POSGRADO INTERINSTITUCIONAL EN CIENCIA Y TECNOLOGÍA

CENTRO DE INGENIERÍA Y DESARROLLO INDUSTRIAL

Generación de gramáticas con alta expresividad para modelar la dinámica de objetos en un sistema de visión

TESIS

QUE PARA OBTENER EL GRADO ACADÉMICO DE:

MAESTRO EN CIENCIA Y TECNOLOGÍA CON ESPECIALIDAD EN MECATRÓNICA

PRESENTA:

Herlindo Hernández Ramírez.

ASESORES

Dr. Hugo Jiménez Hernández.

M. C. y T. Juan Manuel García Huerta.

SANTIAGO DE QUERÉTARO, QUERÉTARO,
16 de agosto de 2016

Resumen

Los sistemas de monitoreo basados en visión han generado un nicho importante, pues permiten el monitoreo semiautomático en escenarios concretos. Sin embargo, la detección de eventos depende del conocimiento previo del escenario. Algunos autores han enfocado sus investigaciones en desarrollar métodos para encontrar un modelo que mejor describa la dinámica de los objetos, de estas investigaciones surgen los modelos que usan algoritmos de inferencia Gramatical para representar la dinámica de objetos en un escenario, como un conjunto de reglas de producción, donde se hace una representación simbólica de las trayectorias generadas y el criterio del modelo está basado en la aceptación o rechazo de una cadena de símbolos por el modelo

La base de este trabajo basado en el modelo descrito anteriormente, se enfoca en la manera en que se generan las reglas de producción, especialmente en el numero de reglas que son generadas por el algoritmo de inferencia gramatical [1], para describir una trayectoria. Como consecuencia se presenta un método para representar la dinámica de los objetos como un modelo Gramatical con reglas con alta expresividad que las que se obtienen al aplicar el algoritmo de inferencia gramatical SEQUITUR. Para esto es necesario tener la representación de las trayectorias como cadenas de símbolos, siguiendo la metodología propuesta por [1] se obtienen estas cadenas las cuales se usaran para crear el modelo gramatical con un algoritmo con mayor generalización en el momento de generar las reglas de producción, el criterio que se usará para determinar la expresividad del algoritmo es comparar el numero de reglas necesarias para describir la dinámica de una trayectoria con dos conjuntos de reglas de producción cada uno perteneciente a los algoritmos de inferencia

gramatical que se van a estudiar.

Dedicatoria

Esta tesis se la dedico a:

Mis hermanos y en especial a mis padres que han sido pilar fundamental en mi formación personal.

Agradecimientos

La culminación de esta tesis es un trabajo conjunto multidisciplinario, entre personas que saben lo que hacen y que sin su conocimiento y apoyo no hubiera sido posible lo anterior. Agradezco a mis asesores Dr. Hugo Jiménez y M.C y T. Manual García que han sabido ser pacientes y se han ganado el respeto y admiración de mi persona, entre muchas cosas por su capacidad intelectual, paciencia y disponibilidad a atender inquietudes que se fueron sucitando a lo largo de este proceso que fue el desarrollo y culminación del trabajo arriba descrito.

A CIDESI, por su espacio y material prestado para el desarrollo de esta tesis.

A Conacyt, quien me ofreció su beca de maestría para mi manutención a lo largo del proyecto de investigación.

Índice general

Índice de figuras

Capítulo 1

Introducción.

En visión por computadora, hay toda una variedad de líneas de investigación y áreas de aplicación. Entre ellas se encuentran los sistemas que modelan la dinámica de un escenario basado en trayectorias. El análisis basado en la dinámica de trayectorias busca proveer información para la caracterización y entendimiento del comportamiento de cada objeto en la escena. Para este fin, cada objeto es identificado y seguido para poder describir la actividad que realiza. En estos casos el modelo de la escena está definido por dos etapas de aprendizaje. La primera es definir puntos de interés, los cuales son regiones de la imagen donde ocurren eventos de interés. La segunda etapa consiste en definir las rutas de movimiento, las cuales caracterizan como los objetos viajan entre los puntos de interés. El aprendizaje de los puntos anteriores crea un modelo para analizar la escena, este lenguaje permite el análisis de actividades actuales y pasadas, detección de actividades anormales, predicción de futuras actividades y caracterización de interacciones entre los objetos, lo cual es de suma importancia en sistemas de video vigilancia [3]. Sin embargo en las investigaciones recientes aunque estos modelos funcionan y existen aquellos que también se basan en gramáticas para definir la dinámica de un escenario, las zonas de interés se definen manualmente [4]. Además las palabras necesarias para aprender son generadas a partir del resultado de la concatenación de las zonas de interés con las que el objeto ha interactuado, para después inferir una gramática libre de contexto [14]. Lo que conlleva a lo que parece una excesiva discretización del espacio, lo que resulta en un mayor tiempo de espera para saber que ruta se está siguiendo.

1.1. Objetivo general

Encontrar un método de inferencia gramatical, que genere reglas para describir patrones de movimiento, asociados con cadenas de símbolos, que se generan con las trayectorias seguidas por objetos en una ruta (Generar reglas que describan la dinámica de objetos para una o más cadenas de símbolos generadas en la misma ruta de desplazamiento).

1.2. Objetivos particulares

- Definir el modelo gramatical.
- Implementar algoritmo.
- Probar la hipótesis.

1.3. Hipótesis

Si la dinámica del movimiento de un objeto en un escenario genera un conjunto de trayectorias sobre una ruta, y estas pueden representarse como un conjunto de gramáticas que derivan un conjunto de reglas de producción [1], entonces, existe un modelo gramatical, que genera un conjunto de gramáticas mas expresivas, que derivan reglas de producción, que describen los mismos patrones de movimiento o más. .

1.4. Justificación

El modelado de la dinámica del movimiento de objetos puede ser representado como un conjunto de gramáticas que lo describen, a su vez, estas gramáticas generan un conjunto de reglas de producción; sin embargo las gramáticas generadas por un conjunto de trayectorias que pertenecen a la misma ruta no tienen el mismo numero de reglas de producción, esto hace que se requiera un mayor numero de reglas para representar estas gramáticas, aun cuando

estas trayectorias puedan ser similares.

Por esto se busca representar de una manera compacta las trayectorias generadas y asi tener gramáticas más expresivas, lo cual se puede medir con el numero de reglas necesarias para representar la dinámica de un objeto y validarla con el numero de reglas necesarias para representar una trayectoria por 2 modelos de inferencia diferentes, donde el mejor modelo con mayor expresividad será el que utilice el menor número de reglas de producción para hacer dicha representación.

1.5. Planteamiento del problema

La comprensión automática del comportamiento de objetos en video vigilancia es un problema desafiante, ya que involucra la extracción, representación e interpretación de información visual para el entendimiento del comportamiento y reconocimiento de objetos, esta es una tarea que demanda alta capacidad de procesamiento para los ordenadores, ya que es necesario procesar y analizar la información de video. En [1] se crea un modelo gramatical para encontrar la dinámica de movimiento en un escenario fijo, donde las actividades de los objetos generan trayectorias que son representadas como cadenas de símbolos pertenecientes a un alfabeto finito, estas cadenas terminadas se introducen a un algoritmo de inferencia gramatical que genera reglas de producción, las cuales forman un conjunto de gramáticas para cada trayectoria . En consecuencia en este trabajo se plantea encontrar un método de inferencia gramatical, que genere reglas que describan la dinámica de movimiento para una o más cadenas de símbolos, entonces será posible comparar el numero de reglas pertenecientes a una ruta, que corresponden a la dinámica de un objeto en el escenario.

1.6. Alcances y Limitaciones

Los sistemas que modelan la dinamica de objetos usando gramaticas libres de contexto, basan su funcionamiento en las siguientes condiciones:

- La representación simbolica del movimiento, genera cadenas de simbolos correspondientes a las trayectorias de una ruta.
- Los patrones de movimiento corresponden a la dinamica de un objeto en el escenario.
- El modelo de la dinamica del movimiento se obtiene infiriendo una gramatica, sujeta a un tiempo de aprendizaje.

Bajo estas propociciones, todo objeto en movimiento generara trayectorias, la suma de todas estas trayectorias sujetas a un proceso de inferencia generara el modelo gramatical, donde las gramaticas aun siendo mas compactas representaran los mismos patrones de movimientos que el modelo gramatical original [1].
El tiempo de aprendizaje del proceso de inferencia estara sujeto a un modelo propuesto, el cual depende de la velocidad de aprendizaje del modelo.
El modelo no será aplicado a un sistema en tiempo real por la alta demanda de procesamiento y las complicaciones que esto implica, esta parte se dejara pendiente para un trabajo futuro.

Capítulo 2

Antecedentes

Los sistemas de monitoreo basados en visión han generado un nicho importante, pues permite el monitoreo semiautomático en escenarios concretos. Sin embargo, la detección de eventos depende del conocimiento previo del escenario, para la comprensión y caracterización del comportamiento de los objetos en el escenario. Como consecuencia cada objeto es identificado y seguido para determinar su actividad [3, 4]. El resultado de este proceso es un conjunto de trayectorias relacionadas con el movimiento de los objetos. A continuación se mencionan algunos trabajos que han enfocado sus investigaciones en desarrollar métodos para modelar la dinámica de objetos en un escenario.

2.1. Modelado de la escena basado en trayectorias

El seguimiento de objetos usando este método produce trayectorias con un conjunto de mediciones: posición, velocidad y aceleración. Las cuales se representan a través del tiempo en forma de una serie $f_1, f_2, f_3, ..., f_t$. Los sistemas basados en ellas buscan crear un modelo que permita el análisis del escenario que observan. Para tal fin estos modelos presentan una metodología bien definida (Figura 2.1). El cual se da a lo largo de tres pasos: (i) pre-procesamiento, (ii) clasificación y (iii) modelado.

Como en este modelo las trayectorias que se obtienen están asociadas a un tiempo (t) aunque dos objetos sigan la misma ruta si alguno de ellos toma mas tiempo en hacer el recorrido, las trayectorias que se obtendrán serán de diferente longitud. Esta diferencia no permite realizar inmediatamente una comparación entre los vectores para determinar su igualdad. Lo anterior plantea la necesidad de una etapa de pre-procesamiento. Otro punto es la similitud entre trayectorias ya que aunque tengan la misma longitud es necesario determinar cuales pertenecen a la misma ruta. Entonces entender una escena bajo esta metodología, como se

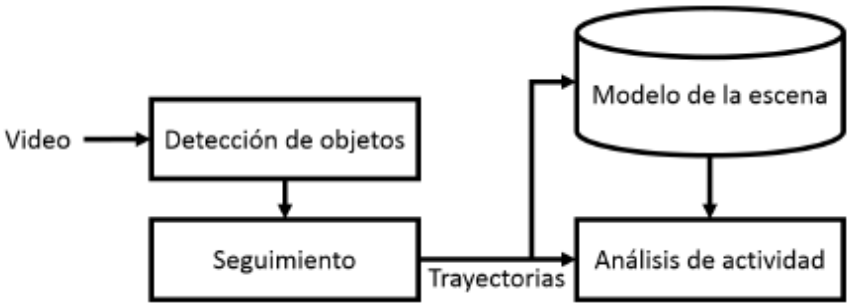

Figura 2.1: Constitución básica de un sistema de vigilancia basado en trayectorias [1]

ha mencionado, requiere encontrar trayectorias y clasificarlas entre similares. Por último una vez que las trayectorias se han clasificado. Las rutas encontradas son modeladas para una inferencia eficiente [5].

2.2. Modelado de la escena usando expresiones regulares

En este método de modelado se propone un una herramienta para el análisis de video que toma ventaja de las expresiones regulares para asociarlas automáticamente a actividades observadas y crear un modelo que las describa.

La observación y seguimiento de un objeto en un escenario, permite la caracterización del movimiento mediante una trayectoria. Una aproximación para lograr una representación simbólica de una trayectoria es sumarizar la concatenación de puntos significantes, por lo tanto una actividad podra ser descrita por su cadena de símbolos correspondiente.

Con este método se seleccionan manualmente las zonas de interés en el escenario, donde ahora cada trayectoria estará formada por la concatenacóon símbolos pertenecientes a las zonas de interés. Las reglas de movimiento correspondientes a las rutas de aprendizaje son extraídas y una Gramática libre de contexto es generada para cada actividad [2].
El método funciona de la siguiente manera:
Se hace el preprocesamiento de las rutas entrantes al sistema, se hace su representación

símbolica y descubrimiento automático de las gramáticas que codifican las reglas de las actividades aprendidas.

Cada trayectoria T es representada de la siguiente manera:

$$T = P_i, t_i; i = 0...N \tag{2.1}$$

Donde cada muestra P_i=(x_i,y_i) es la proyección en el plano del centroide del objeto en movimiento en el tiempo i, las rutas que puede seguir un objeto a dos mismos puntos se puede alcanzar de diferentes maneras, esta es la razón por la que en este método se definen las puntos de interés de manera manual, en lugar de considerar la trayectoria completa. Ahora las actividades son representadas por las posiciones obtenidas de la concatenación de los puntos de interés con los que el objeto ha tenido interacción. Esto permite simplificar la representación de actividad a un conjunto de regiones indexadas y su correspondencia temporal

$$T' = R_j, t_j; j = 0...M \tag{2.2}$$

Donde R_j es un punto de actividad indexado y t_j es la referencia temporal correspondiente.

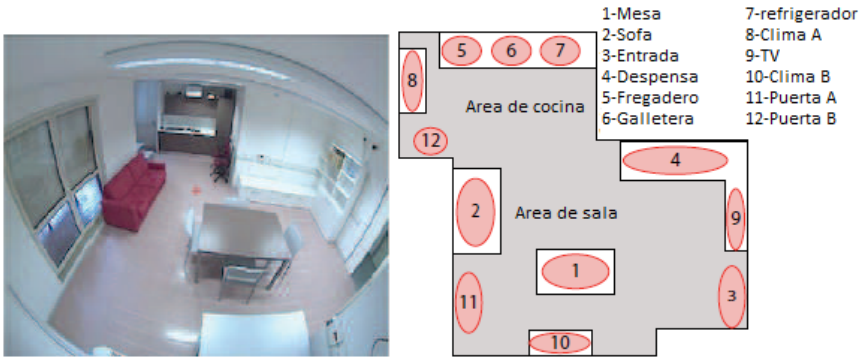

Figura 2.2: Mapa topológico de la escena con puntos de interés definidos [2]

Para reconocer el conjunto de actividades observadas en el escenario se aplica un algoritmo que analiza los símbolos que se van generando con la interacción del objeto y los puntos de

actividad definidos manualmente, estos símbolos forman una cadena la cual se codifica para ser representada por una gramática. Llamaremos $G(n)$ a un conjunto n de gramáticas y $G_max(i)$ a la longitud máxima de cualquiera de las posibles secuencias pertenecientes a la $i - esima$ gramática.

2.3. Generación automatica de gramáticas para la inferencia de actividad

Se plantea un método semi automático para modelar la dinámica dominante de un escenario como un problema de aceptación de una cadena respecto a una gramática. Para realizar esta tarea, se codifica el movimiento en estados que son asociados a una representación simbólica, luego, utilizando un conjunto de cadenas positivas a la gramática se induce un conjunto de reglas sobre las palabras generadas por el escenario.

2.3.1. Modelado del movimiento

El método se divide en cuatro etapas. La primer etapa, modelar movimiento, tiene como objetivo diferenciar los objetos en movimiento. Este proceso resulta complicado ya que depende de las condiciones de la cámara y del escenario, para ello la detección del movimiento en este método se efectúa mediante la diferencia entre la imagen actual (I_t) y el modelo de fondo (F_t) que representa a aquellos objetos que se mantienen estáticos en el escenario en el tiempo t como se define en la ecuación 2.3:

$$M_t = I_t - F_t \tag{2.3}$$

Se asume que F_t se aproxima como I_{t-1} es decir:

$$M_t = I_t - I_{t-1} \tag{2.4}$$

El uso de las imágenes directamente crea una alta detección de cambios lumínicos globales y ruido (destellos, reflejos, sombras y efectos por el formato de compresión. Para evitar esto existen adaptaciones del algoritmo. Una de ellas, diferencias temporales derivativas, hace uso de la norma k-ésima del gradiente de orden n de ambas imágenes. Por lo tanto, la ecuación 2.4 se reescribe de la siguiente forma:

$$M_t = L_k(\nabla^n I_t) - L_k(\nabla^n I_{t-1}) \tag{2.5}$$

Para modelar el movimiento se seleccionan los parámetros $n = 1$ y $k = 2$ de la ecuación 2.5. Por lo tanto se trabaja con $L_2(\nabla I_t(x, y)$, entonces la ecuación 2.5 se reescribe como:

$$M_t = ||(\nabla I_t)|| - ||(\nabla I_{t-1})|| \tag{2.6}$$

El uso de las derivadas de orden n en una imagen ofrece información sobre los cambios de intensidad. Los cambios de intensidad percibidos en una imagen determinan zonas de alto contraste, es decir cambios de color. A estos cambios de color es lo que se le denomina textura en una imagen. Altas frecuencias denotan un índice alto de textura, mientras que bajas frecuencias indican lo contrario. Esta variante evita una alta sensibilidad a cambios lumínicos globales al mismo tiempo que mantiene la simplicidad y baja complejidad del algoritmo.

2.3.2. Búsqueda de estados

T_t representa una plantilla del movimiento en la escena en el tiempo t. A partir de una secuencia de video del escenario y la ecuación 2.7 se obtiene una representación espacial de las zonas de movimiento. Generando una superficie $S(x, y)$ que representa la frecuencia y un estimado de la probabilidad con la que cada pixel puede presentar movimiento. En donde n es la cantidad de evidencia temporal, la cual debe ser suficiente para encontrar zonas de movimiento que conformen la dinámica del escenario.

$$S = \sum_{i=1}^{n} T_i \tag{2.7}$$

Entonces, un estado de movimiento es una región espacial de la proyección de la cámara que contiene un máximo local sobre la probabilidad de la existencia de movimiento en esta región. Por lo tanto, el problema de encontrar estados característicos se puede considerar como una tarea de segmentación. En esta propuesta tal tarea es efectuada mediante la transformada Watershed.

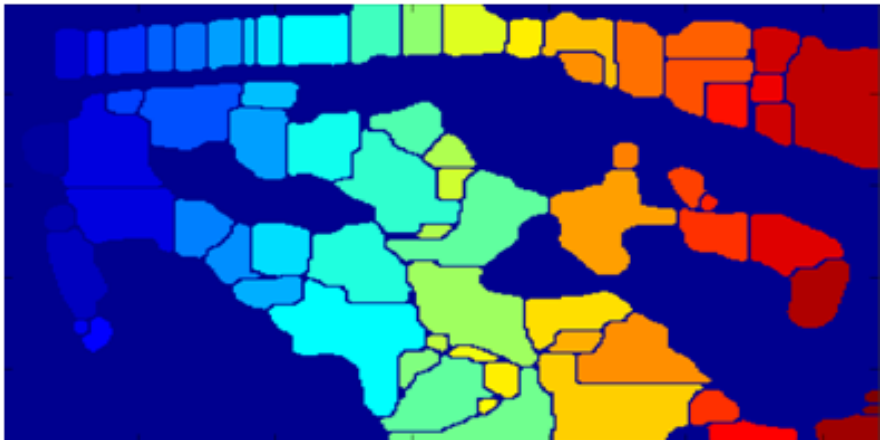

Figura 2.3: Estados encontrados de acuerdo al criterio de pertenencia al movimiento de cada pixel

Posteriormente se da lugar a la codificación del movimiento. La cual adecúa el desplazamiento de los objetos en forma de cadenas.

2.3.3. Codificación del movimiento

Una vez que se descubren los estados se procede a encontrar las gramáticas que modelen la dinámica del escenario. Para ello, primero se realiza una codificación para encontrar las cadenas que representen los ejemplos positivos de los lenguajes objetivo.
Para lo cual se asume que la frecuencia de ocurrencia de ejemplares negativos es baja, de forma que para una secuencia arbitraria en el tiempo, la probabilidad de observar una situación inusual es casi nula. La temporalidad de la secuencia debe contener suficientes ejemplares tanto para detectar las zonas de movimiento (estados) como las dinámicas típicas del escenario.

El proceso de codificación consiste en encontrar a que región r_i pertenece la posición de los centroides $cob_j i$ y así encontrar que símbolo $s_i \epsilon \Sigma$ se esta activando. De tal forma que a lo largo del tiempo se obtienen cadenas de las trayectorias de los objetos.

La generación de estas se vuelve trivial siempre y cuando se identifique a que objeto enlistado se debe agregar una de las nuevas posiciones.

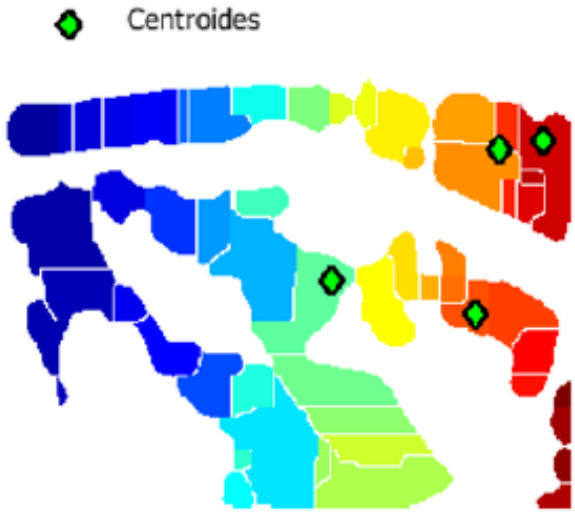

Figura 2.4: Representacion de la correspondencia de los centroides y zonas segmentadas [1]

2.3.4. Inferencia de gramáticas

El aprendizaje de las cadenas obtenidas con el paso anterior se da mediante la generación de reglas gramaticales correspondientes a cada cadena que pertenecen a $S_(n)$, que se representan como el conjunto de todas las gramaticas aprendidas G_n, el proceso se muestra en la Figura 2.5

La medida de similaridad para conocer la correspondencia de una cadena nueva con las rutas encontradas se basa en la búsqueda de la regla que contiene dicha cadena. De encontrarse se trata de una actividad normal, en caso contrario es una actividad anormal.

El proceso de aprendizaje consiste en tomar cada uno de los símbolos generados por el movimiento de un objeto en el escenario, donde al recorrer una trayectoria por completo, la concatenación de estos símbolos generan su representación como una cadena. Esta cadena de

símbolos pertenecientes al movimiento del objeto, es introducida al conjunto de gramáticas aprendidas correspondientes a las trayectorias generadas por los demás objetos en el escenario. Si esta cadena aun no ha sido aprendida por el modelo se introduce a la etapa de inferencia y se almacena como una gramática nueva, este proceso se repite por un tiempo determinado, con base en la especialización que se requiera por el modelo para describir actividades en el escenario.

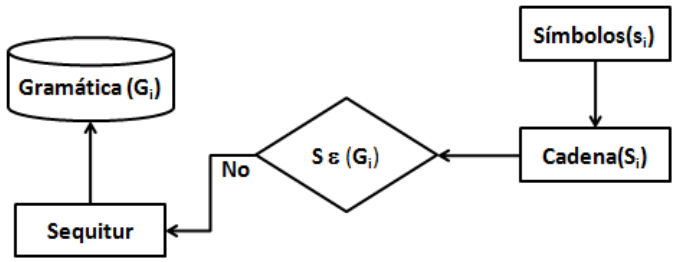

Figura 2.5: Diagrama de bloques de etapa de aprendizaje y generación de modelo gramatical [1]

Capítulo 3

Fundamento teórico.

En este capítulo se abordaran las bases sobre las cuales se desarrolla este trabajo.

3.1. Lenguaje y Gramática

3.1.1. Alfabeto

Un alfabeto es un conjunto de símbolos denotado por Σ. Una cadena de símbolos $x \in \Sigma$ es una secuencia finita de símbolos $|x| = |a_1, ..., a_n|$ donde Σ^* es el conjunto de todas las posibles cadenas que se generan con el alfabeto Σ

3.1.2. Lenguaje

El lenguaje L se define como $L \subseteq \Sigma^*$.Se representa por $\emptyset$ Un lenguaje es finito si contiene un numero finito de palabras e infinito en caso contrario. Se representa mediante $|L|$ número de palabras del lenguaje L.

3.1.3. Gramática libre de contexto

En el caso de una Gramática libre de contexto el ente formal que es capaz de representar a este conjunto de cadenas que pertenecen al lenguaje es una cuádrupla $\{\Sigma, V, R, N_1\}$donde Σes un alfabeto finito de terminales, V es un alfabeto finito de no terminales denotados por $N_1, ..., N_i, R \subseteq V \times (\Sigma \cup V)^*$ es un conjunto finito de reglas de producción, y $N_1 \in V$ es el axioma para representar la regla inicial.

3.1.4. Codificación de diccionario

Los métodos actuales construyen una estructura de diccionario en la que almacenan cada símbolo del alfabeto fuente que cuya representación se lleva a cabo a través de la palabra de código obtenida a partir de su posición en dicho diccionario. Estas mismas palabras de código se utilizan, en la etapa de codificación, para representar la ocurrencia de cada símbolo en el texto. Este proceso se desarrolla de forma sencilla sin más que sustituir cada símbolo en el mensaje por la palabra de código que lo identifica. Por lo tanto, el diccionario puede observarse como una función de mapeo "uno a uno" entre símbolos y palabras de código. La capacidad compresiva de estos métodos reside en que la longitud media de la palabra de código utilizada para la codificación es menor que la longitud media de los símbolos representados.

3.1.5. Sequitur

Sequitur es un algoritmo creado en 1997 por Nevill-Manning y Witten, de compresión basada en diccionario []. Capaz de inferir una estructura a partir de una secuencia discreta de símbolos, remplazando frases repetidas con una regla gramatical que derive la frase y de esta manera encontrar una representación sin pérdida de la entrada. El resultado es una abstracción jerárquica de la secuencia original que a su vez da una idea de su estructura [6]. Sequitur sigue dos restricciones para encontrar la gramática: unicidad de diagrama y utilidad de regla. La primera requiere que ningún par adyacente de símbolos aparezca más de una vez en la gramática. La segunda restricción asegura que toda regla se use al menos dos veces con excepción de la regla inicial. Por ejemplo:

Sea la cadena $S \rightarrow ababcdeab$, el algoritmo toma de símbolo en símbolo hasta que se tiene una pareja de símbolos repetida.

Entonces conforme se leen los símbolos la regla inicial se ve:

$N0 \rightarrow a, N0 \rightarrow ab, N0 \rightarrow aba, N0 \rightarrow abab$

como ab se repite dos veces la gramática queda:

$N0 \rightarrow R1R1, R1 \rightarrow ab$

se continúan añadiendo símbolos

$N0 \rightarrow R1R1c, N0 \rightarrow R1R1cd, N0 \rightarrow R1R1cde, N0 \rightarrow R1R1cdea, N0 \rightarrow R1R1cdeab$

como se encuentra que ab se ha repetido

$N0 \rightarrow R1R1cdeR1, R1 \rightarrow ab$

3.2. BISECCIÓN

Este algoritmo fue propuesto por Kieffer, Yang, Nelson y Cosman[referenciar!!!]. Este algoritmo funciona sobre una cadena de símbolos σ. Se selecciona el entero j de mayor longitud tal que $2^j < \sigma$. Se divide σ en dos subcadenas de longitud 2^j y $\sigma - 2^j$. Se repite este proceso de partición recursivamente en cada subcadena obtenida de longitud mayor a 1. Se crea un no terminal [7] para cada una de las distintas cadenas generadas en este proceso con longitud mayor a uno. Cada no terminal puede ser definido por una regla con símbolos a la derecha [8].

Ejemplo 1. Consideremos la cadena $\sigma = 1110111010011$. Particionamos recursivamente y asociamos un no terminal con cada una de las subcadenas generadas

$S \rightarrow T_1T_2$

$T_1 \rightarrow U_1U_1 \; T_2 \rightarrow U_21$

$U_1 \rightarrow V_1V_2 \;\; U_2 \rightarrow V_2V_3$

Donde:

$T_1 \rightarrow 11101110 \;\; T_2 \rightarrow 10011$

$U_1 \rightarrow 1110 \;\; U_2 \rightarrow 1001$

$V_1 \rightarrow 11 \;\; V_2 \rightarrow 10 \;\; V_3 \rightarrow 01$

3.3. SECUENCIAL

Kieffer y Yang desarrollaron un algoritmo similar a SEQUITUR [9, 6] pero mejorado el cual se describirá a continuación.

Secuencia procesa las cadenas como sigue. Comienza con una gramática vacía haciendo un recorrido de izquierda a derecha. En cada paso busca el prefijo de mayor longitud que coincida con la parte de la cadena ya procesada y añade ese no terminal a una nueva regla, si no se encuentra coincidencia del prefijo con algún símbolo procesado, entonces se agrega el primer terminal en la porción sin procesar para iniciar una nueva regla. En el caso que los nuevos símbolos de la regla aparezcan en algun lugar en la gramática sin traslaparse, se reemplazan por un nuevo no terminal que define a esos símbolos, finalmente sin algún no terminal aparece solo una vez después de su sustitución, se reemplaza este por su definición y se borra la regla correspondiente a este.

Ejemplo: Consideremos la cadena $\sigma = xxxxxDxxxxxC$ Conforme se leen los símbolos la gramática en el tercer símbolo es: $S \rightarrow xxx$. Cuando el cuarto símbolo se agrega a la regla inicial, hay 2 pares de símbolos xx. Por lo tanto una nueva regla es agregada a la gramática $R_1 \rightarrow xx$ y ambos símbolos xx repetidos son reemplazados por R_1 en la gramática, quedando como sigue:

$S \rightarrow R_1R_1$

$R_1 \rightarrow xx$

En los siguientes pasos conforme se procesa la cadena σ la regla inicial se expande a $S \rightarrow R_1R_1xD$. Continuando con el procedimiento de recorrido de simbolos al procesar casi la mayoría de símbolos de σ la regla inicial se vuelve a expandir $S \rightarrow R_1R_1xDR_1R_1$. Ahora el par R_1R_1 se repite 2 veces, una nueva regla es creada y aplicada a la regla inicial.

$S \rightarrow R_2xDR_2$

$R_2 \rightarrow R_1R_1$

$R_1 \rightarrow xx$

En el siguiente paso, x es agregada a la regla inicial $S \rightarrow R_2xDR_2x$. Ahora R_2x aparece dos veces. Una regla nueva es creada para $R_3 \rightarrow R_2x$ y sustituida en $S \rightarrow R_3DR_3$. Al final del

recorrido de todos los símbolos se tiene la siguiente Gramática:

$S \rightarrow R_3 D R_3 C$

$R_3 \rightarrow R_1 R_1 x$

$R_1 \rightarrow xx$

Como se observa la regla R_2 aparece solo una vez después de ser sustituida por R_3. Por lo tanto la ocurrencia de R_2 basado en la definición de R_3 es remplazada por R_1 [8].

3.4. LZ78

Es un algoritmo de compresión basado en diccionario. Las salidas codificadas por el algoritmo constan de dos elementos: Una referencia indexada a la coincidencia de símbolos de mayor longitud y el primer simbolo no terminal.

El algoritmo también agrega el indíce y símbolo codificado a el diccionario. Cuando un símbolo que aun no esta en el diccionario es encontrado, la codificacion tiene el valor 0 en el indíce. Con este método el algoritmo gradualmente construlle una gramática con reglas de producción.

Ejemplo: Sea la cadena $\sigma = 00121212102101$ a ser codificado usando LZ78, primero se busca la coincidencia de mayor longitud en la cadena inicial σ En el primer paso el diccionario esta vacio y no existen símbolos para comparar la coincidencia de mayor longitud. Así el primer símbolo de σ es $S \rightarrow 0$, este se agrega al diccionario y su codificación es:

$C_1 \rightarrow 00$

Donde el primer símbolo del terminal [7] es el índice de la conícidencia del símbolo, como en este caso no se encontró coincidencia porque el diccionario esta vacio, se asigna 0. Recorriendo la cadena original σ y tomando el segundo símbolo $S \rightarrow 00$ se observa que este símbolo ya se encuentra en el diccionario representado con C_1. Se busca la coincidencia de mayor longitud y se encuentra que es:

$S \rightarrow 001$

$C_1 \rightarrow 00$

$C_2 \rightarrow 11$

El diccionario incrementa conforme se recorre la cadena σ, se recorre un simbolo:

$S \rightarrow 0012$

$C_1 \rightarrow 00$

$C_2 \rightarrow 11$

$C_3 \rightarrow 02$

Como 2 no esta en el diccionario se crea una regla C_3 donde el primer terminal es el índice del símbolo, como este no tiene coincidencia se asigna el valor 0 para representarlo. Como se observa en la regla C_1. Se recorren todos los símbolos de la cadena σ hasta que se recorren todos los símbolos quedando el diccionario como sigue:

$S \rightarrow 00121212102101$

$C_1 \rightarrow 00 \rightarrow 0$

$C_2 \rightarrow 11 \rightarrow 01$

$C_3 \rightarrow 02 \rightarrow 2$

$C_4 \rightarrow 01 \rightarrow 1$

$C_5 \rightarrow 31 \rightarrow 21$

$C_6 \rightarrow 50 \rightarrow 210$

$C_7 \rightarrow 61 \rightarrow 2101$

La concatenación de las reglas del diccionario generado por LZ78 describen la cadena σ con 7 reglas de escritura.

3.4.1. LZ77

Es un algoritmo para compresión de datos sin pérdida de información. LZ77 emplea un principio llamado ventana deslizante, el cual asemeja pasar los datos por una ventana de un tamaño fijo, cualquier dato fuera de esta ventana no puede ser referenciado ni codificado. Entre más datos sean codificados, la ventana se desliza a lo largo de estos, Removiendo los

datos que ya han sido codificados de la vista y agregando nuevos datos a ser codificados.

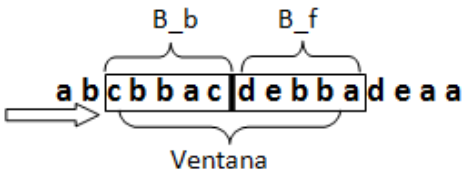

Figura 3.1: Ventana de tamaño 10, con buffer de búsqueda y Buffer futuro.

La ventana ($n = 10$) se divide en un buffer de búsqueda ($Bi = cbbac$), que contiene los datos que ya han sido procesados y en un buffer futuro ($Si = debba$), que contiene los datos que aun no se han codificado.

Los pasos para el funcionamiento del algoritmo se describen como sigue:

1) Se define el tamaño de la ventana "n", y la longitud de la palabra a codificar Ls.

2) Inicialmente el conjunto $B_1 = 0^{n-Ls}S(1, Ls)$, los primeros $n - Ls$ símbolos de B_1 son ceros, seguidos por los primeros L_S símbolos de la cadena S.

3) El puntero p_I es la posición del elemento del buffer de búsqueda que coincide con el primer elemento buffer futuro.

4) La palabra codificada es dada por:

$C_i = C_{i1}C_{i2}C_{i3}$ donde:

$C_{i1} = p_i - 1$

C_{i2} =longitud de $s_i = l_i - 1$

C_{i3} = El último elemento de s_i

5) Actualizar el buffer de búsqueda recorriendo los primeros l_i símbolos y introduciendo l_i nuevos símbolos en el buffer futuro.

Por ejemplo se tiene la siguiente cadena de símbolos a ser codificada:

$S = ababbacdcdbcb$

Se define el tamaño de la ventana $n = 10$, se define el tamaño de la palabra $L_s = 5$, de

acuerdo con el paso 2, quedaría:

$B_1 = 00000ababb$

Se busca la máxima coincidencia de elementos entre el buffer de búsqueda y futuro

$p_1 = 1; S_1 = a; l_1 = 1;$

Se codifica la primer palabra $C_0 = (0, 0, a) \rightarrow a$

Como no existe coincidencia entre elementos, se toma el primer elemento del buffer futuro y se recorren l_1 símbolos hacia el buffer de búsqueda y se introducen l_1 símbolos a el buffer futuro.

$B_2 = 0000ababba$

Se vuelven a repetir los pasos anteriores

$p_2 = 1; S_2 = b; l_2 = 1;$

Se codifica

$C_1 = (0, 0, b) \rightarrow b$

se toma el primer elemento del buffer futuro y se recorren l_2 símbolos hacia el buffer de búsqueda y se introducen l_2 símbolos a el buffer futuro

$B_3 = 000ababbac$

Se vuelven a repetir los pasos anteriores

$p_3 = 4; S_3 = abb; l_3 = 3;$

Se codifica

$C_2 = (3, 2, b) \rightarrow abb$

$C_3 = (2, 1, c) \rightarrow ac$

$C_4 = (0, 0, d) \rightarrow d$

$C_5 = (3, 2, b) \rightarrow cdb$

$C_6 = (2, 1, b) \rightarrow cb$

uniendo la palabra codificada

$C_0C_1C_2C_3C_4C_5C_6 = ababbacdcdbcb$

Se observa que al codificar la cadena completa se generan siete reglas para representarla.

Capítulo 4

Análisis de propuesta.

En este capítulo se muestra la propuesta de tesis. La cual consiste en encontrar un modelo que genere gramáticas mas expresivas, para representar actividades en un sistema de visión, donde la expresividad de estas gramáticas se medirá con el número de reglas generadas para un conjunto de trayectorias, comparado con las reglas generadas para el mismo conjunto de trayectorias con otro modelo gramatical. Para ello se codifica el movimiento en estados asociados con una representación símbolica, luego utilizando un conjunto de cadenas se induce un conjunto de reglas que representen estos movimientos.

La propuesta está compuesta por 5 etapas, la primer etapa consiste en encontrar estados para ello es necesario diferenciar los objetos de cambios lumínicos y ruido, solo así será posible segmentar el movimiento a partir de diferencias temporales derivativas, para encontrar estados y definir un conjunto finito de símbolos que relacione la frecuencia de pertenencia al movimiento de cada pixel con un alfabeto. En la segunda etapa se codifica el movimiento con lo cual se adecúa el desplazamiento de los objetos en forma de cadenas. En la tercer etapa se establece un criterio de tiempo de aprendizaje del modelo para no caer en sobre aprendizaje o aprendizaje pobre, ya que cualquiera de estas dos condiciones podría afectar la expresividad del modelo. En la cuarta etapa se infieren gramáticas para cada uno de los modelos, esta etapa consiste en introducir las mismas cadenas al algoritmo propuesto y el modelo a comparar, basado en las ideas de construccion de modelo ya mencionadas y así poder crear un modelo gramatical para tener punto de partida para la última etapa. La cual consiste en comparar el numero de reglas generadas para el conjunto de cadenas pasadas por cada uno de los modelos gramaticales y medir su expresividad, donde el criterio que se plantea para decidir qué modelo es más expresivo, consiste en comparar numero de reglas que genera cada uno de los modelos y el modelo que menos reglas genere, sera el mas expresivo.

4.1. Encontrar estados

Para encontrar estados es necesario modelar el movimiento, para esto se seleccionan los parámetros $n = 1$ y $k = 2$ de la ecuación 4.1.

$$M_t = L_k(\nabla^n I_t) - L_k(\nabla^n I_{t-1}) \tag{4.1}$$

Donde:

$$L_k(\nabla^n I_t) = \| \frac{\partial^n I_t}{\partial x^n}, \frac{\partial^n I_t}{\partial y^n} \|_k$$

Por lo tanto se trabaja con $L_2(\nabla I_t(x, y))$, entonces la ecuación 4.1 se reescribe como:

$$M_t = ||(\nabla^n I_t(x, y))|| - ||(\nabla^n I_{t-1}(x, y))|| \tag{4.2}$$

El uso de derivadas en una imagen ofrecen información sobre los cambios de intensidad. Los cambios de intensidad percibidos en una imagen determinan zonas de alto contraste, es decir cambios de color. A estos cambios de color es lo que se le denomina textura en una imagen. Altas frecuencias denotan un índice alto de textura, mientras que bajas frecuencias indican lo contrario.

Una vez que se separan a aquellos pixeles que tienen una alta probabilidad de pertenecer a objetos en movimiento se modela cada uno a través del tiempo con una función de decaimiento exponencial recursiva 4.3. Esta ecuación permite ser tolerable ante cambios abruptos en regiones debidas a reflejos y destellos, ya que cualquier elemento $M_t(x; y)$ necesita excitarse con una frecuencia determinada para ir incrementando su energía y acercarse al valor máximo. De lo contrario esta se irá perdiendo hasta llegar exponencialmente a un cero relativo en un tiempo t. Tanto la frecuencia como el tiempo t se relacionan directamente con el coeficiente de decaimiento, el cual se calcula con la ecuación 4.4.

$$T_t = \alpha T_{t-1} + (1 - \alpha)M_t \tag{4.3}$$

$$\alpha = 1 - \frac{1}{f_v T_m} \tag{4.4}$$

Donde f_v es la frecuencia de adquisición del sensor y T_m es la temporalidad del movimiento a segmentar.

Cada T_t representa una plantilla del movimiento en la escena en el tiempo t. A partir de una secuencia de video del escenario y la ecuación 4.4 se obtiene una representación espacial de las zonas de movimiento. Generando una superficie $S(x, y)$ que representa la frecuencia y un estimado de la probabilidad con la que cada pixel puede presentar movimiento. En donde n es la cantidad de evidencia temporal, la cual debe ser suficiente para encontrar zonas de movimiento que conformen la dinámica del escenario.

$$S = \sum_{i=1}^{n} T_i \tag{4.5}$$

Entonces, un estado de movimiento es una región espacial de la proyección de la cámara que contiene un máximo local sobre la probabilidad de la existencia de movimiento en esta región. Por lo tanto, el problema de encontrar estados característicos se puede considerar como una tarea de segmentación. En esta propuesta tal tarea es efectuada mediante la transformada $Watershed$.

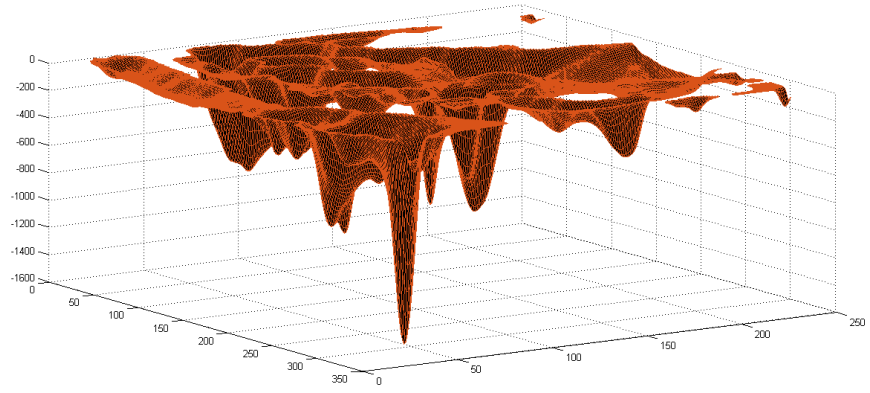

Figura 4.1: Superficie generada al aplicar la ecuacion 4.5

Al aplicar la ecuación 4.5 se obtiene una superficie (Fig. 4.1), la cual mantiene zonas de la imagen relacionadas con el movimiento. En donde las zonas de máximos representan una buena referencia para encontrar estados. Los estados se encuentran aplicando una segmentación morfológica. Específicamente la transformada Watershed. Se realiza la operación $L = Watershed(SG(x, y))$. Como resultado L se constituye por un conjunto de

regiones disjuntas. La técnica puede detectar zonas no aptas para convertirse en estados. Denominados estados falsos. Como consecuencia de reflejos continuos, generalmente ocasionados por superficies reflectivas como ventanas y zonas del pavimento. Por ello una vez dado el proceso de etiquetamiento se lleva a cabo una prueba de consistencia. Aquí las zonas etiquetadas en L deben cumplir las siguientes condiciones para convertirse en estados:

- Un estado tiene área suficiente para detectar movimiento.
- Ningún estado debe estar aislado.

La primer condición se plantea para asegurar que los estados obtenidos tengan un área mínima para ser validos. Esta restricción evita la selección de estados pequeños causados por pixeles o zonas ruidosas o debido a problemas de la cámara o elementos reflectivos en el escenario [1]. La segunda condición es necesaria desde el punto de vista de la semántica del escenario, debido a que cualquier objeto que se desplace en la escena, para ser caracterizado debe ser representado como la sucesión de estados espacialmente adyacentes. Por lo tanto estados aislados representan movimiento aislado y no trayectorias. Entonces, la adyacencia de los estados debe garantizarse como se observa en la Fig 4.2.

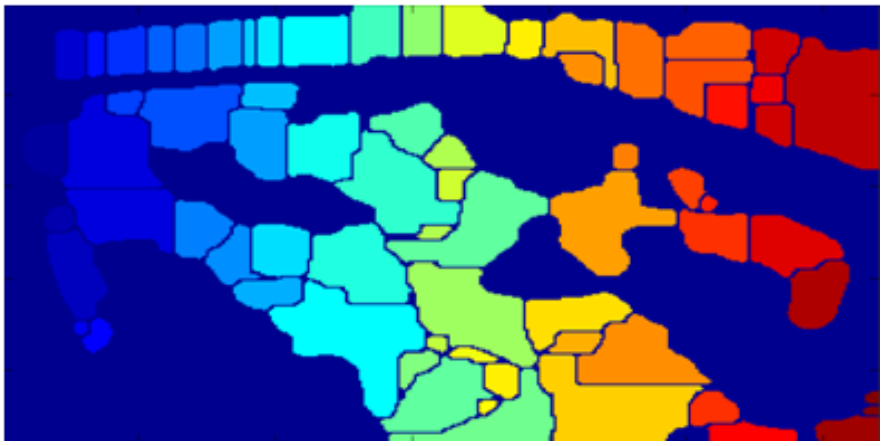

Figura 4.2: Regiones encontradas

4.2. Codificar el movimiento

Una vez que se descubren los estados se procede a encontrar las gramáticas que modelen la dinámica del escenario. Para ello, primero se realiza una codificación para encontrar las cadenas que representen los ejemplos positivos de los lenguajes objetivo. Para lo cual se asume que la frecuencia de ocurrencia de ejemplares negativos es baja, de forma que para una secuencia arbitraria en el tiempo, la probabilidad de observar una situación inusual es casi nula. La temporalidad de la secuencia debe contener suficientes ejemplares tanto para detectar las zonas de movimiento (estados) como las dinámicas típicas del escenario. Para esto se segmenta el movimiento con la metodología empleada en una etapa anterior. Sin embargo, en esta ocasión se cuenta el número de elementos contenidos en la escena.

Figura 4.3: Relacion entre la correspondencia de un estado con la imagen

Esto da la noción de cuantos objetos Obj_i se encuentran en movimiento. A partir de este número se crea una lista de tamaño igual a la cantidad de elementos activos y se añade la codificación de posición del centroide $cObj_i$ de cada objeto. Al tratarse de matrices que mantienen el mismo marco de referencia el proceso de codificación consiste en encontrar a

que región r_i pertenece la posición de los centroides cObji y así encontrar que símbolo $s_i \in \Sigma$ se esta activando (Figura 4.3). De tal forma que a lo largo del tiempo se obtienen cadenas de las trayectorias de los objetos. La generación de estas se vuelve trivial siempre y cuando se identifique a que objeto enlistado se debe agregar una de las nuevas posiciones. Para ello se consideran dos criterios espaciales. El primero es la distancia máxima l_max a la cual dos centroides se encuentran para decir que corresponden a un mismo Obj_i. El segundo es que se cumpla $|adj(r_{st}, r_{st-1})| = 1$ entre los últimos símbolos anexados a las cadenas y los recién codificados. Al terminar este proceso las cadenas encontradas tienen la siguiente estructura S_1 =71, 71, 71, 71, 71, 71, 67, 67, 67, 67, 67, 67, 67, 61, 61, 61, 61, 61, 61, 61, 61, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53. La cual corresponde a la trayectoria generada por un objeto en movimiento.

4.3. Convergencia en el aprendizaje

Posteriormente, se define un conjunto lo suficientemente grande para determinar que el sistema ha aprendido, en donde no se estipula criterio alguno para este parámetro tan importante, más que el necesario. A partir de este conjunto de cadenas se comienza el siguiente procedimiento de aprendizaje

1. se construye un conjunto de gramáticas en un principio vacío.
2. se toma una cadena y se verifica si existe alguna gramática capaz de aceptarla.
3. si no existe la cadena se introduce al algoritmo de inferencia y se genera una nueva gramática.
4. si existe la cadena se representa como un patron de movimiento ya aprendido.
5. se repite hasta que las cadenas se acaben.

   En la figura 4.5 se muestra la gramática encontrada una vez que algoritmo de inferencia gramatical procesa la cadena S_1 [1].

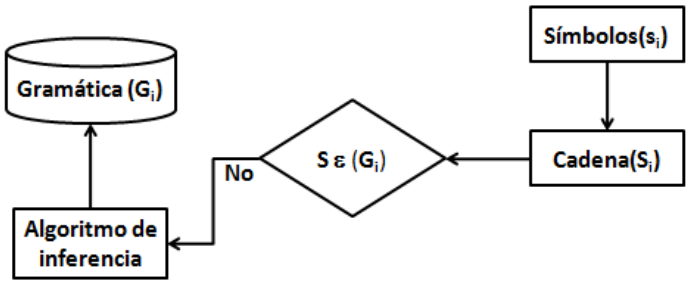

Figura 4.4: Diagrama de bloques fase de aprendizaje

Bajo esta metodología cada gramática representa una trayectoria que se considera comportamiento normal para el escenario. Entonces el modelo, que es un conjunto de gramáticas, no es más que el conjunto de trayectorias aceptadas como normales en el escenario. Por ende cualquier cadena que no sea aceptada por el modelo se trata de una trayectoria anormal y puede requerir de mayor atención.

El modelado del sistema se da mediante un tiempo de aprendizaje definido, donde la aceptación o rechazo de una cadena determina si la ruta puede ser descrita por alguna de las gramáticas aprendidas.

$N_0 \rightarrow N_1, N_1, N_1, N_2, N_2, N_2, 67, N_4, N_4, N_8, N_8, N_7, N_6, 53$

$N_1 \rightarrow 71, 71$

$N_2 \rightarrow 67, 67$

$N_3 \rightarrow 61, 61$

$N_4 \rightarrow N_3, N_3$

$N_5 \rightarrow 53, 53$

$N_6 \rightarrow N_5, N_5$

$N_7 \rightarrow N_6, N_6$

$N_8 \rightarrow N_7, N_7$

Figura 4.5: Gramática G_1 correspondiente a S_1 y reglas de producción derivadas

De lo anterior es prudente, preguntarse, ¿cuánto tiempo se debe de ejecutar la fase de aprendizaje?, para contestar esta pregunta primero se analizar la manera en que se

construye el modelo acorde a la metodología.

El modelado del sistema se da mediante un tiempo de aprendizaje definido, donde la aceptación o rechazo de una cadena determina si la ruta puede ser descrita por alguna de las gramáticas aprendidas. De lo anterior es prudente, preguntarse, ¿cuánto tiempo se debe de ejecutar la fase de aprendizaje?, para contestar esta pregunta primero se analizar la manera en que se construye el modelo acorde a la metodología.

4.3.1. Gramáticas y la temporabilidad

Se tiene 3 cadenas distintas, S_1,S_2,S_3.

S_2 =71, 71, 71, 71, 71, 71, 67, 67, 67, 67, 67, 67, 67, 61, 61, 61, 61, 61, 61, 61, 61, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53

$N_{02} \rightarrow$ N1, N1, N1, N2, N2, N2, 67,N4, N4, N8, N8, N7, N6, N5,53

S_3=71, 71, 71, 71, 71, 71, 67, 67, 67, 67, 67, 67, 67, 61, 61, 61, 61, 61, 61, 61, 61, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53

$N_{03} \rightarrow$ N1, N1, N1, N2, N2, N2, 67,N4, N4, N8, N8, N8,53

Al comparar S_1, S_2, S_3, se observa que contiene el mismo patrón de estados lo cual significa que siguen la misma ruta de movimiento $(71, 67, 61, 53)$. Sin embargo su representación gramatical difiere pues la frecuencia con la que aparece cada uno de los símbolos cambia, entonces sólo la regla inicial N_{02} puede expresar a S_2, de la misma manera N_{03} sólo puede expresar a S_3.

Al realizar esta observación se entrevé que las repeticiones temporales pueden descartarse si el trabajo considera el patrón de desplazamiento y no la duración o rapidez con la que se realiza. Para corroborar lo anterior se realizan pruebas sobre el escenario.

la cual corresponde a una ruta recorrida por un objeto, introduciendo esta cadena en la etapa mencionada en la figura 4.5 se obtiene la siguiente gramática.

Se toman 3174 y se infiere una gramática sobre ellas, al final el proceso de inferencia se observa que el numero de gramáticas generadas es el mismo que el de las cadenas inferidas (Figura 4.6).

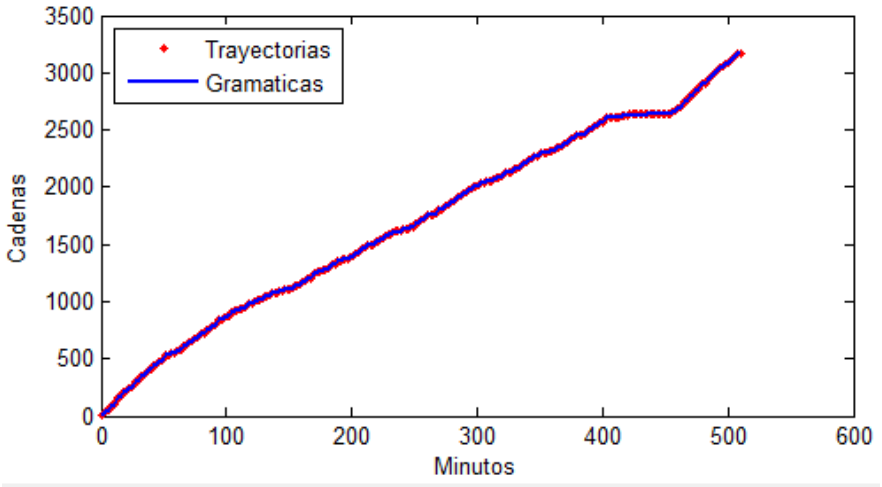

Figura 4.6: Gramaticas genereadas en un intervalo de tiempo

El problema de estos resultados es la cantidad de información que se necesita para representar las trayectorias, además de que la falta de repetitividad en las trayectorias hace casi imposible que el algoritmo converja y por lo tanto llegue a una estabilidad en el aprendizaje. Por ejemplo para S_1 se generan 8 reglas de producción y a su vez S_2 y S_3 tienen 8 reglas cada una. Por lo tanto para representar sólo estas 3 trayectorias, con el modelo de aprendizaje original se necesitan 24 reglas de producción. Lo que significa una mayor necesidad de información para representar lo mismo.

Al observar estos resultados se crea la idea de qué pasa si se modifica la manera en que se introducen los datos en Sequitur, surge una nueva pregunta ¿existirá una gramática que pueda representar a más de una trayectoria? Una forma de responder esta pregunta consiste en eliminar la repetitividad de las cadenas (trayectorias), por consiguiente al menos un par

de cadenas correspondientes a una ruta pueden ser expresadas por una misma gramática. Se toma como ejemplo las cadenas anteriores, sin repeticiones: $S'_1 \rightarrow 71, 67, 61, 53$, $S'_2 \rightarrow 71, 67, 61, 53$, $S'_3 \rightarrow 71, 67, 61, 53$. Al realizar esta modificación al introducir estas cadenas en la fase de aprendizaje, se generara una gramática única $N_{01} \rightarrow 71, 67, 61, 53$

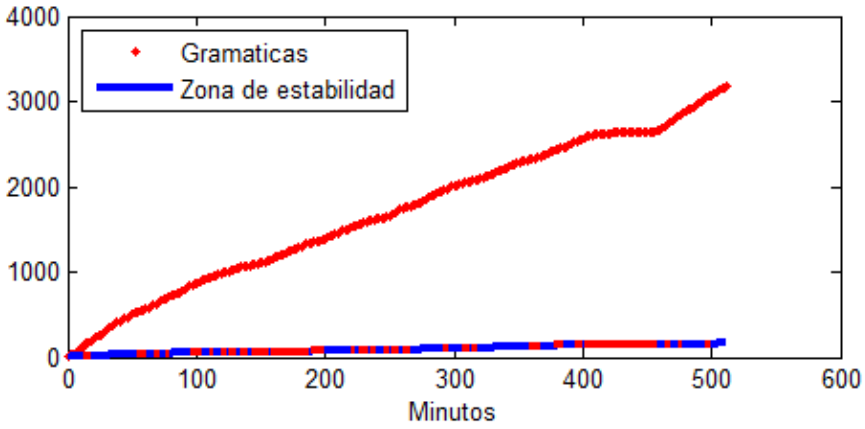

Figura 4.7: Gramáticas generadas para trayectorias sin repeticiones

En este punto el numero de gramáticas aprendidas en el tiempo tiene que comenzar a converger en algún momento, al realizar la experimentación se observa que la idea general es adecuada pues como se observa en la Figura 4.7, al eliminar las repeticiones de las cadenas se observa que el numero de reglas necesario para describir el mismo número de cadenas disminuye considerablemente por lo tanto se puede asumir que al necesitar menos reglas para describir el mismo número de trayectorias, el tiempo de aprendizaje puede reducirse considerablemente, ya que al eliminar las repeticiones una misma gramática puede representar dos o más trayectorias.

En la curva C que se muestra en la Figura 4.7 se observan 2 momentos donde el numero de gramáticas aprendidas a lo largo del tiempo incrementan drásticamente. Entonces se realiza un criterio de estabilidad que sea capaz de realizarse de manera semiautomático. Para ello se establece lo siguiente: se toma un valor de tiempo (en minutos) para considerar que el sistema es estable, en la Figura 4.8 se aprecia en rojo las zonas donde el sistema se mantuvo estable, no todos los intervalos de estabilidad son de interés, porque los intervalos de interés

deben ser representativos de la curva C, para resolver el problema de selección, se grafica las longitudes de los intervalos de estabilidad encontrados en la Figura 4.8

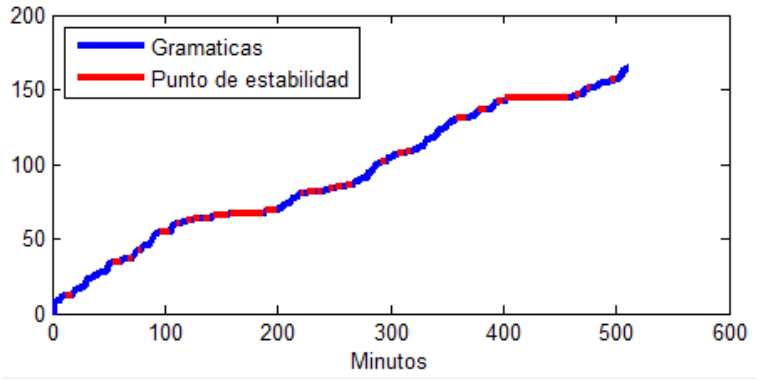

Figura 4.8: Puntos de estabilidad para gramáticas aprendidas

En la Figura 4.9 se observan dos locales máximos, que correspondes a los intervalos de máximo aprendizaje del modelo gramatical, asi que el siguiente paso es usar la transformada máxima extendida para encontrar el máximo local de los intervalos característicos de estabilidad de la curva C.

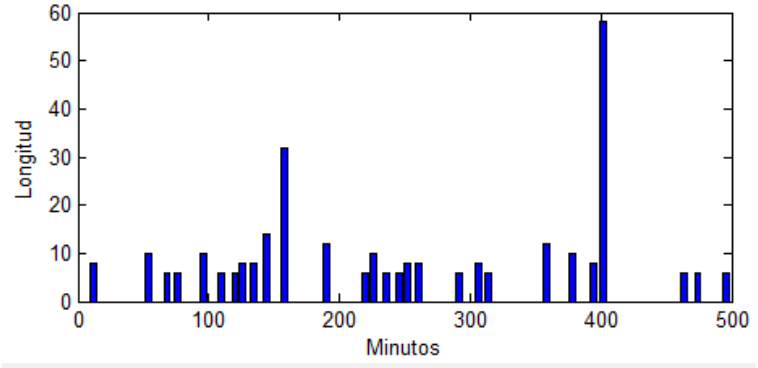

Figura 4.9: Longitud de puntos de estabilidad de la curva C

Ahora solo una sección de la curva C se segmenta, especifícamente la que pertenece al intervalo de 0 a 158 minutos. Analizando este intervalo la siguiente curva es propuesta:

$$\hat{y} = a + b\, ln x \tag{4.6}$$

Esta curva muestra la velocidad de aprendizaje de las gramáticas, durante el intervalo de

tiempo mencionado. Una curva continua se crea, para analizar las características de la derivada de primer orden en un punto, las cuales se relacionan con la curva original C, cuando comienza a crecer lentamente se asocia con la idea de que se llego a un punto donde las grámaticas comienzan a ser aprendidas mas lentamente, este factor dw aprendizaje puede expresarse como:

$$\frac{d\hat{y}}{dx} = k \tag{4.7}$$

Si la constante de la ecuación 4.7 es mayor a uno, significa que se han encontrado varias gramáticas nuevas a lo largo del tiempo, si la constante k es igual a uno significa que la nueva gramática aprendida ya existe. Entonces tomando en cuenta esto, el criterio para establecer el tiempo de aprendizaje es seleccionando un valor de k

4.4. Inferir gramática

Con el criterio k definido para el tiempo de aprendizaje, se toman las cadenas generadas por el movimiento de los objetos en el escenario, donde ahora el conjunto de cadenas se unen para formar una cadena $S_G \rightarrow S_i, S_{i+1}, ..., S_n$, donde S_G es la cadena generada por la concatención de las cadenas $S_i, S_{i+1}, ..., S_n$ generadas en un tiempo t para ser introducidas al proceso de inferencia de gramaticas. El proceso formado por los bloques planteados en la figura 4.10 corresponden a una fase de aprendizaje. Cuyo objetivo es adquirir el modelo de la dinámica del escenario. Este modelo esta formado por un conjunto de gramáticas $G_G \rightarrow G_i, G_{i+1}, ..., G_n$. Donde cada gramática $G_i \rightarrow R_i, R_{i+1}, ...R_n$ cuenta con su conjunto de terminales, no terminales, estado inicial y reglas de escritura que expliquen las cadenas S_i obtenidas en la fase de entrenamiento.

Al seguir esta fase de aprendizaje se realiza una metodología capaz de inferir actividades normales o anormales en un sistema de visión de manera automática. Para ello es necesaria una codificación del movimiento, basándose en la generación de cadenas previamente mencionada. Posteriormente, la medida de similaridad ha utilizar para conocer la correspondencia de una cadena nueva con las rutas encontradas se basa en la búsqueda del

lenguaje $L(G_i)$ que contiene dicha cadena. De encontrarse se trata de una cadena referente a una gramática aprendida por el modelo.

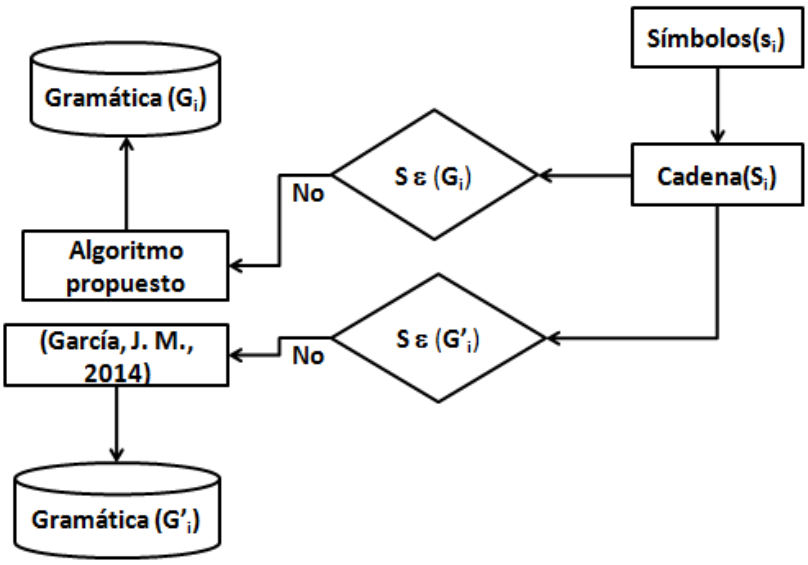

Figura 4.10

4.4.1. Expresividad del modelo

Con el modelo de gramáticas obtenido por la parte de aprendizaje se cuenta con dos lenguajes $L(G_i)$ y $L(G_i')$ de los cuales se comparara el numero de reglas generadas por cada uno de ellos para representar un trayectoria. La selección del algoritmo que se usara para validar la expresividad de las gramáticas se realizara comparando el modelo original [1] con un conjunto de algoritmos de inferencia, para analizar cual de estos ofrece una mejor descripción de la gramática generalizando las reglas requeridas para expresar una cadena. Para ello se toman un conjunto de cadenas para su análisis con cada uno de los algoritmo mencionados.

$S_1 \rightarrow aaaaaaaabbbbbbbbccccccccddddddddd$

$S_2 \rightarrow aaaaaaaabbbbbbbbccccccccddddddddddd$

$S_3 \rightarrow aaaaaaaabbbbbbbbccccccccddddddddddd$

Al introducir estas cadenas en cada uno de los algoritmos mencionados se obtiene el siguiente resultado:

Algoritmo	Reglas para S_1	Reglas para S_2	Reglas para S_3
SEQUITUR	8	9	9
BISECCION	8	8	8
LZ77	5	5	5
LZ78	15	15	15
SECUENCIAL	8	8	8

Tabla 1. Gramáticas vs Trayectorias, con repeticiones

La discriminación de los algoritmos se hace basado en el numero de reglas de producción necesarias para describir el conjunto de cadenas analizadas, se observa que con SEQUITUR, BISECCION Y SECUENCIAL, se obtienen resultados muy parecidos, donde LZ78 es el que mas reglas genera y LZ77 es el que menos reglas genera. Por lo tanto el algoritmo con el que se van a realizar pruebas para comparar el numero de reglas generadas en un escenario contra SEQUITUR.

El proceso para validar la propuesta se realizara siguiendo la metodología seguida en [1] usando un algoritmo que mejores resultados arrojo, basado en el análisis mencionado anteriormente.

Capítulo 5

Modelo Experimental y resultados

A continuación se presentan los resultados experimentales obtenidos tras el análisis de los algoritmos presentados en la sección anterior, se valida la propuesta comparando los modelos en 2 escenarios cada uno con características topológicas y morfológicas diferentes.

5.1. Condiciones experimentales

5.1.1. Infraestructura

En cuanto a la infraestructura se cuenta con 2 escenarios con características diferentes en cada uno de ellos, de los cuales uno son las instalaciones del Centro de Ingeniería y Desarrollo Industrial (CIDESI) dentro del laboratorio de modelado, simulado y visión por computadora. El escenario de prueba se encuentra dentro del edificio donde se localiza el laboratorio y a continuación se muestra una imagen. Se cuenta con una cámara con una resolución de 320 X 240 pixeles a 30 cuadros por segundo. La cual estuvo fija e inmóvil durante la adquisición de la información.

Figura 5.1: Escenario de pruebas 1

El segundo escenario es un cruce vehicular donde se observa el flujo de coches y personas en ambos sentidos, la interacción entre los vehículos y objetos es alta, ya que es un área concurrida con alto tránsito vehicular y peatonal, por lo tanto los objetos a detectar son de distintas magnitudes lo que hace el procesamiento más complicado, lo que será un buen punto de partida para validar el algoritmo planteado, la cámara que se uso se encuentra en la parte más alta de un edificio de 25 metros de altura, la cámara está configurada para adquirir 15 fps con resolución de 320 x 240 pixeles, en la Figura 5.3 se observa la perspectiva visual de la cámara y el flujo de movimiento.

Como se menciono en cada uno de los escenarios se cuenta con una cámara con características diferentes, en cuanto a la resolución y adquisición de información, además se cuenta con dispositivos para comunicar la cámara con el dispositivo de procesamiento en este caso una computadora portátil, en la Figura 5.2 se observan los componentes utilizados y sus características principales.

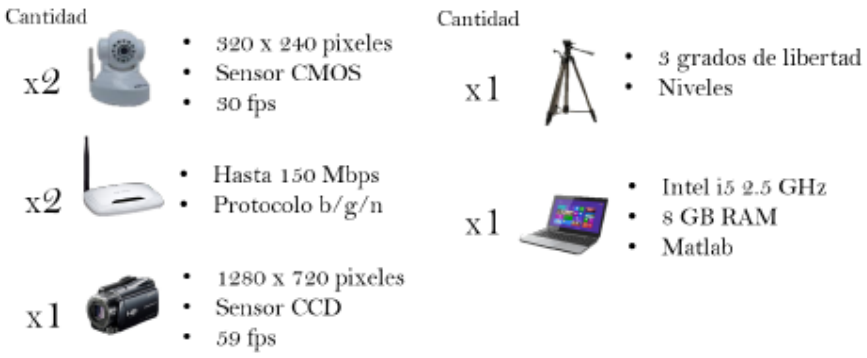

Figura 5.2: Material usado para la validacion de la propuesta

Los escenarios de prueba presentan varios niveles de estructura. El primero se conforma por la planta baja de un edificio altamente concurrido, el segundo es un cruce vehicular, en cada uno de estos escenarios la perspectiva de la cámara así como la distancia son diferentes lo que es bueno para el análisis de la propuesta ya que no se manejan similitudes en la adquisición de las imágenes lo que hace mas robustos y reales los resultados obtenidos.

Figura 5.3: Escenario de prueba 2

5.1.2. Inicialización parametrica

Siguiendo la metodología planteada para realizar la validación del algoritmo propuesto en el Capitulo 4 que consiste en 5 etapas, donde el objetivo es segmentar el escenario para poder extraer cadenas correspondientes al movimiento de los objetos y así inferir una gramática mas expresiva.

i) Encontrar estados. Se comienza por modelar el movimiento, donde el objetivo es diferenciar los objetos en movimiento del fondo de la imagen, este proceso se logra usando la ecuación 4.2, como se observa en la figura 5.4, el fondo y los objetos estáticos se eliminan de la escena.

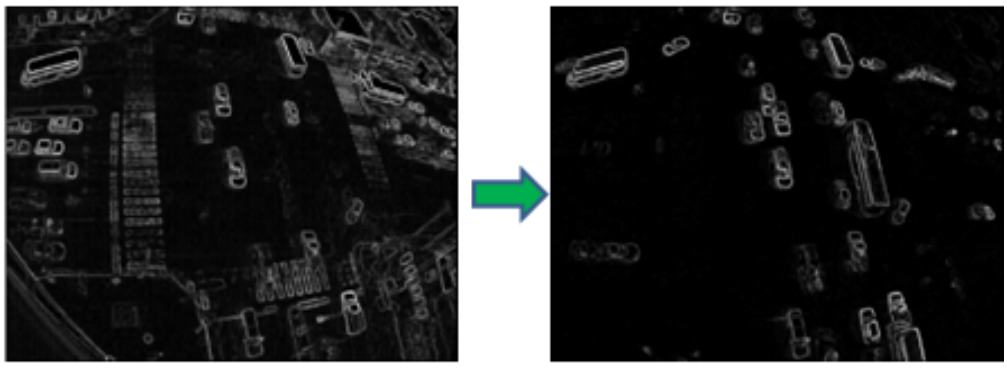

Figura 5.4: Supresion de fondo

Con este proceso ya es posible identificar y etiquetar cada uno de los objetos en movimiento para calcular la pertenecía al movimiento de cada pixel en la escena, este proceso se logra haciendo un acumulamiento respecto al tiempo de las regiones donde ocurre el movimiento para al finalizar obtener una superficie como la de la siguiente figura, donde las regiones de maximo movimiento son las zonas en color rojo y las regiones donde hubo poco movimiento se representa con color azul, como se observa en la figura 5.5.

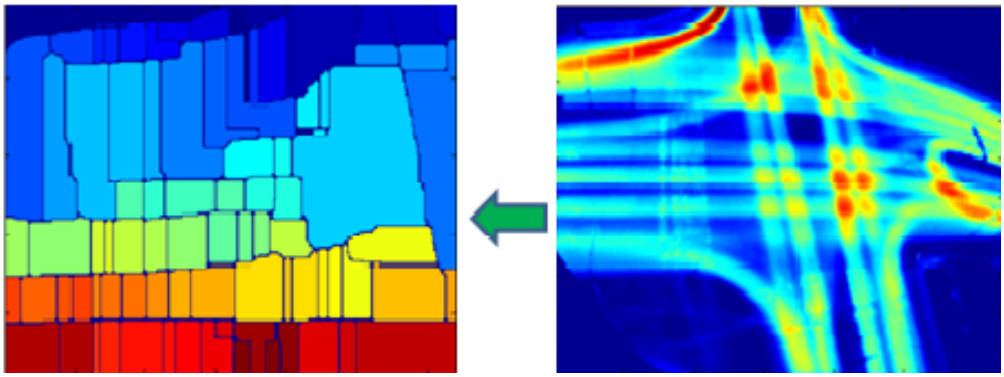

Figura 5.5: Pertenencia al movimiento de los objetos

La representación en forma de superficie de la figura 5.5 corresponde a montañas en las regiones de máxima pertenencia al movimiento y superficie plana a las zonas donde no se registro pertenencia, para aplicar la transformada watershed, se hace un proceso de invertimiento de esta superficie donde los máximos locales representados como montañas ahora serán mínimos locales y así poder hacer el proceso de inundamiento, para segmentar la imagen y encontrar los estados pertenecientes al movimiento de los objetos en la escena.

ii) Codificar el movimiento. Ahora es posible, la codificación del movimiento, donde cada estado de la imagen tiene un símbolo asignado y la interacción de un objeto en movimiento con cada estado genera un símbolo, para la representación de la trayectoria seguida a lo largo del tiempo.

Donde al encontrarse un objeto en el área perteneciente a un estado, este genera un símbolo respecto al tiempo y al hacer el recorrido por la escena, el objeto pasa por al menos dos estados de la escena para completar su trayectoria, al finalizar el recorrido, todos los

símbolos obtenidos por la interacción con los estados forman una cadena que corresponde a la dinámica de este objeto en la escena.

Las cadenas obtenidas son de la forma $S_i \rightarrow a_1a_2...a_n, S_{i+1} \rightarrow a_1a_2...a_n, ..., S_n \rightarrow a_1a_2...a_n$

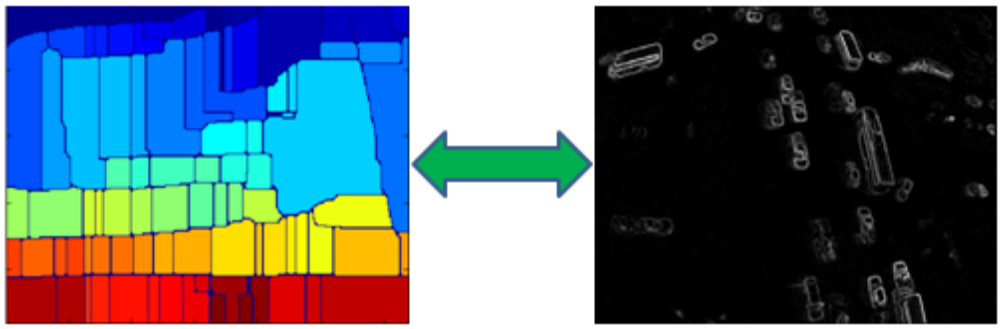

Figura 5.6: Codificacion del movimiento

Con las cadenas obtenidas se procede a estimar el tiempo de aprendizaje necesario para las trayectorias obtenidas en el escenario.

iii) Convergencia del aprendizaje. Este proceso se logra estimando el tiempo de aprendizaje necesario por el sistema para aprender el mayor numero de gramáticas. sin caer en un aprendizaje pobre o en caso contrario un sobre aprendizaje, ya que estos dos casos resultarían en problemas para describir la dinámica de los objetos en la escena. Esto porque al tener un aprendizaje pobre, los patrones de movimiento para describir la dinámica de objetos no serian suficientes ya que se tendría un conjunto pequeño de cadenas para describir las actividades de los objetos, donde actividades correctas podrían tomarse por el modelo como incorrectas, caso contrario con un sobre aprendizaje, el conjunto de gramáticas seria muy grande y las actividades incorrectas podrían tomarse como aceptables por el modelo

Por esto el criterio de aprendizaje para el modelo se da aplicando la ecuación 4.7 a la curva generada por el conjunto de cadenas obtenidas, por los objetos en movimiento en la escena, en este caso los objetos son vehículos, la curva está sujeta a la circulación con respecto el

tiempo, ya que la circulación de coches a lo largo del día no es uniforme ni constante y está sujeta a las horas de mayor tráfico, de esto se entrevé que la curva generada no será constante y tendrá zonas donde no se aprendió ninguna gramática, estas zonas son las que marcan pauta para poder decir que más de una nueva cadena ya ha sido aprendida por el modelo

iv) Inferir gramática. Con el tiempo estimado de aprendizaje, se introducen las cadenas terminadas ala fase de aprendizaje donde todas las cadenas que hayan sido aprendidas por el modelo, pertenecientes al conjunto de cadenas del tiempo de aprendizaje estimado por la ecuación 4.7, se descartaran al pertenecer ya al modelo, en caso contrario estas cadenas se introducirán al algoritmo de inferencia LZ77 y al algoritmo de inferencia SEQUITUR. Las gramáticas obtenidas por cada uno de estos modelos son de la forma $G_i \rightarrow R_i, R_{i+1}, ..., R_n$ donde G_i es la gramática correspondiente a una cadena S_i y $R_i \rightarrow a_1, ..., a_n$ son las reglas derivadas de la gramática

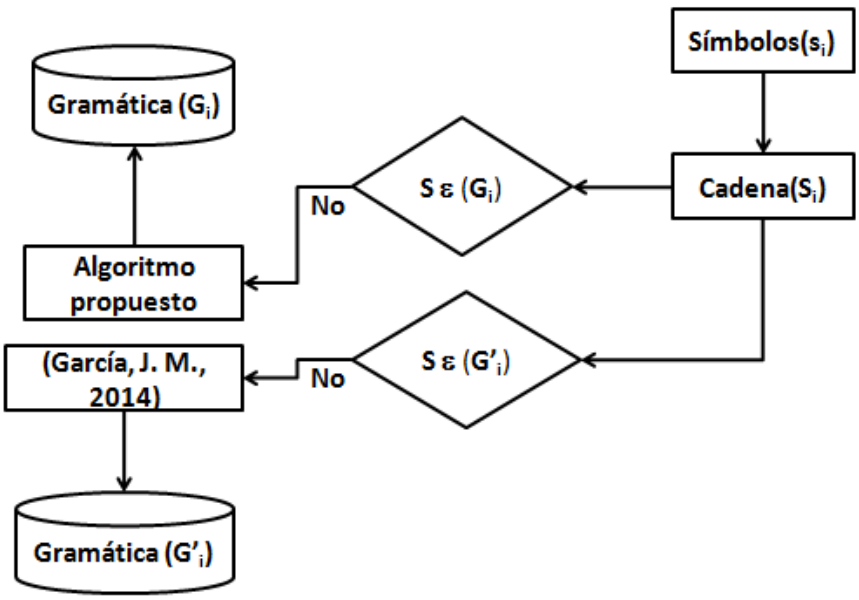

Figura 5.7: Diagrama de bloques de proceso de aprendizaje del modelo

iii) Comparar expresividad. La expresividad del modelo propuesto se determinara

comparando el numero de reglas necesarias para representar la dinámica del movimiento de los objetos en el escenario, por cada uno de los dos modelos a analizar. Las gramáticas correspondientes se extraen de los modelos, donde se checa una a una para validar la expresividad, donde lo que se observa son gramáticas mas compactas, lo cual valida la expresividad del nuevo modelo.

5.1.3. Análisis de resultados

Con la metodología desarrollada, se procede a validar el planteamiento de la hipótesis, la cual propone que se pueden obtener gramáticas mas compactas para representar la dinámica del movimiento de objetos lo cual se refleja en la expresividad de estas. Para esto, a continuación se muestran los resultados obtenidos en los dos escenarios de prueba:

i) Escenario de prueba 1. En el primer escenario se obtuvieron 3345 cadenas en un tiempo de 8.33 horas de video, para la generación del modelo gramatical se aplico el criterio establecido para el tiempo de aprendizaje con un valor de k=1, se encontró que el tiempo de aprendizaje que mejor se aproxima es de 43 minutos, con el cual se aprendieron 190 gramáticas de las 420 totales que se generarían si se analizaran todas las cadenas, lo cual equivale a un 45 % del total.

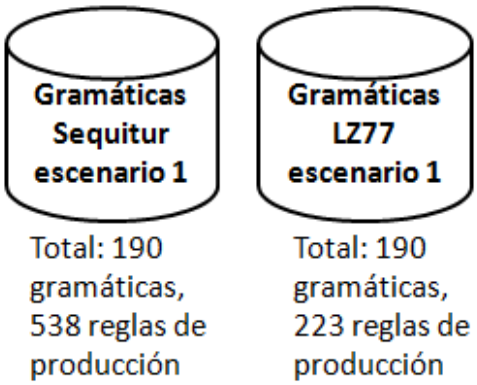

Figura 5.8: Comparativa gramáticas reglas en escenario 1.

Las reglas generadas por sequitur son 538 reglas para 190 gramáticas, LZ77 generó 223 reglas para las mismas gramaticas, con esto se observa que para el escenario 1 el nuevo modelo genera reglas mas expresivas.

2) Escenario de prueba 2. En este escenario se obtuvieron 3874 cadenas en un tiempo de 2.04 horas de video, para la generación del modelo gramatical, se encontró que el tiempo de aprendizaje que mejor se ajusta es de 71 minutos, con este tiempo de aprendieron 2234 gramáticas, lo cual corresponde a un 57 % del total de gramáticas.

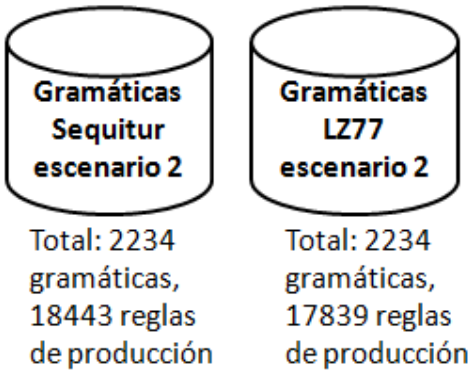

Figura 5.9: Comparativa gramáticas reglas en escenario 2

Se observa en los dos escenarios que el numero de gramáticas que se generan para cada uno de ellos tiene reglas más expresivas por lo cual es prudente hacer esta afirmación.

Capítulo 6

Conclusiones.

Como conclusiones se tiene que a travéz de los resultados obtenidos en el capitulo 5, se valida la hipótesis planteada, ya que se demuestra que se pueden obtener gramáticas mas expresivas para representar trayectorias, esta demostracion se hace comparando las reglas necesarias para representar las gramaticas generadas para un escenario por [1], contra las reglas necesarias para representar las gramaticas generadas por el metodo planteado en el capitulo 5, la evidencia de los resultados obtenidos muestran que con el metodo propuesto se reduce considerablemente el número de reglas necesarias para la representación gramatical de las actividades de los objetos en un escenario.

También se logra obtener un método automático para determinar el tiempo de entrenamiento del sistema basado en la evidencia visual del escenario y la dinámica de los objetos, con esto el tiempo del proceso de inferencia gramatical o fase de aprendizaje del modelo se mejora, todo esto basado en las información que arroja la pendiente de la recta tangente, la cual aporta evidencia para definir cuando el sistema ha aprendido lo suficiente o esta a punto de converger, este criterio se toma comparando un valor k con la velocidad decrecimiento de una curva c donde el creterio para establecer cuando el sistema ha aprendido lo suficiente se establece mediante la constante antes mencionada k.

Bibliografía

[1] J. M. Garcia, *Generacion automatica de modelos gramaticales para la inferencia de actividad en un sistema de vision*. PhD thesis, CIDESI, 2014.

[2] M. Daldoss, N. Piotto, N. Conci, and F. G. B. De Natale, "Learning and matching human activities using regular expressions," *Proceedings - International Conference on Image Processing, ICIP*, no. OCTOBER, pp. 4681–4684, 2010.

[3] H. Jiménez-Hernández, J. J. González-Barbosa, and T. Garcia-Ramírez, "Detecting abnormal vehicular dynamics at intersections based on an unsupervised learning approach and a stochastic model," *Sensors*, vol. 10, no. 8, pp. 7576–7601, 2010.

[4] F. G. D. N. Andrea Rosani, Nicola Conci, "Human Behavior Recognition Using Using a Context-Free Grammar," *JOURNAL OF ELECTRONIC IMAGING - Article Number: 033016*, vol. 23, no. 3, pp. 0–40, 2014.

[5] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, 2008.

[6] C. G. Nevill-manning, *Infeering Secuential Structure*. PhD thesis, University of Waikato, 1996.

[7] j. G. Brookshear, "TEORÍA DE LA COMPUTACIÓN Lenguajes formales, automatas y complejidad," *ADDISON-WESLEY IBEROAMERICANA*.

[8] M. Charikar, E. Lehman, D. Liu, R. Panigrahy, M. Prabhakaran, and A. Sahai, "The Smallest Grammar Problem," vol. 51, no. 7, pp. 2554–2576, 2005.

[9] C. G. Nevill-Manning and I. H. Witten, "Identifying Hierarchical Structure in Sequences: A linear-time algorithm," *arXiv preprint cs/9709102*, vol. 7, pp. 67–82, 1997.

[10] M. Mernik, D. Hrncic, B. Bryant, A. Sprague, J. Gray, Q. L. Q. Liu, and F. Javed, "Grammar inference algorithms and applications in software engineering," *2009 XXII International Symposium on Information, Communication and Automation Technologies*, 2009.

[11] I. Transactions and O. N. Information, "On the Complexity of Finite Sequences," vol. I, no. 1, 1976.

[12] R. Parekh and V. Honavar, "Grammar inference, automata induction, and language acquisition," *Handbook of natural language processing*, pp. 1–60, 2000.

[13] G. Alvarez, J. Ruiz, and P. García, "Comparison of two recent algorithms for grammatical inference of regular languages by means of non-deterministic automata," *Ingeniería y Competitividad*, vol. 11, no. 1, pp. 21–36, 2009.

[14] A. D'Ulizia, F. Ferri, and P. Grifoni, "A survey of grammatical inference methods for natural language learning," *Artificial Intelligence Review*, vol. 36, no. 1, pp. 1–27, 2011.

[15] I. I. Electronics and V. C. Workshop, "School of Engineering , Computing and Science , Swinburne University of Technology (Sarawak Campus), Malaysia," 2012.

[16] C. Nevill-Manning and I. Witten, "Compression and explanation using hierarchical grammars," *The Computer Journal*, vol. 40, no. 2 and 3, pp. 103–116, 1997.

[17] J. Salas, H. Jim??nez, J. Gonz??lez, and J. Hurtado, "Detecting unusual activities at vehicular intersections," *Proceedings - IEEE International Conference on Robotics and Automation*, no. April, pp. 864–869, 2007.

[18] H. Hernandez-ramirez, J. M. Garcia-huerta, T. Hernandez-diaz, E. Reyes-santos, and H. Jimenez, "Dynamic Modeling by Grammatical Inference : The learning Convergence,"

[19] D. Hrnčič, M. Mernik, and B. R. Bryant, "Improving grammar inference by a memetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 5, pp. 692–703, 2012.

[20] J. Sha, Y. Zhao, W. Xu, H. Zhao, J. Cui, and H. Zha, "Trajectory analysis of moving objects at intersection based on laser-data," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 289–294, 2011.

[21] N. Many, "A Universal Algorithm for Sequential Data Compression," 1977.

[22] C. de la Higuera, "Current Trends in Grammatical Inference," pp. 28–31, 2000.

[23] T. Goan, N. Benson, and O. Etzioni, "A Grammar Inference Algorithm for the World Wide Web," *AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[24] N. La Serna Palomino and U. Román Concha, "Técnicas de Segmentación en Procesamiento Digital de Imágenes," *Revista de Ingeniería de Sistemas e Informática*, vol. 6, no. 2, pp. 9–16, 2009.

[25] E. Purdy, "Grammatical Methods in Computer Vision," *Vasa*, no. March, pp. 1–23, 2013.

[26] A. Clark, "Grammatical Inference and First Language Acquisition," *COLING 2004 Psycho-Computational Models of Human Language Acquisition*, pp. 27–34, 2004.

[27] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell, "A Novel Sequence Representation for Unsupervised Analysis of Human Activities," no. August 2008.

[28] S. Miller, "The Method of Least Squares and Signal Analysis," pp. 1–7, 1992.

[29] K. Kim, D. Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," no. 1, pp. 1164–1171, 2011.

[30] L. Miclet, *Grammatical Inference*, vol. 7. 1990.

[31] I. JACOB ZIV, FELLOW, IEEE, AND ABRAHAM LEMPEL, MEMBER, "Compression of lndiwdual Sequences via Variable-Rate Coding," vol. IT-24, no. x, pp. 530–536, 1978.

[32] A. Yilmaz, O. Javed, and M. Shah, "Object tracking," *ACM Computing Surveys*, vol. 38, no. 4, pp. 13–es, 2006.

[33] I. Ziv, Jacob, Fellow, "Coding Theorems for Individual Sequences," *IEEE Transactions on information theory*, vol. IT-24, no. 4, 1978.

[34] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, 2008.