente.

V.3.3 Experimento 3: Caso del 70% de ahorro

Este experimento se realizó modificando el conjunto de reglas mostradas en la simulación. Los cambios realizados al conjunto de reglas permitió un ahorro del 69.9%. La Figura 63 es la imagen que resulta de aplicar el algoritmo de infección con las reglas modificadas. Se muestra mejoras en ciertas partes de la imagen y deterioro en otras partes. La evolución de los pixeles sobre la imagen se muestra en la Figura 64.

La Tabla XIV genera un 70% de ahorro en operaciones. Se modificaron las reglas 11

Tabla XIV: Tabla de reglas que genera un 70% de ahorro en operaciones

C	#	IZQ ARR	COL ARR	DER ARR	IZQ FILA	DER FILA	IZQ ABA	COL ABA	DER ABA	VEC EXP	VEC PROP	VEC EXT	VEC AUT	A
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	1
0	8	5	5	5	5	5	5	5	5	3	5	5	3	1
0	9	5	0	5	1	0	1	2	5	5	5	5	5	1
0	10	5	0	5	0	1	5	2	1	5	5	5	5	1
0	11	5	5	5	5	5	5	5	5	2	5	5	5	2
0	12	5	5	5	5	5	5	5	5	5	3	5	5	1
2	13	5	5	5	5	5	5	5	5	2	5	5	5	3
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1

Columna (C) indica el valor del estado de la célula central.
Columna (#) indica el número de regla.
Columna 3-10, indican el valor del estado en una posición específica del vecindario.
Columna 11-14, indican la cantidad de estados que se repiten en el vecindario.
Columna (A), indica el valor del estado que obtendrá la célula central cuando la regla sea verdadera.

y 13 disminuyendo la cantidad de vecinos EXPLOTADOS (*Enfermos*) y PROPUESTOS (*Infectados*).

En la Figura 64 se observa que la curva de los pixeles AUTOMÁTICOS (*Inmunes*) se encuentran por encima de la curva de los pixeles EXPLOTADOS (*Enfermos*). En la Tabla XV se puede observar el comportamiento numérico de la evolución, el decremento de los pixeles NO-EXPLOTADOS (*Sanos*) y el aumento de los pixeles que se van asignando automáticamente. El algoritmo ahorra un 58% en tiempo de proceso evaluando el 30% de los pixeles.

Tabla XV: Experimento 70%. Datos generados por el algoritmo de infección

CICLO	COLUMNA-1	COLUMNA-2	COLUMNA-3	COLUMNA-4
0	353471	0	0	0
1	353471	263	533	0
2	352675	449	1325	530
5	346458	3448	6719	4685
10	327694	9906	21009	17582
20	273245	28979	57261	53321
30	237404	37914	80600	78795
50	131632	79308	144149	142881
100	63330	92910	198387	197409
150	16943	100590	236598	236004
200	4093	104144	245331	245239
250	577	105818	247109	247052
300	0	106021	247450	247412

Hora Inicial: Tue Mar 16 21:40:55 2004
Hora Final: Tue Mar 16 21:48:10 2004

Tiempo transcurrido: **00:07:15** Contra: 00:17:18 del método Exhaustivo

Porcentaje de Ahorro en operaciones: $(247412 \times 100) \div 353626 = 69.9\%$
Porcentaje de Ahorro en tiempo: 58%

COLUMNA-1:NO-EXPLOTADOS (*Sanos*)
COLUMNA-2:EXPLOTADOS (*Enfermos*)
COLUMNA-3:PROPUESTOS (*Infectados*)
COLUMNA-4:AUTOMÁTICOS (*Inmunes*)

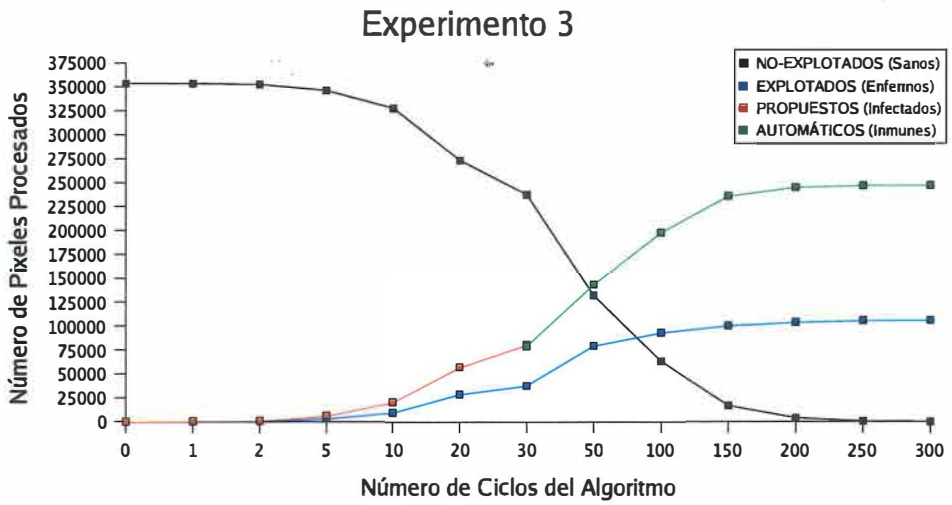


Figura 64: Experimento 70%. Gráfica de resultados de la evolución de cada uno de los estados.

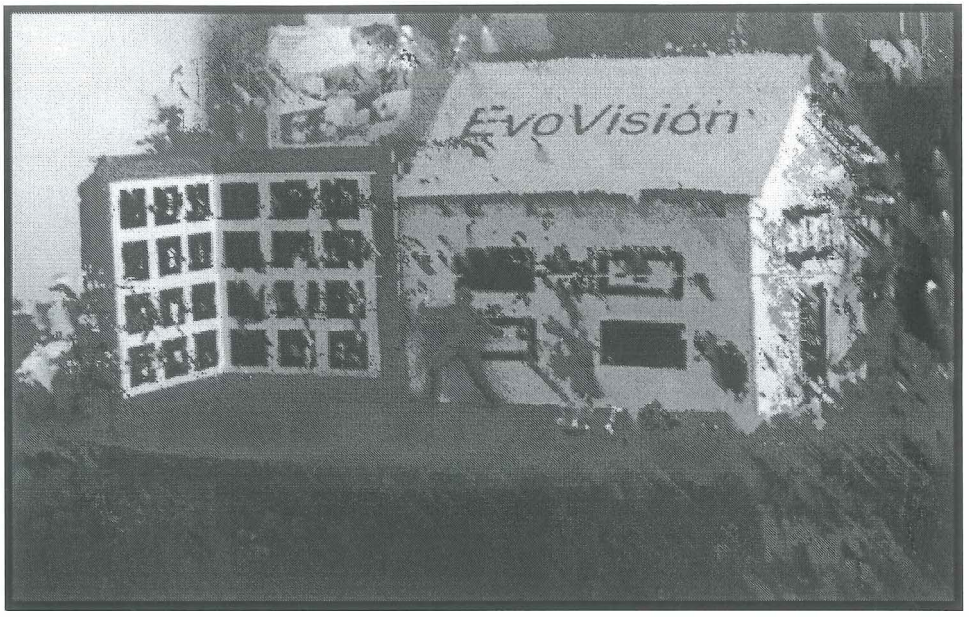


Figura 65: Experimento 78%. Imagen resultante del algoritmo de infección a la cual se le aplicó un filtro de mediana.

V.3.4 Experimento 4: Caso del 78% de ahorro

Los cambios realizados para este experimento fue principalmente a la regla 7,8,9,10 y 11 del conjunto de reglas originadas en la simulación para el caso del 47% de ahorro. En la tabla de reglas (Tabla XVI) se aumenta el número de pixeles PROPUESTOS (*Infectados*) y con ello aumenta el número de pixeles AUTOMÁTICOS (*Inmunes*). La Figura 65 es el resultado de aplicar el algoritmo de infección con el conjunto de reglas mostradas en la Tabla XVI.

La Tabla XVI genera un 78% de ahorro en operaciones. Los parámetros que se cambiaron fue especialmente en las reglas de la 7-11 para aumentar el número de estados PROPUESTOS (*Infectados*) permitiendo con ello obtener un mayor número de estados AUTOMÁTICOS (*Inmunes*).

Tabla XVI: Tabla de reglas que genera un 78% de ahorro en operaciones

C	#	IZQ ARR	COL ARR	DER ARR	IZQ FILA	DER FILA	IZQ ABA	COL ABA	DER ABA	VEC EXP	VEC PROP	VEC EXT	VEC AUT	A
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	2
0	8	5	5	5	5	5	5	5	5	3	5	5	3	2
0	9	5	0	5	1	0	1	2	5	5	5	5	5	2
0	10	5	0	5	0	1	5	2	1	5	5	5	5	2
0	11	5	5	5	5	5	5	5	5	1	5	5	5	2
0	12	5	5	5	5	5	5	5	5	5	3	5	5	1
2	13	5	5	5	5	5	5	5	5	3	1	5	5	3
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1

Columna (C) indica el valor del estado de la célula central.
Columna (#) indica el número de regla.
Columna 3-10, indican el valor del estado en una posición específica del vecindario.
Columna 11-14, indican la cantidad de estados que se repiten en el vecindario.
Columna (A), indica el valor del estado que obtendrá la célula central cuando la regla sea verdadera.

Tabla XVII: Experimento 78%. Datos generados por el algoritmo de infección

CICLO	COLUMNA-1	COLUMNA-2	COLUMNA-3	COLUMNA-4
0	353526	0	0	0
1	353471	27421	86761	0
2	239289	33869	111718	85845
5	177427	41116	147535	132277
10	146377	45128	167589	157195
20	112388	51125	195191	184511
30	84280	56189	218544	207238
50	36435	64374	257793	246667
100	12675	68523	276118	265996
150	3757	70608	283168	272782
200	95	71491	286020	275511
250	0	71510	286083	275577
300	0	71510	286083	275577

Hora Inicial:Tue Apr 6 19:26:57 2004
Hora Final:Tue Apr 6 19:32:34 2004

Tiempo transcurrido: **00:05:37** Contra: 00:17:18 del método Exhaustivo

Porcentaje de Ahorro en operaciones: $(275577 \times 100) \div 353626 = 77.9\%$
Porcentaje de Ahorro en tiempo: 68%

COLUMNA-1:NO-EXPLOTADOS (*Sanos*)
COLUMNA-2:EXPLOTADOS (*Enfermos*)
COLUMNA-3:PROPUESTOS (*Infectados*)
COLUMNA-4:AUTOMÁTICOS (*Inmunes*)

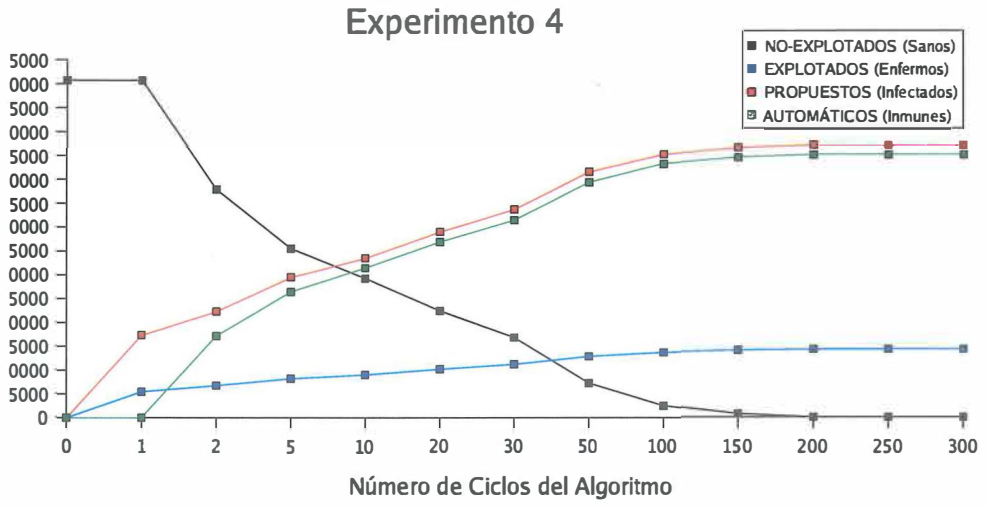


Figura 66: Experimento 78%. Gráfica de resultados de la evolución de cada uno de los estados.

En la Tabla XVII se puede ver númericamente el número de pixeles que van decrementado y aumentando para cada uno de los estados durante el proceso de evolución y el tiempo que se tarda el algoritmo en realizar la correspondencia. Se puede observar que el algoritmo al evaluar el 22% de pixeles en la imagen izquierda, se ahorra un 68% en procesamiento.

La curva que representa los pixeles AUTOMÁTICOS (*Inmunes*) en la Figura 66 está muy por encima de la curva de los pixeles EXPLOTADOS (*Enfermos*) debido a que el porcentaje de ahorro es de un 78%. Por otro lado, se visualiza el decremento de los pixeles NO-EXPLOTADOS (*Sanos*) durante el transcurso de la propagación.

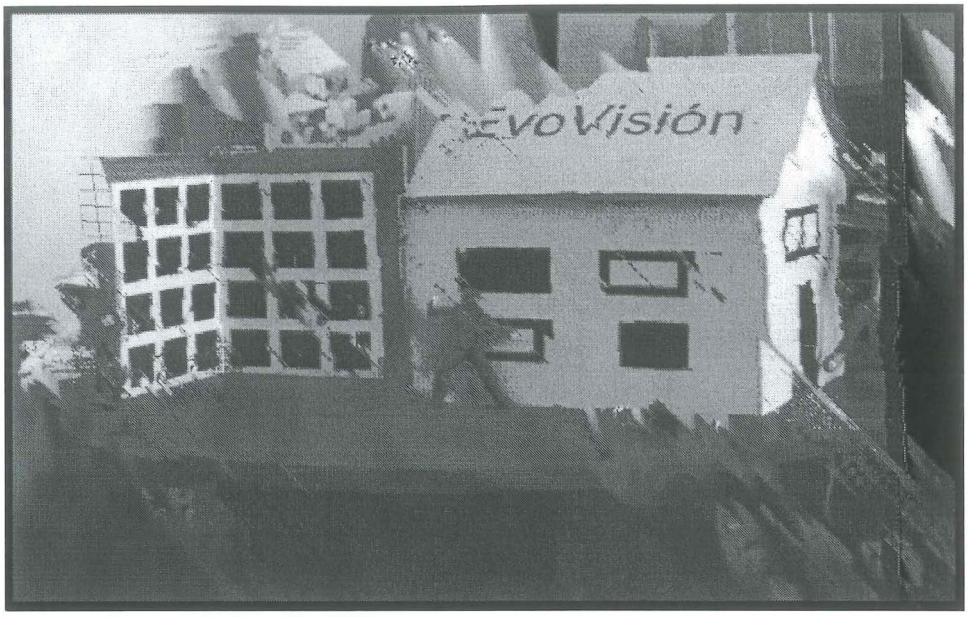


Figura 67: Experimento 99%. Imagen resultante del algoritmo de infección a la cual se le aplicó un filtro de mediana.

V.3.5 Experimento 5: Caso del 99% de ahorro

Este experimento se realizó para verificar los resultados del algoritmo con un mínimo número de cálculos. La Figura 67 es la imagen que resulta después de aplicar el algoritmo con un ahorro de 99% de operaciones. Se puede apreciar que existen partes en la imagen con muy buena calidad y otras donde algunos objetos no se reconstruyen por completo.

La Tabla XVIII genera un 99% de ahorro en operaciones. Se modifican las reglas de tal modo que solamente se evalúe la estructura inicial que se forma de las reglas 3-6 y se evalúe el pixel que cumpla con la regla 14.

La Tabla XIX expresa numéricamente la evolución de cada uno de los estados y el tiempo que tardó el algoritmo en realizar la correspondencia. El algoritmo ahorra

Tabla XVIII: Tabla de reglas que genera un 99% de ahorro en operaciones

C	#	IZQ ARR	COL ARR	DER ARR	IZQ FILA	DER FILA	IZQ ABA	COL ABA	DER ABA	VEC EXP	VEC PROP	VEC EXT	VEC AUT	A
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1
0	2	0	0	0	0	0	0	0	0	5	5	5	5	0
0	3	5	5	5	0	0	0	1	5	5	5	5	5	1
0	4	5	0	1	5	1	5	0	5	5	5	5	5	1
0	5	1	0	5	1	5	5	0	5	5	5	5	5	1
0	6	1	1	1	0	0	5	5	5	5	5	5	5	1
0	7	5	5	5	5	5	5	5	5	3	5	3	5	2
0	8	5	5	5	5	5	5	5	5	3	5	5	3	2
0	9	5	0	5	1	0	1	2	5	5	5	5	5	2
0	10	5	0	5	0	1	5	2	1	5	5	5	5	2
0	11	5	5	5	5	5	5	5	5	1	5	5	5	2
0	12	5	5	5	5	5	5	5	5	5	3	5	5	2
2	13	5	5	5	5	5	5	5	5	3	1	5	5	3
2	14	5	5	5	5	5	5	5	5	5	3	5	5	1

Columna (C) indica el valor del estado de la célula central.
Columna (#) indica el número de regla.
Columna 3-10, indican el valor del estado en una posición específica del vecindario.
Columna 11-14, indican la cantidad de estados que se repiten en el vecindario.
Columna (A), indica el valor del estado que obtendrá la célula central cuando la regla sea verdadera.

Tabla XIX: Experimento 99%. Datos generados por el algoritmo de infección

CICLO	COLUMNA-1	COLUMNA-2	COLUMNA-3	COLUMNA-4
0	353526	0	0	0
1	353471	2	1854	0
2	351615	447	4916	1402
5	340457	807	19672	12176
10	310964	1252	46619	41151
20	241621	1612	115120	110058
30	150014	1770	205343	201484
50	115794	1870	239190	235573
100	52416	2002	302086	298799
150	13033	2057	340955	338126
200	3223	2081	350324	347912
250	148	2081	353349	350987
300	0	2081	353461	351132

Hora Inicial:Mon Mar 15 23:55:08 2004
Hora Final:Mon Mar 15 23:57:24 2004

Tiempo transcurrido: **00:02:16** Contra: 00:17:18 del método Exhaustivo

Porcentaje de Ahorro en operaciones: $(351132 \times 100) \div 353626 = 99.2\%$
Porcentaje de Ahorro en tiempo: 87%

COLUMNA-1:NO-EXPLOTADOS (*Sanos*)
COLUMNA-2:EXPLOTADOS (*Enfermos*)
COLUMNA-3:PROPUESTOS (*Infectados*)
COLUMNA-4:AUTOMÁTICOS (*Inmunes*)

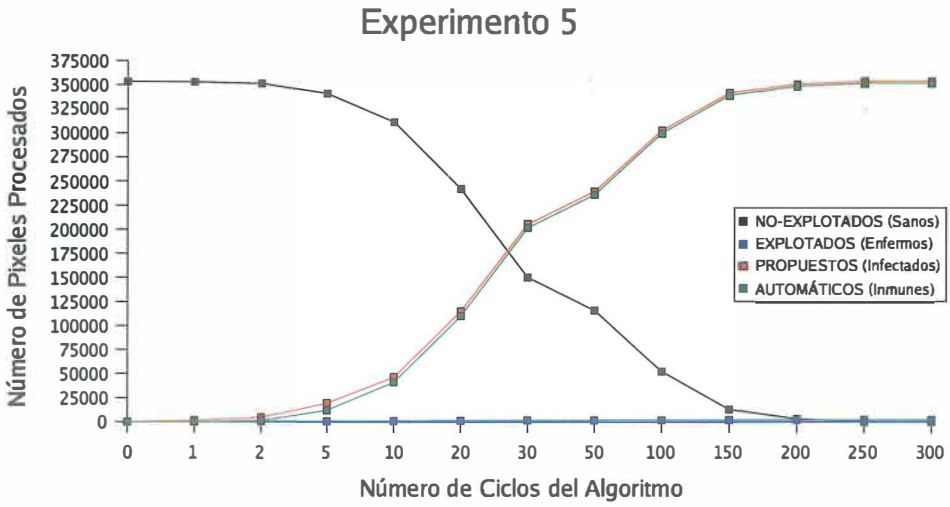


Figura 68: Experimento 99%. Gráfica de resultados de la evolución de cada uno de los estados.

un 87% en tiempo en comparación al método exhaustivo, y sólo evalúa un 1% de los pixeles contenidos en la imagen.

La evolución se puede apreciar visualmente en la gráfica de la Figura 68, la curva de los pixeles AUTOMÁTICOS (*Inmunes*) se encuentra muy por encima de los pixeles EXPLOTADOS (*Enfermos*). La curva que representa los pixeles NO-EXPLOTADOS (*Sanos*) queda invertida en comparación con la curva de los pixeles AUTOMÁTICOS (*Inmunes*) ya que cada vez que la curva de los pixeles NO-EXPLOTADOS (*Sanos*) desciende, la curva de los AUTOMÁTICOS (*Inmunes*) sube casi en la misma proporción.

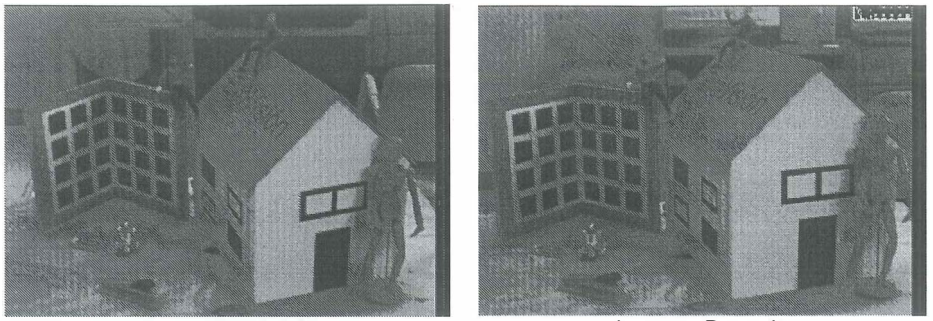


Imagen Izquierda Imagen Derecha

Figura 69: Imágenes utilizadas como datos de entrada para el algoritmo de infección en este experimento.

V.3.6 Experimento 6

Este experimento consistió en utilizar un par de estéreo distinto al utilizado en los experimentos anteriores. El par estéreo se obtuvo realizando movimientos de la cámara mayores que los anteriores (rotación y traslación), por lo tanto, la diferencia entre las imágenes es mayor. Lo anterior se hizo con el fin de probar el algoritmo de infección en un contexto más difícil que el anterior.

La Figura 69 muestra las imágenes de entrada. La diferencia de rotación y traslación entre las imágenes en cada uno de los ejes es el siguiente:

Rotación:	**Traslación:**
$R_x = 3.04°$	$t_x = 164.96mm$
$R_y = 1.98°$	$t_x = 164.96mm$
$R_z = 2.69°$	$t_z = 203.94mm$

donde, R_x, R_y y R_z representan la rotación en x,y y z respectivamente. t_x, t_y y t_z representan las traslaciones en x, y y z respectivamente.

El resultado de este experimento con el conjunto de reglas que permiten el ahorro de un 47% en operaciones se muestra en la Figura 70 y la Figura 71, muestra el resultado del

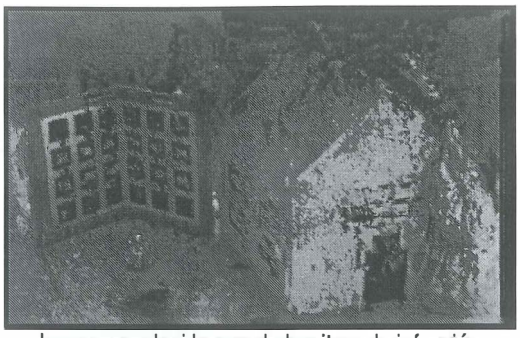


Figura 70: Resultado del algoritmo de infección utilizando las reglas del 47% para este experimento.

Figura 71: Resultado del algoritmo de infección utilizando las reglas del 99% para este experimento.

algoritmo de infección con un ahorro del 99% en operaciones. La calidad del resultado no es muy buena debido al movimiento que tuvo la cámara cuando se adquirieron las fotografías. Sin embargo, el experimento nos permitió confirmar que el problema es más difícil y en un futuro, adaptar el algoritmo para movimientos mayores a los de un par estereoscópico, aunque el algoritmo haya sido ideado para este tipo de imágenes donde los movimientos de las cámaras no son muy grandes.

V.4 Conclusiones

Se llevaron a cabo simulaciones que proporcionaron confiabilidad en el conjunto de reglas para iniciar la implementación del algoritmo de infección con las imágenes reales. Los resultados mostraron un buen desempeño del algoritmo ya que a pesar del ahorro de operaciones la calidad de la imagen no se pierde en relación a una imagen producida por el algoritmo de búsqueda exhaustiva; al contrario, se mejora ligeramente tal como se muestra en las distintas experimentaciones de la sección V.3.

Una vez que se diseñó un conjunto de catorce reglas que permitieran ahorrar un 47% en cálculos, se analizó la posibilidad de cambiar la tabla de reglas para obtener distintos niveles de ahorro sobre el mismo par de imágenes usadas como datos de entrada.

Nos dimos cuenta también que cualquier cambio en el conjunto de reglas se refleja de manera muy sensitiva en el comportamiento de la propagación. Los cambios pueden hacerse fácilmente variando las cantidades numéricas y los estados dentro de la tabla.

La evolución del algoritmo con las imágenes reales muestra cómo se va generando la propagación de la correspondencia alrededor de los núcleos de infección, lo cual es similar a la evolución de la simulación de las reglas en la sección V.2. El tiempo del algoritmo varía de acuerdo al número de pixeles AUTOMÁTICOS (*Inmunes*) que se generen en el proceso de correspondencia. El tiempor del algoritmo de búsqueda exhaustiva no varía porque siempre recorre pixel a pixel toda la imagen, evaluando y recalculando todos los datos para encontrar el pixel correspondiente en la imagen derecha.

Capíulo VI

Conclusiones y Trabajo Futuro

Se propuso un nuevo enfoque bioinspirado para la solución del problema de correspondencia denso, el cual denominamos algoritmo de infección. Uno de los objetivos principales de la investigación consistió en implementar un algoritmo de correspondencias que permitiera ahorrar tiempo en cálculos cuidando la calidad de la imagen. El par estéreo cuenta con los problemas principales de la correspondencia como la oclusión, baja textura, campo de visión, entre otros.

El algoritmo tiene características dinámicas las cuales permiten cambiar el funcionamiento del autómata a fin de obtener distintos niveles de ahorro en cálculos. Lo anterior, es debido a que el conjunto de reglas es flexible para modificar, agregar o borrar reglas y así obtener los diferentes porcentajes de ahorro.

La experimentación real confirmó la hipótesis de que al menos se podría ahorrar un 47% en cálculos. Primero se realizó la simulación con la meta de alcanzar un nivel de ahorro alrededor del 50% en cálculos. La simulación consistió en obtener el conjunto de reglas que se aplicarían en la implementación con el par estéreo.

Una vez que se implementó el algoritmo, se realizaron distintas pruebas modificando ciertos parámetros del archivo de reglas para obtener diferentes porcentajes de ahorro y de esta forma visualizar los resultados. Las modificaciones fueron hechas para obtener un 0%, 70%, 78% y 99%. Cada uno de los casos anteriores mostraron buenos resultados, sobre todo el caso del 99% ya que ciertas regiones de la imagen son muy nítidas a pesar que los cálculos realizados son realmente mínimos. Sin embargo, existen ciertas regiones

donde la correspondencia no es buena.

Los experimentos mostraron que para una disparidad pequeña entre el par de imágenes, es posible realizar un número amplio de asignaciones automáticas sin perder la calidad de la imagen. En cambio, para una disparidad grande la calidad de la imagen se pierde, tal como se mostró en el último experimento.

El algoritmo muestra buenos resultados independientemente de la modificación del conjunto de reglas para los distintos casos; evita realizar cálculos para determinar el pixel correspondiente en la imagen derecha creando consigo un efecto *"dominó"* en la propagación al asignar automáticamente la correspondencia. El ahorro de un alto número de cálculos permite disminuir el tiempo de ejecusión del algoritmo.

Para utilizar el algoritmo de infección cada imagen debe contener una mira de calibración porque el sistema es calibrado. La mira de calibración es un objeto puesto en la escena con el cual se obtienen datos del mundo para llegar a conocer a través de la calibración, los parámetros intrínsecos y extrínsecos. Los datos para la correspondencia dependen de la veracidad de la calibración debido a que se utiliza la geometría epipolar en la correspondencia.

Utilizar la información calibrada nos permite obtener una reconstrucción tridimensional precisa del objeto y conservar las medidas que tiene en el mundo real.

El algoritmo de infección tomará las bases para la reconstrucción tridimensional de un objeto completo utilizando un mínimo número de fotografías como entrada de datos. Para ello, se propone lo siguiente como trabajo futuro:

- Localizar la ubicación precisa de las oclusiones en el par estereoscópico. Esto permitirá conocer dónde se encuentra la información que hace falta para considerla

en la siguiente fotografía. La ubicación se refiere en dónde empieza y termina la oclusión en el par de imágenes. Se necesitaría analizar los distintos algoritmos existentes que tratan el problema de la oclusión.

- Mejorar el tiempo del proceso de correspondencia esperando que sea en tiempo real. Si el algoritmo mejora el tiempo a segundos puede ser utilizado en la robótica móvil, así como también en otras áreas.

- Ampliar las características de nuestro algoritmo a fin de planear una secuencia de vistas que permitan lograr la reconstrucción de la escena en 3D. Esto es debido a que el algoritmo de infección hace la correspondencia para dos imágenes.

- Es conveniente desarrollar el concepto del algoritmo de infección utilizando un sistema no calibrado. Esta opción permitirá que el algoritmo sea usado en otro contexto como en las aplicaciones al aire libre y que no dependa de los datos de la calibración para su funcionamiento.

Como se mencionó anteriormente, este trabajo de investigación prentende fijar las bases para un proyecto a futuro. Este proyecto está relacionado a otro problema abierto dentro de visión por computadora llamado en inglés *Next Best View Problem*. Este problema consiste en reconstruir un objeto completo con el mínimo número de vistas, lo cual indica que será necesario conocer el posicionamiento que deberá tener la cámara para tomar cada una de las fotografías. El problema de correspondencia continúa siendo un problema abierto dentro de visión desde hace más de dos décadas. Este trabajo de investigación aunque no dió la solución general al problema, permitió analizarlo detalladamente y proponer un algoritmo inspirado en la naturaleza aplicado hasta el momento a la correspondencia densa estéreo[1].

[1]En inglés, Dense Stereo Matching

Bibliografía

Abbey, H. 1952. "An examination of the reed frost theory of epidemics". Human Biology, 24:201-233 p.

Ballard, D. y C. Brown 1982. "Computer vision". Prentice Hall, Englewood Cliffs, New Jersey. 523 pp.

Barnard, S. y M. Fischler 1982. "Computational stereo". ACM, Computer Surveys, 14(4):553-572 p.

Birchfield, S. y C. Tomasi 1999. "Multiway cut for stereo and motion with slanted surfaces". En: sponsored by the IEEE Computer Society, editor, "International Conference on Computer Vision ICCV99", volume 1. 489-495 p.

Brown, M., D. Burschka, y G. Hager 2003. "Advances in computational stereo". IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(8):993-1008 p.

Chaudhuri, Chowdhury, Nandi, y Chattopadhyay 1997. "Additive cellular automata, theory and applications". IEEE Computer Society Press, second edition. 364 pp.

Cox, Hingorani, Rao, y Maggs 1996. "A maximum likelihood stereo algorithm". Computer Vision and Image Understanding, 63:542-567 p.

Dhond, U. y K. Aggarwal 1989. "Structure from stereo". IEEE Transactions on Systems and Man and Cybernetics, 19(6):1489-1509 p.

Faugeras, O. 1996. "Three-dimensional computer vision: a geometric viewpoint". MIT Press, Massachutsetts Institute of Technology, second edition. 695 pp.

Faugeras, O. y G. Toscani 1986. "The calibration problem for stereo". En: Press, I. C. S., editor, "Proceedings of IEEE Computer Vision and Pattern Recognition". 15-20 p.

Forsyth, D. y J. Ponce 2002. "Computer vision: A modern approach". Prentice Hall, first edition. 693 pp.

Fusiello, A., V. Roberto, y E. Trucco 1997. "Experiments with a new area-based stereo algorithm". En: Bimbo, A. D., editor, "Proc. of the 9th International Conference on Image Analysis and Processing, ICIAP '97,". Florence, Italy. 669-676 p.

Gonzalez, R. y R. Woods 2002. "Digital image processing". Prentice Hall, Inc., Saddle River, New Jersey, second edition. 793 pp.

Gruen, A. 1985. "Adaptive least squares correlation: A powerful image matching technique". S.Afr.J of Photogrammetry, Remote Sensing and Cartography, 14(3):175-187 p.

Hartley, R. 1997. "In defense of the eight-point algorithm". IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), 19(6):580-593 p.

Hartley, R. y A. Zisserman 2001. "Multiple view geometry in computer vision". Cambridge University Press, United Kingdom, first edition. 607 pp.

Heath, M., S. Sarkar, T. Sanocki, y K. Bowyer 1998. "Comparison of edge detectors: A methodology and initial study". Computer Vision and Image Understanding, 69(1):28-54 p.

Horaud, R. y O. Monga 1995. "Vision par ordinateur". Hermes, 14 rue Latinez 75017, Paris, France, second edition. 425 pp.

Kitchen, L. y A. Rosenfeld 1982. "Gray level corner detection". Pattern Recognition Letters, 1:95-102 p.

Lhuillier, M. y L. Quan 2002. "Quasi-dense reconstruction from image sequence". En: Heyden, A., G. Sparr, M. Nielsen, y P. Johansen, editores, "7th European Conference on Computer Vision, ECCV'2002". Copenhagen, Denmark. 125-139 p.

Maniatty, W., B. Szymanski, y T. Caraco 1998. "Parallel computing with generalized cellular automata". Parallel and Distributed Programming Practices, 1(1):85-104 p.

Mitchell, M. y S. Forrest 1995. "Genetic algorithms and artificial life". Artificial Life, 1(3):267-289 p.

Mühlmann, K., D. Maier, J. Hesser, y R. Manner 2001. "Calculating dense disparity maps from color stereo images, an efficient implementation". En: sponsored by IEEE Computer Society Technical Committee on Pattern Analysis, M. Intelligence, I. Corporation, y P. G. Research, editores, "IEEE Workshop Stereo and Multi-Baseline Vision". Kkauai, Hawaii. 30-36 p.

Ohta y Kanade 1985. "Stereo by intra-and intra-scanline search using dynamic programming". IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), 7:139-154 p.

Olague, G. y B. Hernández 2004. "A new accurate and flexible model based multi-corner detector for measurement and recognition". Aceptado para su publicación en Pattern Recognition Letters.

Press, W., S. Teukolsky, W. Vetterling, y B. Flannery 1994. "Numerical recipes in c". Cambridge University Press, 40 West 20th Street, New York, second edition. 994 pp.

Ramirez, R. 2000. "Síntesis de imágenes a partir de fotografías". Tesis de Maestría, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Baja California. 218 pp.

Roy, S. y I. Cox 1998. "A maximum-flow formulation of the n-camera stereo correspondence problem". En: sponsored by the IEEE Computer Society, editor, "IEEE Proc. of Int. Conference on Computer Vision". Bombay, India. 492-499 p.

Sipper, M. 1997. "Evolution of parallel cellular machines". Lecture Notes in Computer Science, Springer-Verlag, 1194:198 pp.

Steven, L. 1993. "Artificial life: A report from the frontier where computers meet biology". Vintage Books USA, reprint edition edition. 393 pp.

Sun, C. 2002. "Fast stereo matching using rectangular subregioning and 3d,maximum-surface techniques". International Journal of Computer Vision, 47(1/2/3):99-117 p.

Tomasi, C. y R. Manduchi 1998. "Stereo matching as a nearest-neighbor problem". IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), 20(3):333-340 p.

Zitnick, L. y T. Kanade 2000. "A cooperative algorithm for stereo matching and occlusion detection". IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), 22(7):675-684 p.

Bibliografía del Autor

Contribuciones sobre este trabajo de investigación:

- Congresos internacionales con estricto arbitraje.

  Olague, G., Fernández, F., Pérez C., and Lutton, E. (2004)

  "The Infection Algorithm: An Artificial Epidemic Approach for Dense Stereo Matching"

  Parallel Problem Solving from Nature (PPSN)., Lecture Notes in Computer Science.

  Birmingham, U.K., September 18-22, 2004.

- Congresos nacionales.

  Pérez, C., Fernández, F., Lutton, E., and Olague, G.

  "Dense Stereo Matching for 3D Reconstruction using Artificial Life."

  IV Simposio La Óptica en la Industria.

  9 y 10 de Septiembre de 2004, Ensenada B.C.

Apédice A

Especificaciones del funcionamiento del sistema

Este apéndice contiene las especificaciones del funcionamiento del sistema de reconstrucción. Tiene como objetivo dar a conocer el formato de los datos de entrada, datos de salida y las dependencias de cada uno de los módulos y las funciones dentro de cada módulo.

A.1 Módulo para la Adquisición de Fotos

Este módulo usa el detector de esquinas KyR Kitchen y Rosenfeld (1982) y genera las imágenes izquierda y derecha con las esquinas detectadas.

Datos de Entrada: fotoizq00001.tif y fotoder00001.tif.
Datos de Salida: fotoizqKR.tif y fotoderKR.tif.
Función del Código: xcv_reconst::tomarFotoIZQ() y xcv_reconst::tomarFotoDER().

A.2 Módulo para la Calibración de la cámara izquierda

Este módulo realiza la calibración de la cámara izquierda y genera la matriz M izquierda. La calibración puede hacerse con 12 o 64 puntos de la mira de calibración.

El submenú del módulo es:

Captura de 12 coordenadas de la mira en la imagen izquierda.
Datos de Entrada: uv12izq.dat y xyz12.dat

Datos de Salida: uvxyzizq.dat

Función del Código: xcv_reconst::captura12CoordIZQ()

Captura de 64 coordenadas de la mira en la imagen izquierda.

Datos de Entrada: uv64izq.dat y xyz64.dat

Datos de Salida: uvxyzizq.dat

Función del Código: xcv_reconst::captura64CoordIZQ()

Generar la matriz de proyección M izquierda.

Datos de Entrada: uvxyzizq.dat

Datos de Salida: parametrosIzq.dat

Función del Código:xcv_reconst::generaMatrizM_IZQ()

A.3 Módulo para la Calibración de la cámara derecha

Este módulo realiza la calibración de la cámara derecha y genera la matriz M derecha. La calibración puede hacerse con 12 ó 64 puntos de la mira de calibración.

El submenú del módulo es:*Captura de 12 coordenadas de la mira en la imagen derecha.*

Datos de Entrada: uv12der.dat y xyz12.dat

Datos de Salida: uvxyzder.dat

Función del Código: xcv_reconst::captura12CoordDER()

Captura de 64 coordenadas de la mira en la imagen derecha.

Datos de Entrada: uv64der.dat y xyz64.dat

Datos de Salida: uvxyzder.dat

Función del Código: xcv_reconst::captura64CoordDER()

Generar la matriz de proyección M derecha.

Datos de Entrada: uvxyzder.dat

Datos de Salida: parametrosDer.dat

Función del Código:xcv_reconst::generaMatrizM_DER()

A.4 Módulo para la Generación de puntos 3D

Este módulo genera los puntos 3D una vez que se lleva a cabo la correspondencia y la calibración.

Datos de Entrada:reglas.dat, dimensiones.dat, parametrosIzq.dat, parametrosDer.dat

Datos de Salida: datos.txt, Puntos3D.dat, MatrizAs.dat, MatrizEs.dat, FotoFinal.tif

Función del Código: xcv_reconst::algoritmoInfeccion()

A.5 Uso del Sistema

El sistema puede generar la imagen intermedia calibrando ambas cámaras, o bien, si la calibración fue realizada por lo menos una vez. La imagen intermedia representa la correspondencia visual entre la imagen izquierda y la imagen derecha.

Los pasos para generar la correspondencia entre dos fotografías tomadas desde distintos puntos de vista, son los siguientes:

Si no se ha realizado la calibración:

1. En el menú principal seleccione la opción *3D_Reconstruction*, luego elija *Acquire LEFT Photo* para generar la imagen con las esquinas detectadas (fotoizqKR.tif).
2. En el menú principal seleccione la opción *File*, después elija la opción *Load Image* para cargar la imagen. Seleccione la imagen fotoizqKR.tif que se encuentra en ./xcv/ Una vez hecho esto, seleccione la opción *LEFT Camera Calibration* del menu *3D_Reconstruction* y seleccione la opción 12 *Coord.MIRA* o 64 *Coord.MIRA* según sea el caso. Los 12 o 64 puntos pueden seleccionarse mas fácil acercando la imagen en las esquinas correspondientes. Con la tecla CTRL+Boton-Izq-Mouse se puede hacer Zoom++ y con CTRL+Boton-Der-Mouse un Zoom–.
3. Seleccione del menú principal la opción *3D_Reconstruction*, después elija la opción *LEFT Camera Calibration* y dé un click en *Generate Matrix M LEFT*.
4. Repetir el paso 2 hasta el 4 pero ahora con la imagen Derecha.
5. Seleccione del menú principal la opcion *3D_Reconstruction*, posteriormente elija la opción *3D Points Generation* y de un click en *Stereo Matching* para llevar a cabo la corresondencia con el algoritmo de infección.

Si la calibración fue realizada anteriormente:

1. Si se desea llevar a cabo la correspondencia con las mismas imágenes no es necesario realizar todos los pasos anteriores si la calibración se realizó mínimamente en una ocasión con esas imágenes. En este caso, solamente es necesario llevar a cabo el paso 5.

Apédice B

Visualización de las reglas

Este apéndice tiene como objetivo mostrar las imágenes y los grafos que representan visualmente el funcionamiento de las reglas en los demás casos 0%, 78% y 99% de ahorro en operaciones. El caso del 47% fue descrito detalladamente en el capítulo IV como ejemplo del funcionamiento del algoritmo de infección. El caso del 70% de ahorro no se muestra en el apéndice dado que los estados iniciales y finales son los mismos que el conjunto de reglas que ahorran un 47% mostrados en la tabla de reglas, solo varía la información interna de la tabla de reglas. La diferencia radica en la información que se necesita para aplicar cada regla, tal como la cantidad de vecinos EXPLOTADOS (*Enfermos*). Comparar la Tabla X de la Sección V.3.1 y la Tabla XIV de la Sección V.3.3.

En todos los casos como se mencionó en el capítulo IV las primeras 6 reglas no cambian. Las dos primeras se usan para no realizar cambios en el autómata cuando el pixel central se encuentre en estado NO-EXPLOTADO (*Sano*) o EXPLOTADO (*Enfermo*). Las siguientes 4 reglas fueron pensadas para la estructura inicial alrededor del núcleo de infección.

Las reglas pueden ser dibujadas con la ayuda de las tablas de datos donde fueron codificadas numéricamente.

B.1 Caso del 0%

En este caso, la única regla que fue modificada es la regla 11 (ver Figura 72). El grafo que representa las transiciones de las reglas para evaluar cada pixel de la imagen es el grafo que se muestra en la imagen 73.

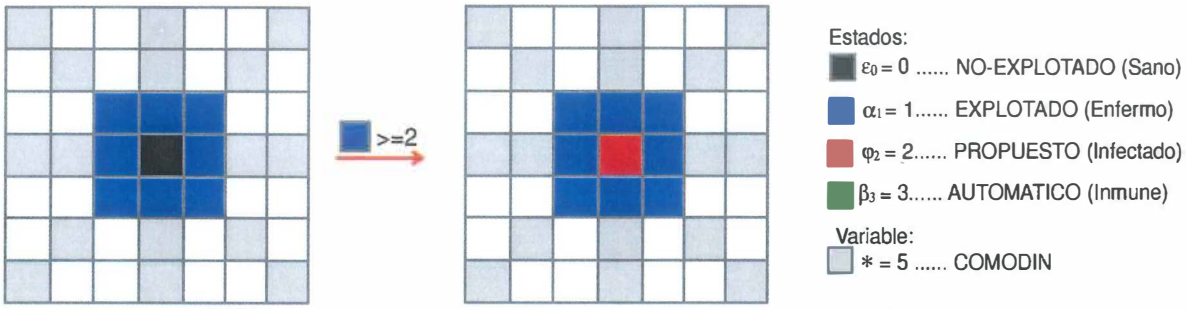


Figura 72: Regla 11 para el caso del 0% de ahorro en operaciones.

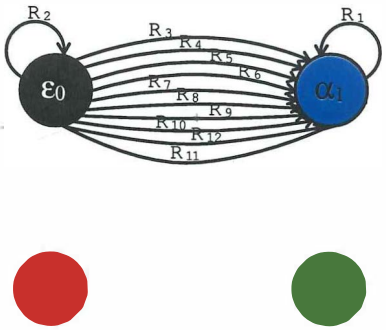


Figura 73: Grafo que representa las transiciones de todas las reglas para el caso del 0% de ahorro.

B.2 Caso del 78%

Para ahorrar un 78% de operaciones se modifican las reglas 7,8,9,10,11 y 13 tal como se muestra a continuación en las imágenes.

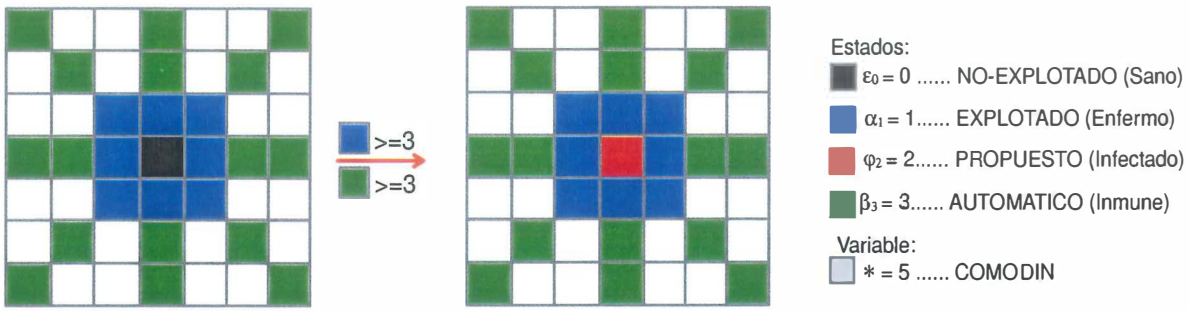


Figura 74: Regla 7 para 78%. El cambio surgió en el estado final del pixel.

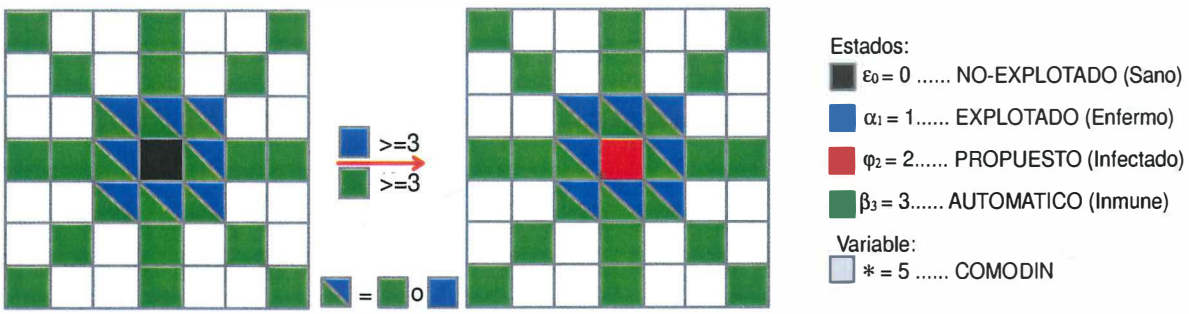


Figura 75: Regla 8 para 78%. Se cambió el estado final del pixel a un estado PROPUESTO (*Infectado*).

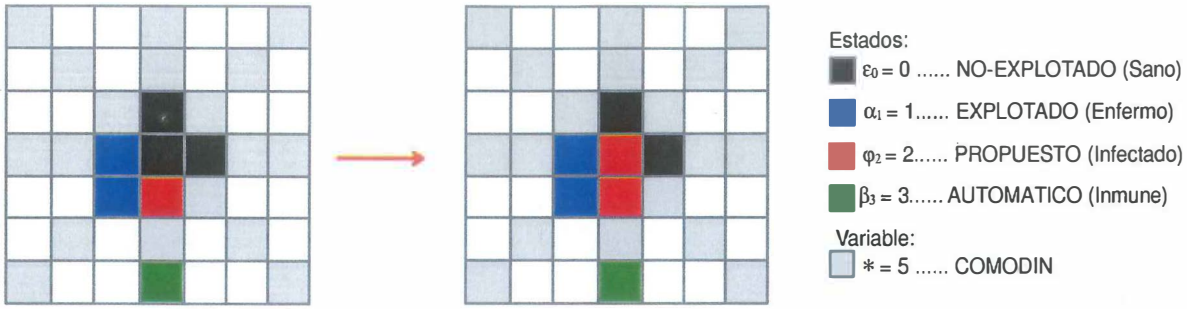


Figura 76: Regla 9 para 78%. De igual forma, el cambio fué en el estado final del pixel.

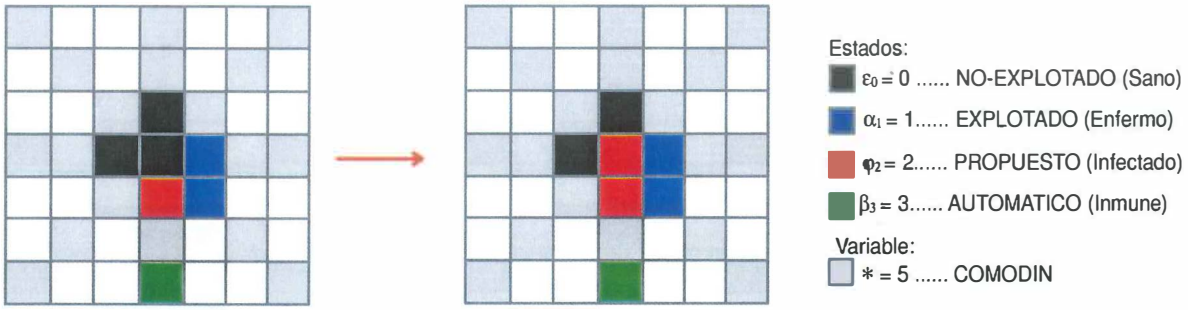


Figura 77: Regla 10 para 78%. Se cambió el estado final del pixel a un estado PROPUESTO (*Infectado*).

El grafo que representa al conjunto de reglas para el caso de 78%, es el que se muestra en la Figura 80.

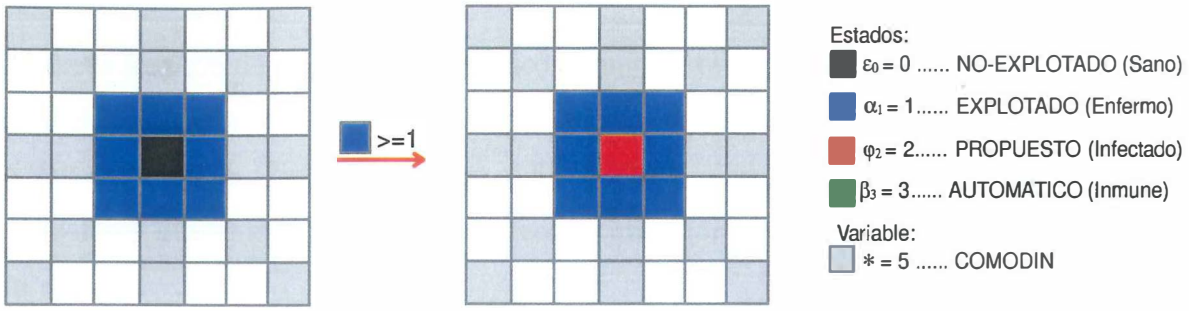


Figura 78: Regla 11 para 78%. Se disminuyó el número de vecinos EXPLOTADOS (*Enfermos*).

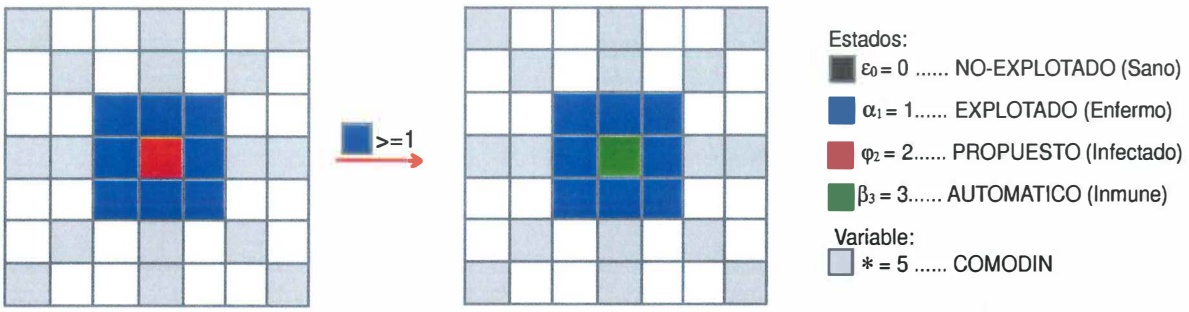


Figura 79: Regla 13 para 78%. Se disminuyó el número de vecinos PROPUESTOS (*Infectados*).

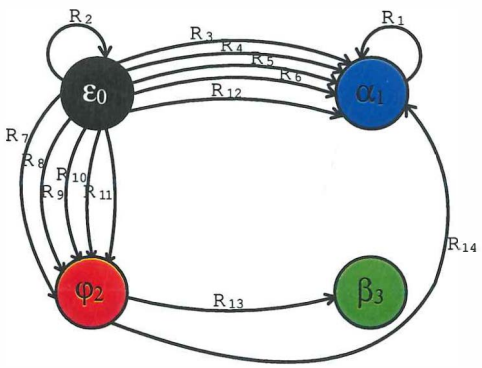


Figura 80: Grafo que representa las transiciones de cada regla cuando se ahorra un 78% de cálculos.

B.3 Caso del 99%

Ahorrar un 99% de operaciones significó aumentar el número de estados finales a PROPUESTOS (*Infectados*) y disminuir la cantidad de vecinos EXPLOTADOS (*Enfermos*) e PROPUESTOS (*Infectados*). Con el fin, que se realizaran un mínimo de evaluaciones. Si se desea que se evalúen solo los núcleos de infección, entonces, se tendría que eliminar la regla 14.

Las reglas que fueron modificadas fueron 7,8,9,10,11,12 y 13. Se muestran en la figura 81 , 82, 83 , 84 , 85 , 86 y 87.

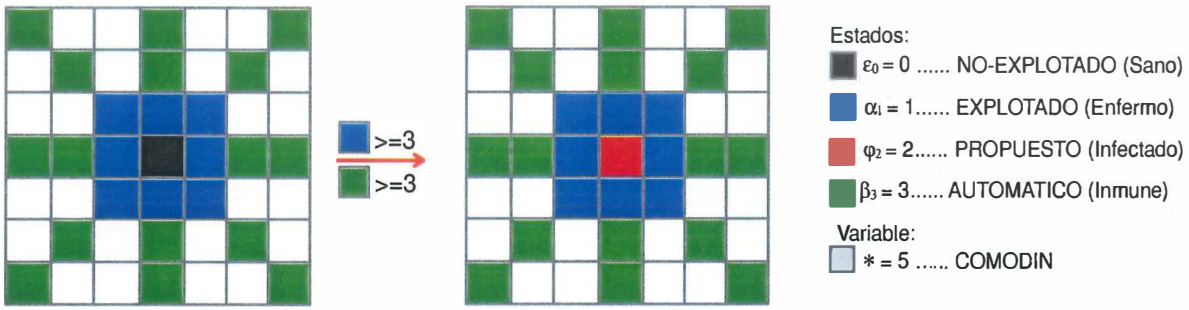


Figura 81: Regla 7 para 99%. El cambio realizado en la regla 7 fue en el estado final de la transición.

El grafo que representa las transiciones del conjunto de reglas usadas para ahorrar un 99% de operaciones, se muestra en la Figura 88.

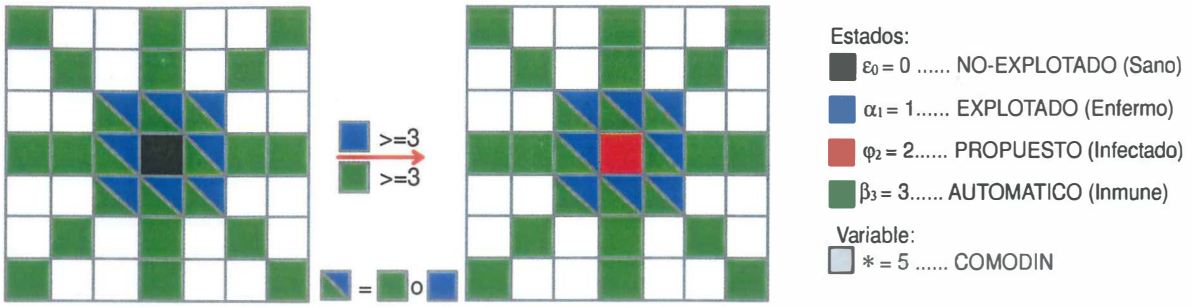


Figura 82: Regla 8 para 99%. El estado final cambió al estado PROPUESTO (*Infectado*).

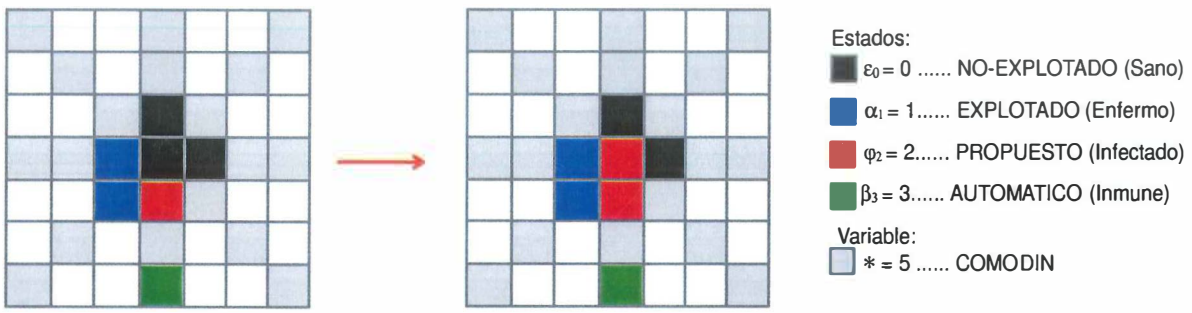


Figura 83: Regla 9 para 99%. Se cambió el estado final a estado PROPUESTO (*Infectado*).

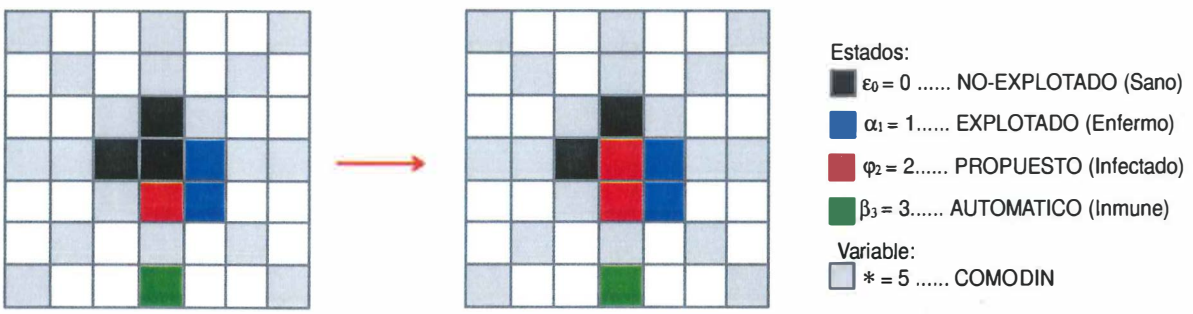


Figura 84: Regla 10 para 99%. Esta regla cambia al igual que las anteriores en el estado final del pixel.

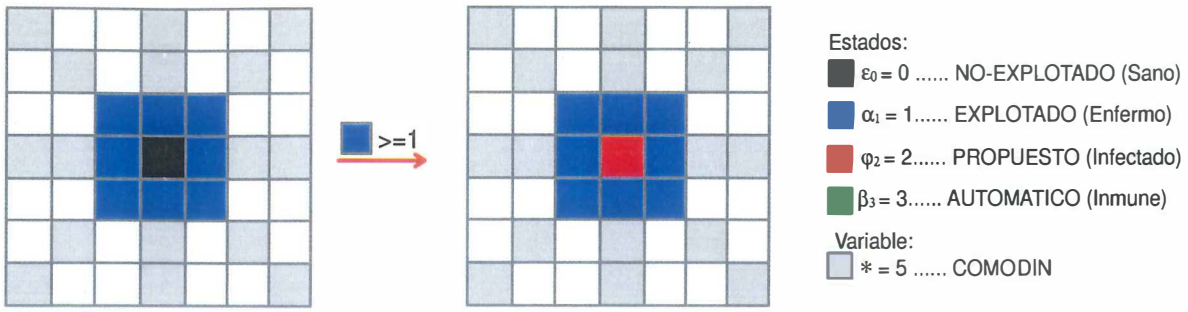


Figura 85: Regla 11 para 99%. Se disminuyó el número de vecinos EXPLOTADOS (*Enfermos*).

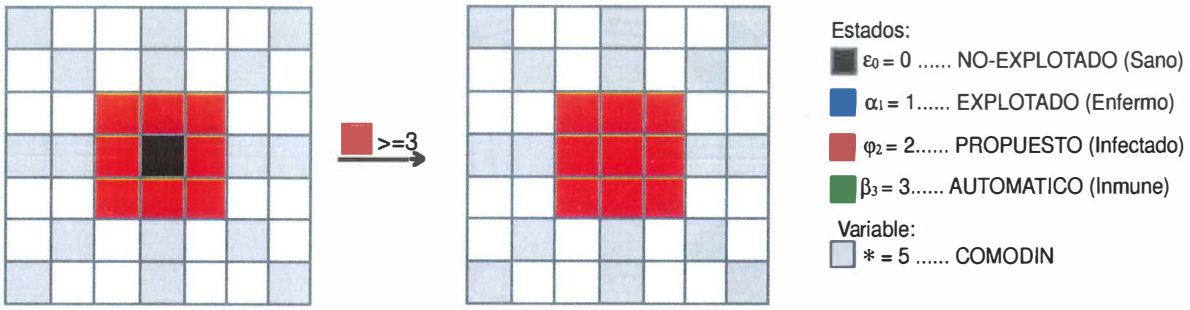


Figura 86: Regla 12 para 99%. Se cambió el estado final del pixel central.

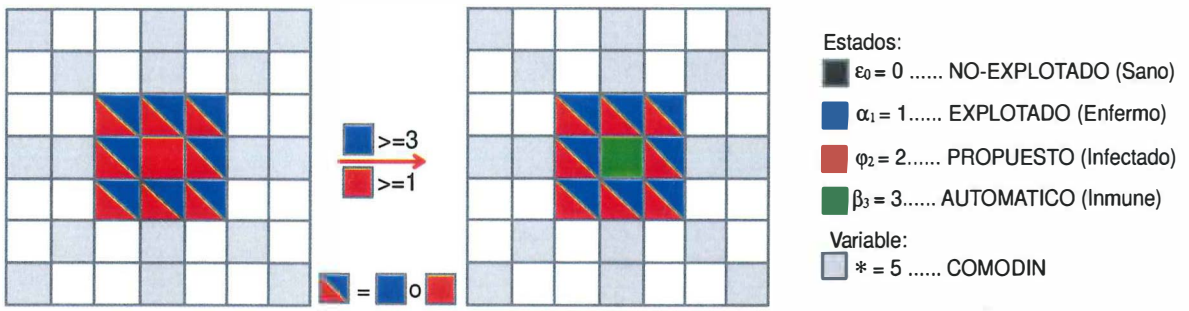


Figura 87: Regla 13 para 99%. Se disminuyó el número de vecinos PROPUESTOS (*Infectados*).

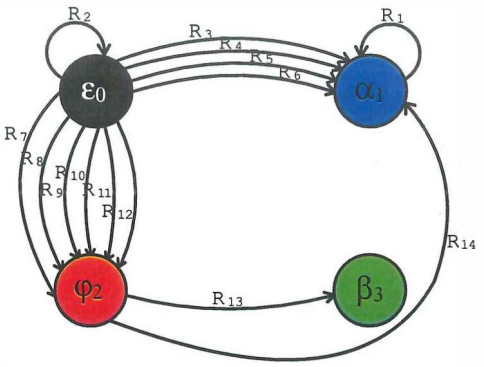


Figura 88: Grafo que muestra las transiciones de cada una de las reglas para el caso de 99% de ahorro.