Universidad Nacional Autónoma de México

Posgrado en Ciencia e Ingeniería de la Computación

Seguimiento de objetos con el uso de características locales invariantes a escala y rotación

T E S I S

que para obtener el grado de:

Maestro en Ingeniería (Computación)

P R E S E N T A

Gerardo Carrera Mendoza

Director: Dr. Jesús Savage Carmona

Co-director: Dr. Walterio Mayol-Cuevas

Ciudad Universitaria, México D.F., Marzo de 2007

Agradecimientos

A Dios por darme la maravillosa oportunidad de vivir y acompañarme en todo momento.

A mis padres por su amor infinito y entre muchas cosas, por su apoyo incondicional.

A mi hermano por ser un gran ejemplo, un gran compañero y sobre todo un gran amigo.

A Nadyn por su comprensión y apoyo en esta parte de mi vida.

Al Dr. Savage por su ayuda, por su apoyo y por fomentar en mí el camino de la investigación.

Al Dr. Walterio Mayol-Cuevas por su paciencia, por sus valiosos consejos y por ser además de un excelente supervisor durante mi estancia en la Universidad de Bristol, G. B., un gran ejemplo seguir.

Al apoyo económico que me brindo CONACYT y DGEP para la realización de estos estudios de posgrado.

Al Posgrado en Ciencia e Ingeniería en Computación por el apoyo dado para la asistencia a diversas actividades que me ayudaron en una formación más integral.

Al los amigos del laboratorio de bio-robótica de la UNAM: David Esparza, Javier Jimenez, Sergio Cuellar, David Cortes, Javier Rodriguez, Rafael Sobrevilla, Adalberto Hernandez.

A la UNAM.

Índice general

Índice de figuras

Resumen

En esta tesis se presenta la implementación de un sistema seguidor de objetos robusto a oclusiones parciales, rotación y escala para una variedad de diferentes objetos permitiendo movimientos normales de éstos y de la cámara. Para lograrlo, el objeto es representado por puntos característicos o de interés, los cuáles son descritos en múltiples resoluciones, obteniendo así una representación del punto en diferentes escalas, añadiendo un cómputo más rápido debido a que se hace fuera de línea.

Los puntos de interés son extraídos con el detector de Harris [13], y descritos usando un algoritmo llamado SURF (Speeded Up Robust Features) [25]. Para seguir al objeto se utiliza un filtro de Kalman conocido como UKF(Unscented Kalman Filter) estimando la posición y la escala del objeto, con el uso de ésta aproximación se muestran buenos resultados con respecto a la velocidad del algoritmo, detección del objeto, seguimiento y recuperación del sistema aun cuando existen oclusiones totales del objeto.

Las aportaciones principales de la tesis son las siguientes: 1) el uso de puntos de interés extraídos con el detector de Harris y descritos con el algoritmo de SURF, 2) el modo en el que se logra la invariancia a escala, para lo cuál, los puntos que representan al objeto durante el primer cuadro (frame) son descritos a múltiples escalas y 3) el uso del UKF para predecir la escala y la posición del objeto, logrando así reducir el costo computacional del algoritmo

Capítulo 1

Introducción

Esta tesis se enfoca al área de visión por computadora, en donde el principal objetivo es extraer información relevante de las imágenes. Las imágenes digitales son representadas por pixeles, ordenados en un m-vector, donde usualmente m= 1, 2 o 3 (cabe destacar que en esta tesis se utilizará una representación en matrices de dos dimensiones). Los pixeles tienen diferente color e intensidad y dependiendo de la resolución, éstos son imperceptibles al ojo humano y se tiene la percepción de la imagen como un todo. Para una computadora los pixeles siguen siendo números sin sentido, el problema es hacer que éstos adquieran un significado.

Dentro de la visión por computadora se encuentran muchas áreas como son, por ejemplo: reconocimiento de objetos, modelado de objetos, procesos de control, detección de eventos y seguimiento de objetos en una secuencia de imágenes. En este análisis se desarrolla un sistema seguidor de objetos con la capacidad de representarlos sin tener un modelo preciso de su forma, además la representación es invariante a algunos cambios geométricos que sufre el objeto a lo largo del seguimiento.

Desde el punto de vista biológico, si se observa el sistema visual de los seres humanos es posible darse cuenta de que exhibe una impresionante robustez a cambios complejos en los eventos visuales, uno de estos eventos es el seguimiento de objetos, es decir, no perder de vista el objeto, algún animal o algo del interés en una persona en esos momentos. Es probable que no se reconozca el objeto, pero se extraen todas las características necesarias como para no perderlo de vista, inclusive con diferentes transformaciones como es el tamaño, rotaciones e iluminación. Biológicamente se da por hecho pero computacionalmente es una tarea que requiere de gran atención.

Si se transporta la idea del párrafo anterior a un sistema de visión por computadora entonces se tiene que dotar al sistema con la habilidad de manejar diversas situaciones, además, aun cuando la velocidad de los procesadores estándar aumenta cada vez mas, es necesario mantener la complejidad computacional lo mas baja posible para obtener un sistema que pueda trabajar rápidamente, procurando llegar a un desempeño en tiempo real.

El seguimiento de objetos es muy importante en el área de visión por computadora ya que surgen diferentes aplicaciones como son: seguridad y vigilancia, manejo de

tráfico de automóviles, realidad aumentada, robótica y manejo autónomo de vehículos. Por ejemplo:

En [20] se desarrolla un sistema para catalogar objetos, en donde el proceso es el siguiente: primero un paquete es puesto en un dispositivo clasificador mediante un mecanismo transportador; al término de éste, los paquetes son llevados manualmente a una caja en una banda transportadora. El sistema propone el uso de robots que recojan los paquetes y los lleven hacia las cajas, para hacer esto el robot tiene que conocer la posición, orientación y la región de agarre del objeto. Una cámara detecta las esquinas del paquete comparándolas con las de un modelo (*template*) y un filtro de Kalman se utiliza para estimar la posición y la orientación del paquete.

Una de las características que se pueden extraer de un objeto es el color, en [3] se segmenta la imagen obteniendo regiones identificadas por el color. Para representar al objeto, se construyen gráficas de adyacencia usando relaciones entre las posiciones de cada región.

Con el crecimiento acelerado de nuevas tecnologías, surge la necesidad de una interacción más real entre seres humanos y máquinas, en [37] un robot detecta y hace seguimiento de caras humanas alrededor de su vecindad y después interpreta los gestos naturales de la persona a seguir. En este trabajo [37] se hace uso del color y la forma de la cara, las cuales son incorporadas a un filtro de partículas para posteriormente realizar el seguimiento.

Como se puede observar, existen diferentes aproximaciones con las que se puede optar para resolver la tarea enfocada en la tesis, pero en un seguidor visual típico es posible distinguir principalmente dos componentes: 1) Representación y Localización del objeto, y 2)Filtrado y asociación de los datos [16]. Desde el punto de vista de la representación del objeto, se pueden distinguir aproximaciones en las que se usa una mínima cantidad de información extraída del objeto, como es el color [10][46], intensidad [36], puntos característicos [47], histogramas especializados de color [15], etc. Otra idea podría ser la integración de múltiples características para tener una mejor representación del objeto [21].

También están las aproximaciones que usan un modelo específico del objeto; la idea es útil por ejemplo, cuando se requiere seguir objetos sólidos en los cuales el modelo esta basado principalmente en los bordes o en una representación más detallada usando una parametrización como B-splines [53]. Si se analiza la idea anterior, es necesario tener una representación más detallada de la forma geométrica del objeto con los contornos perfectamente definidos, dadas las características anteriores, se pueden encontrar aplicaciones en las que este método ha dado muy buenos resultados [7][6]. Todas las diferentes aproximaciones que existen son buenas en términos de la aplicación y los resultados que se desean obtener.

En el segundo componente del seguidor, la idea principal es estimar el siguiente estado del objeto y para hacerlo, es necesario el uso de un estimador, comúnmente llamado "filtro", el cual en base a ciertas mediciones o estados anteriores puede predecir o estimar el siguiente estado. En la literatura se pueden encontrar diferentes

filtros usados en problemas de seguimiento, bajo ciertas circunstancias y cuando el problema es lineal, se puede asumir que la solución óptima es dada por el Filtro de Kalman (KF). Cuando el problema es no-lineal como en un seguidor de objetos visual entonces una solución es obtenida con el Filtro Extendido de Kalman (EKF), en la literatura el uso del EKF en seguidores visuales ha sido muy usado, se pueden encontrar aplicaciones del filtro anterior en [5][27][17][55]. Aunque el uso del EKF provee de una solución para problemas no-lineales, también conlleva desventajas como es la linearización y las dificultades en su implementación. Una alternativa más reciente es el Unscented Kalman Filter (UKF) [58] el cual presenta distintas ventajas en comparación con el EKF como es la no-linearización. En el capítulo 4 se describen mas ampliamente los filtros mencionados.

El trabajo esta distribuido de la siguiente manera: primero se da una breve reseña de antecedentes que podrán ser útiles al lector para una mejor comprensión de los capítulos siguientes. El capítulo 3 muestra un acercamiento al estado del arte con respecto a las diferentes técnicas utilizadas para la extracción y descripción de puntos de interés. En el capítulo 4 se revisa primeramente el filtro de Kalman y después se describe detalladamente del sistema desarrollado. En el siguiente capítulo se prueba el desempeño del sistema realizando experimentos que incluyen cambios de escala, rotación y oclusiones parciales. Por último, en el capítulo 6 se muestran las conclusiones y los trabajos futuros.

El objetivo de la tesis es presentar la implementación de un sistema seguidor de objetos robusto a oclusiones parciales, rotación y escala para una variedad de diferentes objetos permitiendo movimientos normales de éstos y de la cámara. Para lograrlo, el objeto es representado por puntos característicos o de interés, los cuales son descritos en múltiples escalas, obteniendo así una representación del punto en diferentes escalas lo que añade un cómputo más rápido debido a que se hace fuera de línea. Para seguir al objeto se utiliza el UKF para estimar la posición y la escala del objeto, con el uso del marco propuesto, se muestran buenos resultados con respecto a la velocidad del algoritmo, detección del objeto, seguimiento y recuperación del sistema aun cuando existan oclusiones totales del objeto.

Capítulo 2

Antecedentes

2.1. Características locales

El uso de características locales para el análisis de imágenes se ha convertido en los años recientes en una de las aproximaciones dominantes en varias disciplinas de investigación dentro del área más general llamada "visión por computadora", debido a que no es solamente un método para seleccionar características importantes de la imagen sino que es una nueva representación de la imagen que permite describir los objetos o parte de ellos sin la necesidad de segmentar.

Algunas de las áreas en donde se incluye el uso de características locales, son por ejemplo: reconocimiento de objetos, extracción de imágenes dentro de una base de datos ("image retrieval"), correspondencia entre imágenes ("image matching"), localización y construcción simultanea de mapas ("Simultaneous Localization And Mapping - SLAM"), y seguimiento de objetos("object tracking").

- *Image Matching*

  En el emparejamiento de imágenes o "image matching", la idea principal es comparar dos imágenes y encontrar similitudes punto a punto para después decidir si pertenecen o no a la misma clase, es decir, poder determinar si es la misma escena o el mismo objeto. Se incluyen cambios significativos en las condiciones de la imagen, por ejemplo grandes cambios en la escala (tamaño del objeto), rotación o en el ángulo de visión.

- *Reconocimiento de Objetos*

  En nuestra vida cotidiana, todo el tiempo estamos reconociendo objetos y lo hacemos sin esfuerzo a pesar de que los objetos puedan cambiar de tamaño, color o forma; podemos reconocerlos desde diferentes ángulos, lejos, cerca, parcialmente ocluidos y sin embargo podemos determinar el objeto que estamos viendo. Tratar de implementar sistemas con el uso de una computadora y una cámara que sean capaces de reconocer objetos bajo diversas situaciones adversas, resultaría en una cantidad muy grande de aplicaciones; es decir, por ejemplo, la ayuda a personas con alguna discapacidad, bastaría con imaginarse

a las personas que no pueden ver y que podrían ser ayudados con un sistema de reconocimiento de objetos para saber que está alrededor de ella; otra aplicación directa sería a la robótica.

La idea principal es pues, que a partir de una imagen o una secuencia de imágenes, se reconozcan los objetos que se encuentran dentro de ella, para lograrlo se debe tener una representación del objeto(s) lo suficientemente robusta como para encontrar al objeto idóneo.

- *Image Retrieval*

  La extracción de imágenes dentro de una base de datos o "image retrieval" es un área muy reciente que se ha desarrollado rápidamente. Con el uso de internet y los motores de búsqueda se pueden localizar imágenes dentro de grandes bases de datos solamente tecleando alguna palabra que se relacione con la imagen que se quiera encontrar. Esta área todavía no ha tenido un gran éxito comercial debido a que existen muchas variantes que deben ser investigadas tal y como el uso de información semántica de la escena o del objeto que se quiere extraer, luego entonces, esta área requiere algoritmos robustos de reconocimiento de objetos y de matching.

- *SLAM*

  Es una área que ha sido de gran interés y de ardua investigación principalmente en robótica. El problema a atacar es determinar la localización espacial del robot y la construcción al mismo tiempo de un modelo abstracto del ambiente. Para poder realizar este modelo o mapa del ambiente, es necesario extraer información relevante del entorno, es por ello que el uso de sensores es necesario, principalmente: sonares, lasers o visión por computadora.

  Como se puede concluir del párrafo anterior, el mapa debe de ser siempre el mismo a pesar de encontrarse en otra posición, es decir, si se usa visión por computadora, las estructuras o características que se usen para hacer SLAM deben ser suficientemente distintivas como para que en base a estas características se pueda tener una localización espacial con gran certidumbre.

- *Object Tracking*

  Área que acapara un gran interés debido a que se trabaja no sólo con imágenes estáticas sino con secuencias de imágenes con las cuáles se extrae información acerca de los cambios que ha sufrido la imagen en cada cuadro (*frame*). Se puede asumir que existen básicamente dos aproximaciones generales: i) asumir que el movimiento del objeto se puede entender mejor como el análisis rápido de muchos cuadros estáticos relacionados entre sí, encontrando las diferencias o similitudes entre cuadros sucesivos, por ejemplo: Optical Flow, ii) analizar cada cuadro independientemente de la relación que tenga con los cuadros anteriores, por ejemplo: Segmentación. Una vez que se tiene una representación del objeto a seguir entonces el siguiente paso es estimar la posición siguiente.

Se puede definir el seguimiento de objetos usando visión por computadora como la identificación sucesiva de un objeto a seguir dentro de una imagen con la respectiva estimación de la posición donde se va a mover el objeto en el siguiente cuadro.

2.1.1. Detección de características locales

En las áreas de investigación mencionadas anteriormente, el primer paso es la extracción de información o características dentro de la imagen. De las diferentes aproximaciones que existen para la extracción de la información, la mayoría de ellas se basan principalmente en un análisis global de la imagen por lo que es necesario segmentarla. La idea principal de la segmentación es distinguir los objetos del fondo de la imagen haciendo uso de características como el color o la textura, desafortunadamente estas aproximaciones no son lo suficientemente robustas a oclusiones, iluminación o cambios introducidos arbitrariamente por las condiciones del ambiente o de la imagen.

Existen también aquellas que hacen uso de las características geométricas del objeto, la desventaja principal radica en el uso de modelos, lo que nos lleva a la dificultad de representar objetos en donde la geometría no es fácil de representar. Por lo tanto es crucial tener algoritmos robustos para la detección de características que sean lo suficientemente descriptivas y que resuelvan la mayoría de los problemas inducidos por cambios complejos en la imagen.

Una característica local es una estructura local de la imagen formada por pixeles con alta variación en su intensidad. Los puntos en donde el contenido local cambia en 2 direcciones (vertical y horizontal) son llamados puntos de interés [34]. Estos puntos contienen más información debido a estos cambios y por lo tanto son más representativos para la imagen. Un punto interesante esta representado por una vecindad pequeña y esta definido por las coordenadas de la imagen, el tamaño y la forma de la estructura. Por lo tanto la obtención de puntos de interés requiere de 2 fases, la primera es llamada "detección" y la segunda "descripción" (ver figura 2.1).

El uso de características locales en el contexto del reconocimiento provee de un gran progreso en la solución de los problemas mencionados anteriormente como lo son cambios de escala, rotación, oclusiones parciales y cambios no muy bruscos en la iluminación, por lo tanto muestra un gran avance en términos de robustez, eficiencia y calidad de los resultados.

El uso de puntos de interés para el reconocimiento de objetos y la extracción de imágenes fue comenzado por Cordelia Schmid y Roger Mohr en el año de 1999 [11], en donde se hace uso de características locales en escala de grises calculadas en puntos de interés para una aplicación de "Image retrieval" en una base de datos de 1020 imágenes, mostrando robustez a rotaciones, cambios en la escala y oclusiones parciales, obteniendo una tasa de reconocimiento del 99 % para una variedad de imágenes de prueba tomadas bajo diferentes condiciones visuales.

a) b)

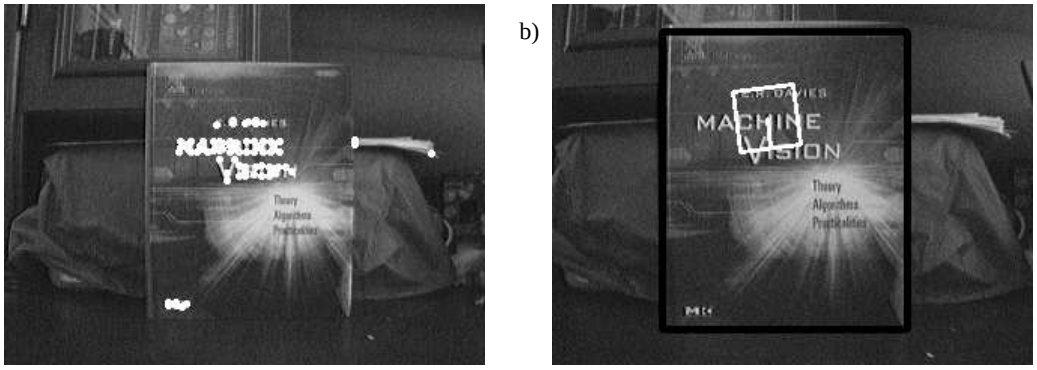

Figura 2.1: El inciso a) representa la detección de puntos usando el detector de Harris, el inciso d) muestra la detección de un punto y la ventana es la vecindad en donde se construye el descriptor

La primera fase para la obtención de puntos de interés, se encarga de detectar todos aquellos puntos de interés que representan la imagen, por lo tanto, el detector ideal debe de proveer de una localización espacial de los puntos más sobresalientes o representativos dentro de una imagen lo suficientemente precisa como para que los puntos en el vecindario correspondan en su mayoría, aún y cuando la imagen haya sido transformada. Una ventaja del uso de detectores de puntos de interés es que reducen la cantidad de datos a procesar.

La segunda se encarga de describirlos en términos de una vecindad formada alrededor del punto de interés, esta descripción es necesaria para después comparar y encontrar estructuras similares en imágenes transformadas, el problema radica en crear una descripción lo suficientemente robusta, compacta y fácil de manipular, para no afectar el costo computacional del sistema.

Las propiedades que debe de cumplir una característica ideal dentro de una imagen son las siguientes [65]:

- *Local:* Sólo se analiza una estructura o una vecindad alrededor del punto interesante.
- *Invariante:* Cuando una transformación se aplica a la imagen, la medida de la característica sigue siendo la misma.
- *Robusta:* En la presencia de factores como lo son: ruido, discretización o compresión; no tienen gran influencia en la característica.
- *Distintiva:* Cada característica debe ser lo suficientemente distinta a las demás como para poder ser localizada en imágenes diferentes.
- *Cantidad:* Muchas características pueden ser generadas sin importar el tamaño del objeto.
- *Precisión:* La localización espacial de los puntos de interés debe de ser muy exacta.

- *Eficiencia:* El costo computacional debe de ser muy cercano al tiempo real (30 cps).

2.2. Transformaciones geométricas

En una imagen los objetos parecen diferentes si son apreciados desde diversas posiciones, lo que sugiere que sufren de distintas transformaciones, es decir, en una transformación se realiza un mapeo en $2D$ del espacio de la imagen fuente a un espacio destino, relacionando un punto fuente (x', y') hacia un punto destino (x, y) de acuerdo a las funciones $x(x', y')$ y $y(x', y')$.

A partir de este momento se usará la notación homogénea de puntos en el espacio ya que provee de una notación consistente para transformaciones euclidianas, afines y proyectivas. Para efectos de esta tesis, esta notación es una representación de los puntos en el plano real R^2, es decir, $(x, y)^T$ a un plano proyectado, superconjunto del plano real cuyas coordenadas homogéneas son $(u, v, w)^T$. Por lo que un punto en $2D$ dentro de la geometría euclidiana $(x, y)^T$ es representado por vectores homogéneos de la forma $p = (u, v, w)^T = (xw, yw, w)^T$, donde w es un número arbitrario diferente de cero. Para recuperar las coordenadas a partir de un vector homogéneo, simplemente se divide entre la componente homogénea, es decir, $(x, y)^T = (u/w, v/w)^T$. Por conveniencia para representar puntos en notación homogénea generalmente se usa $w = 1$, [12].

Dentro de las transformaciones geométricas mas comunes, se encuentran las Afines y las Proyectivas. A continuación se dará una breve descripción de estas[9]:

- *Afines:* Este tipo de transformaciones es un subconjunto de las Proyectivas en donde las líneas paralelas, los planos y los puntos equiespaciados sobre una línea se preservan. Dentro de este tipo de transformaciones, se encuentran las siguientes clasificaciones: translación , rotación, reflexión, escala e inclinaciones (*shearing*) (ver figura 2.2). Cabe destacar que en la literatura existe la distinción de un subconjunto llamado transformaciones euclidianas que incluyen rotación, translación y reflexión. Esta distinción se basa en la preservación de la longitud, los ángulos y la forma de los objetos (en las euclidianas se preservan).

  Formalmente una transformación afin es una transformación lineal mas una translación. Una transformación $T(x)$ es lineal si $T(x + y) = T(x) + T(y)$ y $T(\alpha x) = \alpha T(x)$ para cualquier escalar α. Es afín si existe una constante c y una transformación lineal $L(x)$ tal que $T(x) = L(x) + c, \forall x$ en donde $L(x)$ es una transformación lineal. En general una transformación afín puede ser expresada de la siguiente manera [50]:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

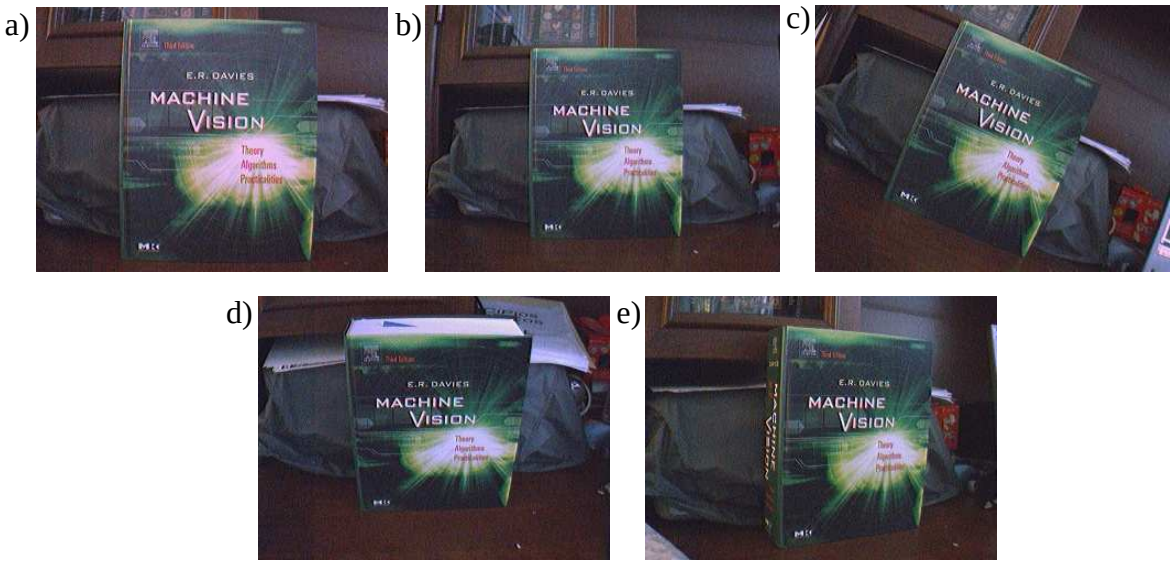

Figura 2.2: Transformaciones Geométricas: a) Imagen inicial, b) Escala, c) Rotación, d) y e) Inclinación (*Shearing*)

Se puede observar que la matriz de 3×3 tiene 6 grados de libertad. Geométricamente, los vectores (a, d) y (b, e) son vectores base del espacio destino, y (c, f) es el origen.

A continuación se ejemplifican 2 transformaciones afines,

* *Translación:* Una operación de translación es un mapeo de la posición de los pixeles de un objeto dentro de la imagen de entrada a otra posición en la imagen de salida, en donde generalmente pero no siempre, la imagen de entrada y de salida tienen la misma dimensión. Si se refiere a un espacio de 2 dimensiones, entonces la idea es trasladar un punto en el plano-xy hacia un nuevo lugar añadiéndole un vector (h, k) (ver figura 2.3). Es decir,

$$\begin{aligned} x' &= x + h, \\ y' &= y + k \end{aligned} \tag{2.2}$$

En coordenadas homogéneas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.3}$$

* *Rotación:*Una operación de rotación es un mapeo de la posición (x, y) de un pixel o de un objeto de la imagen a otra posición (x', y') rotándolo un ángulo θ. Las coordenadas del centro de rotación estan dadas por (x_0, y_0) (ver figura 2.4). Es decir,

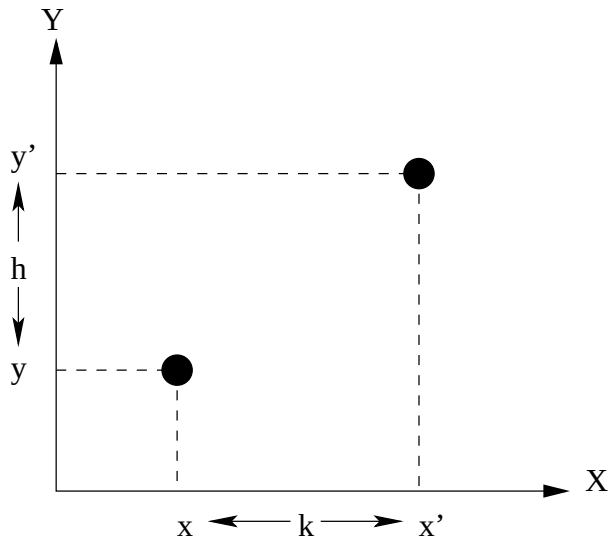

Figura 2.3: Transformaciones Geométricas: Translación

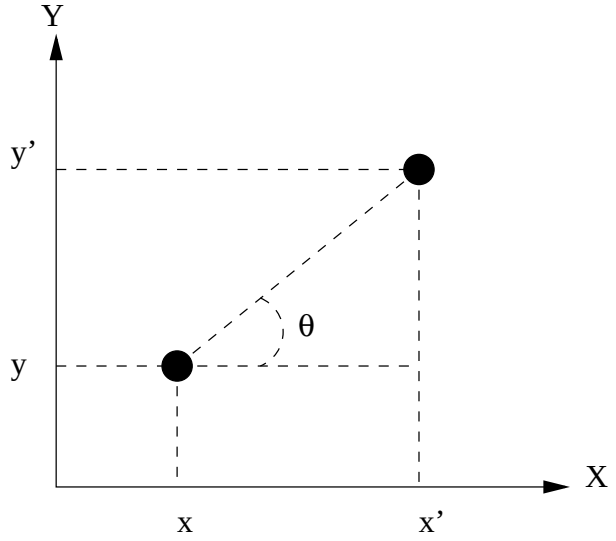

Figura 2.4: Transformaciones Geométricas: Rotación

$$\begin{aligned} x' &= (x - x_0)cos\theta - (y - y_0)sin\theta + x_0, \\ y' &= (x - x_0)sin\theta + (y - y_0)cos\theta + y_0 \end{aligned} \tag{2.4}$$

En coordenadas homogéneas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} (x - x_0) \\ (y - y_0) \\ 1 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{2.5}$$

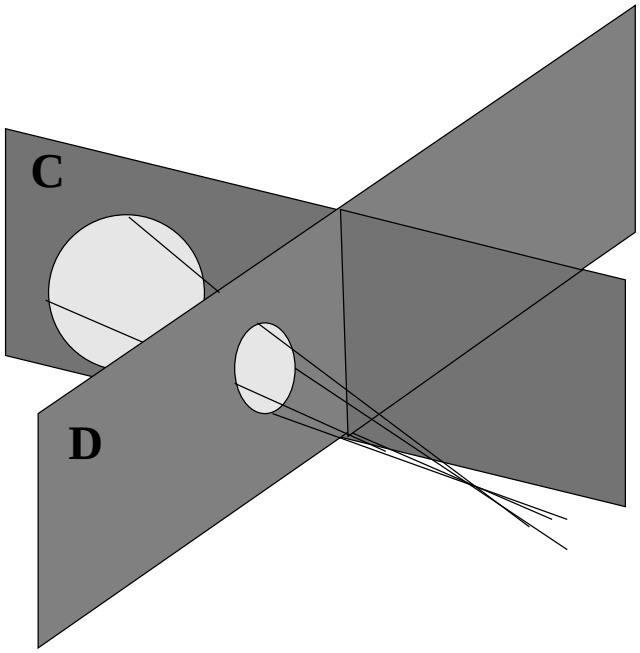

Figura 2.5: Transformación proyectiva de un objeto en el plano C al plano D. Las líneas representan 4 puntos necesarios para la obtención de la matriz de transformación [41]

- *Proyectivas:* Es la proyección de un plano "C" hacia un punto en otro plano "D", ver figura 2.5. En este tipo de transformaciones las líneas se preservan pero no necesariamente el paralelismo. Este tipo de transformaciones es el conjunto más general de transformaciones lineales.

  La forma general de una transformación proyectiva es la siguiente:

$$x' = \frac{ax + by + c}{gx + hy + i}, \quad y' = \frac{dx + ey + f}{gx + hy + i} \tag{2.6}$$

  En coordenadas homogéneas:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.7}$$

Capítulo 3

Estado del Arte

En este capítulo se mostrarán diferentes aproximaciones para la detección y descripción de puntos de interés. Existen muchos algoritmos para la detección de puntos de interés, éstos difieren principalmente en la información que el punto representa. Generalmente los algoritmos son evaluados o clasificados tomando en cuenta dos conceptos: i) distinguibilidad, es decir, que tan bien se pueden igualar los puntos detectados en distintas imágenes de la misma escena; y ii) repetibilidad, que es una comparación geométrica entre los puntos detectados en dos imágenes de la misma escena bajo diferentes transformaciones.

Como se mencionó en el capitulo anterior, el principal parámetro a estimar por el detector, es la posición espacial del punto interesante, si la intención es extraer puntos a diferentes escalas entonces el segundo parámetro importante es la escala del punto. La escala de un punto interesante se relaciona con la resolución a la cual el punto o la característica local es representada en la imagen. Existe un intervalo dado por la mínima y una máxima escala donde el punto de interés es significativo para la imagen. El problema está en encontrar la escala donde el punto se representa mejor dentro de la imagen.

Para poder detectar puntos característicos en diferentes escalas es necesario construir una representación de la imagen en espacio-escala (scale-space). La teoría de espacio-escala ha sido muy estudiada [8][32][61][38] y está demostrado que el kernel (filtro) óptimo para hacer una representación espacio-escala es la función Gaussiana (ver figura 3.1).

A continuación se comienza con una breve introducción en la detección de puntos de interés y después se describirán algunos de los algoritmos más robustos en la literatura en términos de repetibilidad y distinguibilidad.

3.1. Representación Espacio-Escala

Existen diferentes resultados que muestran que bajo transformaciones lineales el único kernel para generar el espacio-escala es el filtro Gaussiano [32][59][60][62][38]. Esta característica es determinada principalmente por las siguientes propiedades:

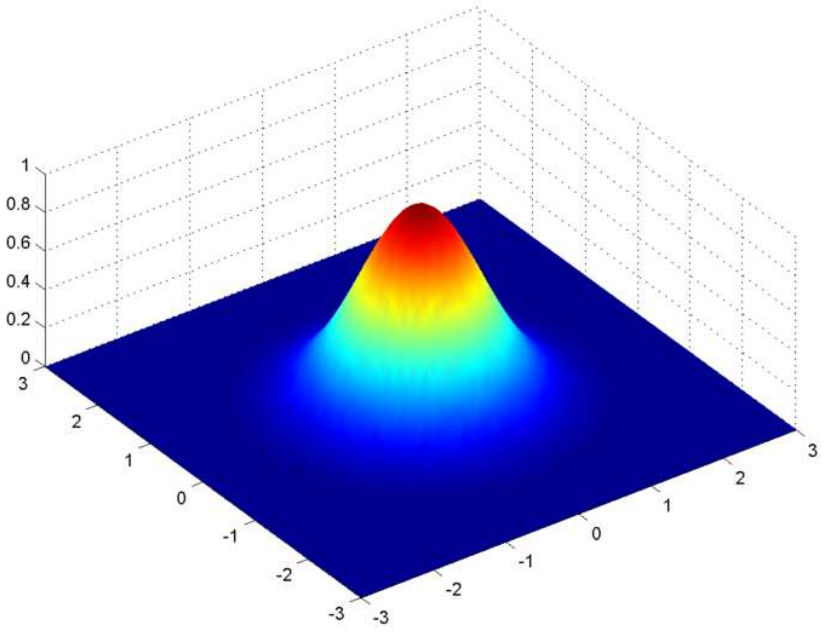

Figura 3.1: Función Gaussiana en 2D

- *Linealidad.*- Un kernel Gaussiano cumple con la propiedad de linealidad que puede ser expresada como sigue:

$$g_{(x,y,\sigma)}(\alpha f(x,y) + \beta h(x,y)) = \alpha g_{(x,y,\sigma)} f(x,y) + \beta g_{(x,y,\sigma)} h(x,y) \tag{3.1}$$

  donde, α y β son escalares, σ es la desviación estándar del kernel Gaussiano y:

$$\begin{aligned} g_{(x,y,\sigma)} f(x,y) &= g(\sigma,x,y) * f(x,y), \\ g_{(x,y,\sigma)} h(x,y) &= g(\sigma,x,y) * h(x,y) \end{aligned} \tag{3.2}$$

- *Separabilidad.*- Un kernel multi-dimensional Gaussiano puede ser obtenido del producto de kernels Gaussianos uni-dimensionales.

- *Causalidad.*- Ninguna estructura adicional debe de ser creada en la transformación de una escala fina a otra mas gruesa.

- *Asociatividad.*- La aplicación de un kernel sucesivamente n veces a una imagen es equivalente a aplicar una sola vez un kernel del tamaño igual a la suma de los n kernels.

$$g(\sigma_1) * I(x) + ... + g(\sigma_n) * I(x) = g(\sigma_1 + ... + \sigma_n) * I(x) \tag{3.3}$$

La representación de una imagen en espacio-escala es un conjunto de imágenes representadas en diferentes niveles discretos de resolución [8]. Esta, se define como una función $L(x, y, \sigma)$, que es resultado de la convolución de una función Gaussiana de escala variable $G(x, y, \sigma)$ con una imagen de entrada $I(x, y)$ [19]:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.4}$$

donde:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.5}$$

y,

(x, y) - posición del punto de interés.

Por lo tanto una imagen en una escala mas gruesa es obtenida suavizando una escala mas fina de la imagen mediante la convolución de un filtro Gaussiano. Esta operación se repite consecutivamente hasta construir la representación espacio-escala. Existen diferentes maneras de construir esta representación (ver figura 3.2):

i) Representación piramidal, en la cual después de suavizar la imagen en una escala mas fina, ésta es muestreada por el correspondiente factor de escala para obtener la imagen en una escala más gruesa que después será suavizada por el correspondiente factor de escala y así sucesivamente, acelerando el proceso ya que en cada escala la imagen se hace más pequeña y el espacio de búsqueda es menor, pero al mismo tiempo se tiene que realizar el cómputo que relaciona la posición de los puntos de interés en las diferentes escalas con la imagen original [49]. Una aplicación para detectar puntos de interés usando la representación piramidal se puede observar en la aproximación de SIFT (*Scale Invariant Feature Transform*), desarrollada por David Lowe [18]

ii) Otra manera de representar el espacio-escala es suavizar la imagen de mas alta resolución o mas fina, con kernels de diferentes escalas sin cambiar el tamaño de la imagen, esta idea se observa en el trabajo SURF (*Speeded Up Robust Features*), hecho por Herbert Bay, Tinne Tuytelaars y Luc Van Gool [25].

3.2. Operadores diferenciales

Generalmente con el uso de operadores diferenciales se puede extraer la información necesaria para el análisis local de señales, se puede observar en mediante la expansión de la serie de Taylor [64].

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + ... + \frac{f^n(x_0)}{n!}(x - x_0)^n \tag{3.6}$$

Para nuestro caso la señal en análisis es una imagen bidimensional y se asume que es una señal diferenciable. La expansión de la serie de Taylor truncada hasta el

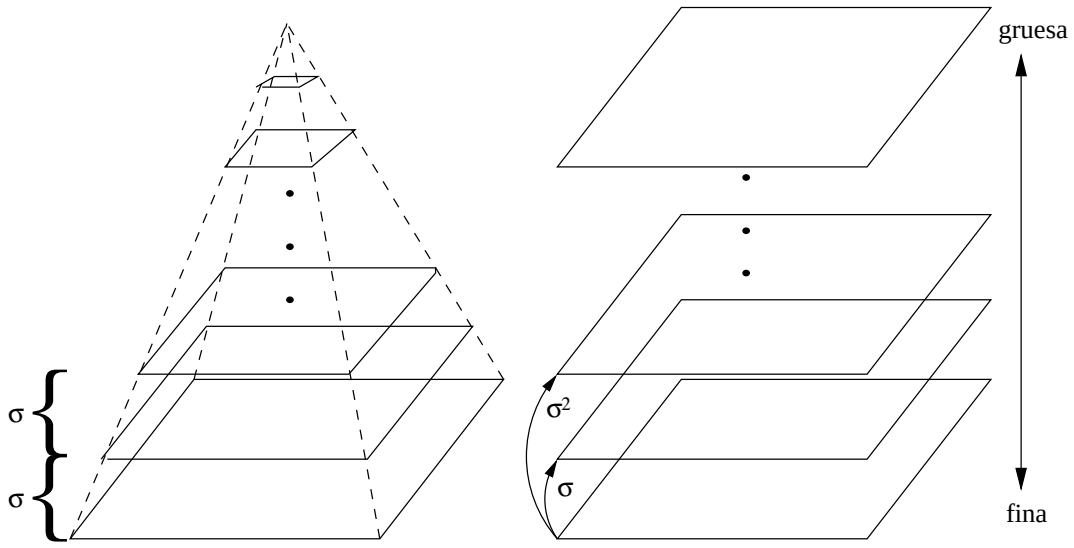

Figura 3.2: a) Representación piramidal, b) Representación construida con la sucesiva aplicación de kernels a la imagen original [34]

segundo orden, ha sido demostrado [32] que localmente aproxima la estructura de una imagen alrededor de un punto.

$$f(x) \approx f(x_0) + x\,\nabla f(x_0)^T + x^T \mathcal{H}(x_0)\,x \tag{3.7}$$

donde ∇ es el operador gradiente, el cuál es un vector que apunta en la dirección de máximo crecimiento (ver figura 3.3), y esta definido como:

$$\nabla f(x_1, x_2, ..., x_n) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n}) \tag{3.8}$$

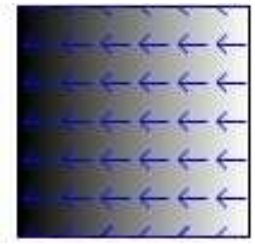

Figura 3.3: Gradiente: Dirección del máximo crecimiento

$(\mathcal{H})$ es la matriz Hessiana, definida por:

$$\mathcal{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{bmatrix} \tag{3.9}$$

Los operadores diferenciales anteriores generalmente son usados de diferentes maneras, el gradiente (ver fig 3.3) normalmente es usado para la detección de características locales o para la descripción de estructuras locales de la imagen [4][13][31]. Esta matriz describe la distribución del gradiente en una vecindad alrededor de un punto. Una manera de hacer un descriptor invariante a rotaciones es considerar las derivadas direccionales alrededor del punto de interés, con esto se obtiene la dirección dominante de la estructura local. En la siguiente sección se explicará detalladamente, dependiendo del algoritmo usado, la manera en hacer el descriptor invariante a rotaciones.

La matriz Hessiana ($\mathcal{H}$) puede ser usada también, tanto para detectar como para describir las propiedades de estructuras locales de una imagen. El uso de esta matriz en el contexto de detección puede ser observado en [52][39] y para la descripción de puntos de interés en [32][11]. Particularmente el uso de la traza y el determinante de esta matriz son de gran interés. La traza de la matriz denota el filtro Laplaciano, generalmente usado para la detección isotrópica (independiente de la dirección) de bordes.

El uso de segundas derivadas como es el caso de la matriz Hessiana nos da una respuesta pequeña exactamente en el punto donde el cambio en la señal es significativo, a diferencia de la primera derivada, en la cual el máximo no es localizado exactamente en el punto donde la señal cambia significativamente, sino en su vecindad [34].

3.3. Revisión de las principales aproximaciones en la detección y descripción de puntos de interés

En esta sección se presenta una revisión breve de algunas aproximaciones dentro de la literatura existente para la detección y descripción de puntos de interés. Estas aproximaciones se han distinguido por su buen desempeño respecto a su repetibilidad y distintividad de los puntos encontrados. Se revisarán a continuación los siguientes algoritmos:

- *Harris* [13]
- *D.Lowe: SIFT* [18]
- *Mikolacjzyk y Schmid: Hessian/Harris-Laplacian* [35]
- *H.Bay: SURF* [25]

3.3.1. Detector de puntos de interés de Harris

Este detector de puntos de interés fue propuesto por Chris Harris y Mike Stephens [13], y es de los detectores más usados debido a su repetibilidad e información

en el contenido. En la evaluación hecha en [12] se puede observar que bajo estos dos criterios, el detector de Harris tiene un mejor desempeño comparado con diferentes detectores existentes en la literatura [29][54][22][66].

Los antecedentes de este algoritmo provienen del detector hecho por Moravec [26], uno de los primeros detectores de puntos de interés basados en la intensidad de la señal. El detector de Moravec está basado en la función de auto-correlación de la señal. Esta función mide las diferencias entre los valores de gris de una ventana comparada con ventanas movidas en varias direcciones. Se consideran 3 casos [13]:

a) Si la región en donde está la ventana es plana (constante en intensidad) entonces todos los cambios al moverse la ventana son pequeños.

b) Si la ventana está sobre un borde, los cambios van a resultar pequeños cuando el movimiento es a lo largo del eje pero grandes cuando el movimiento es perpendicular al borde.

c) Si la ventana está sobre una esquina o un punto aislado entonces todos los movimientos producen cambios grandes.

Se puede observar en la función de auto-correlación, dado un cambio $(\Delta x, \Delta y)$ y un punto (x, y) la función está dada por [12]:

$$f(x,y) = \sum_{W} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2 \tag{3.10}$$

donde I es la imagen evaluada en (x_k, y_k) dentro de la ventana W centrada en (x, y), los movimientos de la ventana son discretos: $(1,0), (1,1), (0,1), (-1,1)$, los cuales representan un movimiento de la ventana de 45 grados (ver figura 3.4). El detector busca si el mínimo de la función, $min(f)$, es mayor que un umbral, si es así entonces existe un punto de interés.

El detector de puntos de interés de Harris mejora la aproximación hecha por Moravec debido a que ésta presenta diferentes problemas, los cuales son mencionados a continuación [13]:

- La respuesta es anisotrópica, es decir, dependiente de la dirección de búsqueda,
- La respuesta es ruidosa debido a que la ventana es binaria y rectangular,
- El operador responde muy fácilmente a los bordes debido a que sólo el mínimo de f es tomado en consideración.

Usando la matriz de auto-correlación (ver ec 3.11) y una ventana Gaussiana, se pueden solucionar los problemas anteriores. La matriz $A(x, y)$ promedia las derivadas de una señal en una ventana W alrededor del punto (x, y) [33]:

$$A(x,y) = \begin{bmatrix} \sum_W (I_x(x_k,y_k))^2 & \sum_W I_x(x_k,y_k) I_y(x_k,y_k) \\ \sum_W I_y(x_k,y_k) I_x(x_k,y_k) & \sum_W (I_y(x_k,y_k))^2 \end{bmatrix} \tag{3.11}$$

Para la matriz A se tienen dos valores característicos α y β los cuales son proporcionales a las curvaturas de la función local de auto-correlación por lo que se tienen igualmente 3 casos:

a) Si los dos valores característicos son pequeños entonces la región es homogénea.

b) Si uno de los valores característicos es grande entonces indica la presencia de un borde.

c) Si los dos valores característicos son grandes entonces esto indica una esquina.

En lugar de usar una ventana binaria y rectangular, se utiliza una función Gaussiana para pesar las derivadas dentro de la ventana, es decir:

$$W(x,y) = \frac{1}{2\pi\sigma^2} exp^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{3.12}$$

No solamente es necesario detectar los bordes o las esquinas sino también medir la calidad de las respuestas, para lograrlo se hace uso de la traza de A y del determinante de A (ver ec 3.13), evitando así el cálculo explícito de los valores característicos.

$$cornerness = Det(A) - k\ Tr(A)^2 \tag{3.13}$$

donde k es una constante tal que $0 < k < 0.25$, usualmente $k = 0.04$.

Una vez que se aplica la ecuación 3.13 a cada pixel, se realiza una detección del máximo en una vecindad de 3×3 pixeles. Para poder hacer una detección más confiable, se hace uso de un umbral, para el cual aquellos pixeles que tengan un valor de *cornerness* arriba del umbral, se considerarán como puntos de interés, aquellos que estén por debajo del umbral se desechan.

3.3.2. Scale Invariant Feature Transform (SIFT)

El algoritmo de SIFT (*Scale Invariant Feature Transform*) fue propuesto por David Lowe [18] y ha ganado mucha popularidad en los últimos años en diferentes aplicaciones donde es necesaria la extracción de características locales dentro de una imagen, como el reconocimiento de objetos y escenas [40][57], seguimiento de objetos [23] o reconocimiento de rostros [42]. Entre las características más importantes están: invariancia a rotación y escala, es parcialmente invariante a cambios en la iluminación y a la posición $3D$ de la cámara, y además es muy distintivo.

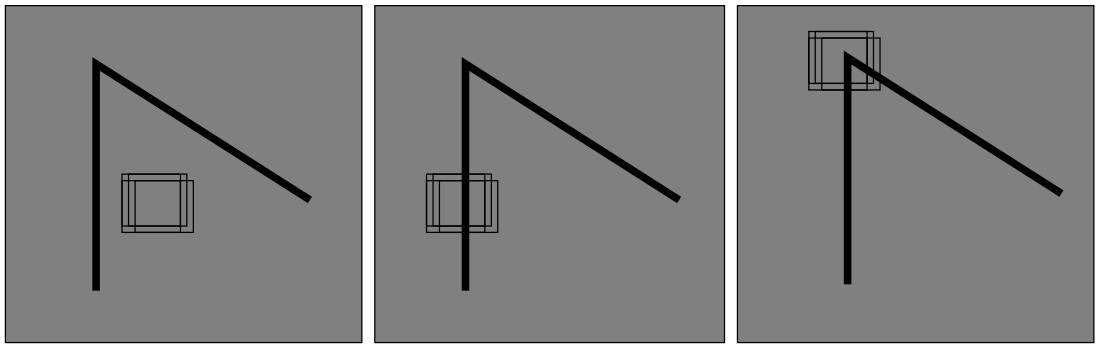

Figura 3.4: Movimientos de la ventana en diferentes direcciones discretas: a) Región plana, b) Borde, c) Esquina

Detección de puntos de interés

El algoritmo SIFT logra la invariancia a escala buscando características estables dentro de todas las posibles escalas construyendo una representación piramidal del espacio-escala (ver figura 3.2a). Para detectar eficientemente las posiciones de los puntos de interés se propone el uso de la función de DoG (*Difference-of-Gaussians*), la cual se aproxima a la función Laplaciana de Gaussiana (LoG) de escala-normalizada $\sigma^2\nabla^2 G$ estudiada por Lindeberg [61]. En comparaciones mas detalladas [34], se demuestra que el máximo y el mínimo de $\sigma^2\nabla^2 G$ producen características más estables de la imagen comparado con otras funciones como lo son la matriz Hessiana, el gradiente o la función de Harris.

La función DoG (D) se logra restando imágenes de escalas sucesivas L separadas por un factor k, es decir:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3.14}$$

El uso de esta función es computacionalmente menos costoso que construir la LoG ya que previamente se construye la representación espacio-escala y solamente es necesaria una resta entre imágenes (ver figura 3.5).

Para detectar los puntos de interés se extrae el mínimo y el máximo de las DoG buscando en una región de $3 \times 3 \times 3$. Para obtener una mejor localización del punto se realiza una extensión al algoritmo en donde se ajusta una función cuadrática de $3D$ (x, y, σ) sobre los puntos previamente localizados, con ésto se obtiene una localización más precisa tanto en la escala como en el espacio [43]. Esta idea usa la expansión de la serie de Taylor de la función espacio-escala, $D(x, y, \sigma)$, es decir:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{3.15}$$

Donde el punto de interés esta denotado por $\mathbf{x} = (x, y, \sigma)$. La ubicación correcta del extremo, es determinada sumando al punto de interés $\mathbf{x}$, un offset $\hat{\mathbf{x}} = (x, y, \sigma)$,

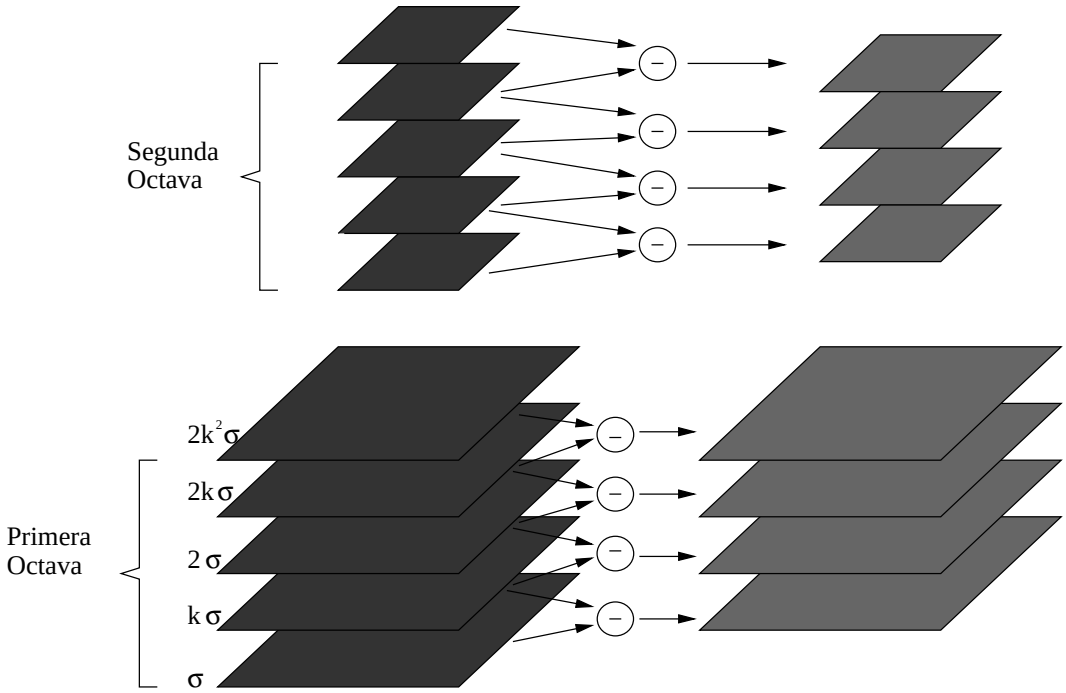

Figura 3.5: Construcción de espacio-escala: La figura de la izquierda hace la representación de espacio-escala como pirámide Gaussiana, la figura de la derecha representa la DoG la cual es construida con la resta de imágenes sucesivas en la pirámide Gaussiana [18]

que es obtenido tomando la derivada de la ec 3.15 con respecto a $\mathbf{x}$ he igualándola a cero,

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}} \tag{3.16}$$

Obteniendo el siguiente sistema de ecuaciones de 3×3,

$$\frac{\partial^2 D}{\partial \mathbf{x}^2}\hat{\mathbf{x}} = \frac{\partial D}{\partial \mathbf{x}} \tag{3.17}$$

$$\begin{bmatrix} \frac{\partial^2 D}{\partial \sigma^2} & \frac{\partial^2 D}{\partial \sigma y} & \frac{\partial^2 D}{\partial \sigma x} \\ \frac{\partial^2 D}{\partial \sigma y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y x} \\ \frac{\partial^2 D}{\partial \sigma x} & \frac{\partial^2 D}{\partial y x} & \frac{\partial^2 D}{\partial x^2} \end{bmatrix} \begin{bmatrix} \sigma \\ x \\ y \end{bmatrix} = - \begin{bmatrix} \frac{\partial D}{\partial \sigma} \\ \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \end{bmatrix} \tag{3.18}$$

Cabe destacar que para obtener un cómputo más eficiente, se pueden aproximar las derivadas mediante la resta de pixeles vecinos, por ejemplo:

$$\begin{aligned} \frac{\partial D}{\partial \sigma} &= \frac{D_{k+1}^{i,j} - D_{k-1}^{i,j}}{2} \\ \frac{\partial D}{\partial x} &= \frac{D_{k}^{i+1,j} - D_{k}^{i-1,j}}{2} \end{aligned} \tag{3.19}$$

donde i, j indican las coordenadas del punto $x(i, j)$ dentro de la imágen y k es el nivel en la escala.

Para tener una mayor estabilidad en los puntos detectados, se hace uso de la traza y el determinante de la matriz Hessiana ($\mathcal{H}$) con lo cuál se rechazan aquellos puntos que estén pobremente definidos a lo largo de los bordes. La ecuación es la siguiente, donde r es un umbral [19]:

$$\frac{Tr(\mathcal{H})^2}{Det(\mathcal{H})} < \frac{(r+1)^2}{r} \tag{3.20}$$

Descripción de puntos de interés

El primer paso para la construcción del descriptor es asignar una orientación consistente a cada punto de interés, incorporando esta orientación al descriptor se logra la invariancia a la rotación. La manera de obtener la orientación dominante es calculando los vectores gradientes en una ventana alrededor del punto de interés, para cada muestra dentro de la ventana se obtiene la magnitud $m(x, y)$ y la orientación $\theta(x, y)$ del gradiente usando diferencias entre pixeles vecinos,

$$\begin{aligned} \theta(x,y) &= tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right), \\ m(x,y) &= \sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \end{aligned} \tag{3.21}$$

Con estas orientaciones se construye un histograma de 36 intervalos cubriendo un rango de 360 grados. Cada muestra es añadida al histograma y es pesada por la magnitud de su gradiente y por una ventana circular Gaussiana con σ igual a 1.5 veces la escala del punto de interés.

En el histograma de orientación se detecta el pico más alto, así como todos aquellos picos que estén por arriba del 80 % del pico más alto por lo que para un mismo punto se pueden tener diferentes orientaciones dominantes por lo que se crean otros puntos de interés con cada orientación dominante.

Para que el descriptor pueda ser invariante a la rotación, la ventana es rotada con respecto a la orientación dominante, es decir, las orientaciones de los gradientes dentro de la ventana son rotados relativamente a la orientación dominante del punto de interés, es decir, cada punto de interés tiene una orientación dominante que siempre apunta en la misma dirección, aun y cuando la imagen esté rotada, al girar la ventana con respecto a esta orientación, se logra que los valores del descriptor siempre sean los mismos

Para construir el descriptor cada muestra es pesada con una función Gaussiana con σ igual a la mitad del tamaño de la ventana del descriptor. La ventana se divide en subregiones de 4×4 en donde se obtiene un histograma con 8 orientaciones por cada subregión. El propósito de la ventana Gaussiana es evitar cambios repentinos

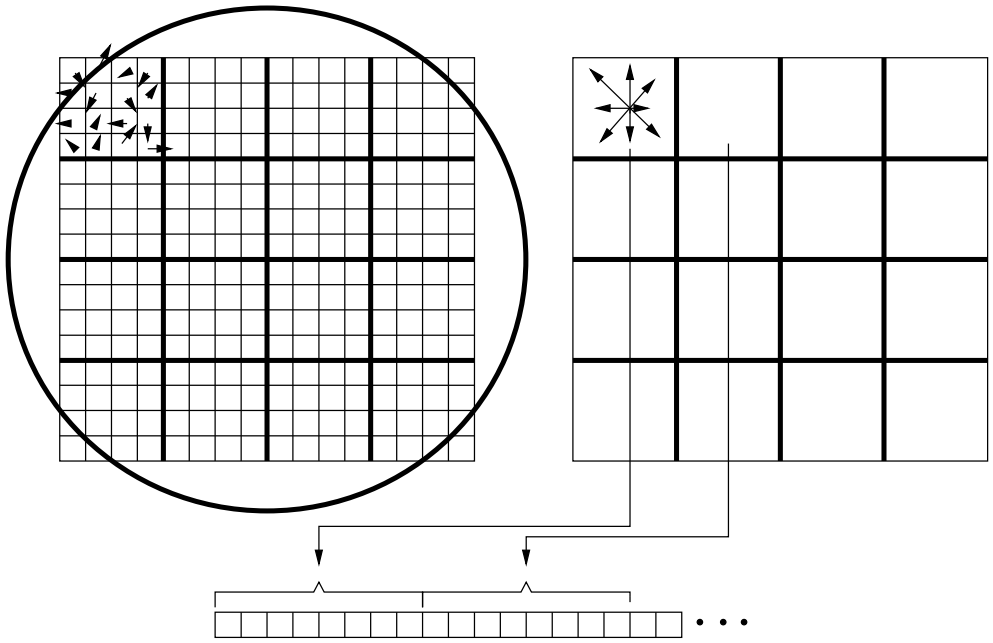

Figura 3.6: Construcción del descriptor de SIFT: La figura de la izquierda representa la ventana cuadrada de donde se obtiene la orientación dominante, donde cada gradiente es pesado con una función Gaussiana. La figura de la derecha representa la ventana rotada a lo largo de la orientación dominante y dividida en subregiones de 4×4 a partir de los cuáles se construye el descriptor. La figura de abajo muestra el descriptor representado como un vector de dimensión 128.

en el descriptor con pequeñas variaciones en la posición de la ventana, así como dar más énfasis a las muestras mas cercanas al punto (ver figura 3.6).

3.3.3. Detector de puntos de interés Harris-Laplacian

Fue desarrollado en 2001 por Mikolajczyk y Schmid [35] y surge para contrarestar las limitantes que tiene el detector de Harris con respecto a la invariancia a escala. En este detector los puntos son detectados usando el detector de Harris. Para lograr la invariancia a escala se hace uso de la selección automática de escala [63] y así seleccionar puntos localizados en el espacio-escala.

Como se vió en la sección 3.3.1, el detector de Harris está basado en la matriz de segundo momento o de auto-correlación, la matriz es adaptada para cambios en la escala, es decir:

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \sigma_D^2 \; g(\sigma_I) * \begin{bmatrix} \frac{\partial L}{\partial x}^2(\mathbf{x}, \sigma_D) & \frac{\partial L}{\partial x \partial y}(\mathbf{x}, \sigma_D) \\ \frac{\partial L}{\partial x \partial y}(\mathbf{x}, \sigma_D) & \frac{\partial L}{\partial y}^2(\mathbf{x}, \sigma_D) \end{bmatrix} \tag{3.22}$$

Donde σ_I es la escala de integración y σ_D es la escala de diferenciación. Como se puede observar en la ec 3.22, la escala de derivación es utilizada para calcular el gradiente en cada punto, y la escala de integración es usada para sumar las

matrices de segundo momento en una ventana alrededor del punto de interés, es decir, la matriz de segundo momento describe la distribución del gradiente en una vecindad local alrededor del punto, en donde las derivadas locales son calculadas con filtros Gaussianos con tamaño determinado por la escala local σ_D. Las derivadas son promediadas en una vecindad suavizando con una ventana Gaussiana de tamaño σ_I.

La calidad de la respuesta dada por la ec 3.13 es modificada de la siguiente forma:

$$cornerness = Det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) - k\ Tr(\mu(\mathbf{x}, \sigma_I, \sigma_D))^2 \tag{3.23}$$

Dado un punto en una imagen y un operador de selección de escala se calculan las respuestas del operador para un conjunto de escalas σ_n. El operador es la función Laplaciana-de-Gaussianas la cual en [35] se demuestra que comparado con otros tiene el mayor porcentaje para encontrar las escalas características correctas. El operador es el siguiente:

$$|LoG(x, \sigma_n)| = \sigma_n^2 |\frac{\partial^2 L}{\partial x^2}(\mathbf{x}, \sigma_n) + \frac{\partial^2 L}{\partial y^2}(\mathbf{x}, \sigma_n)| \tag{3.24}$$

El detector de Harris-Laplace usa la ec 3.23 para localizar los puntos en el espacio-escala y con la ec 3.24 se seleccionan los puntos para los cuáles se logra un máximo sobre la escala. En [35] se proponen dos algoritmos para selección de puntos de interés, el primero de ellos detecta simultáneamente la posición y la escala. El segundo de ellos es más simplificado obteniendo así un algoritmo más eficiente pero menos preciso. A continuación se explica el primero de ellos.

Este algoritmo consta de dos fases: primero se construye una representación de espacio-escala usando la función de Harris para pre-seleccionar las escalas $\sigma_n = \alpha^n \sigma_0$ en donde el punto de interés posiblemente se encuentre. α es el factor de escala entre niveles sucesivos en el espacio-escala (se toman intervalos largos en éste paso, es decir, $\alpha = 1.4$). Se construye la matriz $\mu(\mathbf{x}, \sigma_n)$ con $\sigma_I = \sigma_n$ y $\sigma_D = s\sigma_n$, donde s es un factor constante. Para cada nivel se detectan los puntos extrayendo los máximos en una vecindad de 3×3 alrededor del punto $\mathbf{x}$.

En la segunda fase, para cada punto se aplica un algoritmo iterativo para detectar la posición y la escala del punto de interés. Dado un punto inicial $\mathbf{x}$ con escala σ_I, para cada punto k, los pasos son los siguientes:

1 Encontrar el máximo local sobre escala de LoG para el punto $\mathbf{x}^{(k)}$ en un rango de escalas limitado por $\sigma_I^{(k+1)} = t\sigma_I^{(k)}$ con $t \in$[0.7,...,1.4], de otra manera se rechaza el punto.

2 Detectar la posición espacial $\mathbf{x}^{(k+1)}$ del máximo de la medida de cornerness de Harris más cercana a $\mathbf{x}^{(k)}$ para la $\sigma_I^{(k+1)}$ seleccionada.

3 Ir al paso 1 si $\sigma_I^{(k+1)} \neq \sigma_I^{(k)}$ o $\mathbf{x}^{(k+1)} \neq \mathbf{x}^{(k)}$

3.3.4. Speeded Up Robust Features: SURF

Algoritmo desarrollado en 2006 por Herbert Bay, Tinne Tuytelaars y Luc Van Gool [25], surge como una aproximación para la detección y la descripción de puntos de interés. Los resultados demuestran que iguala o mejora esquemas previamente propuestos con respecto a la repetibilidad, distinctividad, robustez y velocidad de cómputo sin sacrificar el desempeño.

Una de las características del algoritmo que acelera el cómputo es el uso de imágenes integrales y respuestas tipo Haar-wavelet. Las imágenes integrales fueron introducidas por Paul Viola y Michael Jones para un sistema de detección visual de objetos en donde su uso acelera el procesamiento de la imagen haciendo que la detección de características sea más rápida [48].

Un punto (x, y) en una imagen integral $I_{integral}(x, y)$ representa la suma de todos los pixeles dentro de una región rectangular formada por el punto y el origen en la imagen de entrada $I(x', y')$, es decir,

$$I_{integral}(x, y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} I(x', y'). \tag{3.25}$$

Cabe destacar que una vez calculada la imagen integral, se pueden obtener características rectangulares con solo cuatro operaciones agilizando así el computo y permitiendo la rápida implementación de filtros de convolución de tipo caja (box type filters) independientemente de su tamaño [48], usados para la detección y la descripción de puntos de interés (ver figura 3.7).

Detección de puntos de interés

La detección de puntos de interés tanto en la posición como en la escala se basa en el uso de la matriz Hessiana (ver ec 3.9), específicamente el valor obtenido del cálculo del determinante es el que sirve como medida para la extracción de puntos de interés:

$$det(\mathcal{H}_{aprox}) = \frac{\partial^2 D}{\partial x^2}\frac{\partial^2 D}{\partial y^2} - 0{,}9(\frac{\partial^2 D}{\partial x \partial y})^2 \tag{3.26}$$

Se usan filtros de caja ya que aproximan las derivadas Gaussianas de segundo orden y pueden ser evaluadas rápidamente usando la imagen integral (ver figura 3.8).

Debido al uso de imágenes integrales y filtros de caja, el espacio-escala se construye aplicando filtros de diferente tamaño a la imagen original (ver figura 3.2b) con el mismo costo computacional. La salida del filtro de 9×9 se considera como la escala inicial, la cuál es referida como la escala s =1.2 (correspondiente a la derivada Gaussiana con σ =1.2). Las siguientes escalas son obtenidas aplicando gradualmente a la imagen filtros más grandes tomando en cuenta la naturaleza discreta de las imágenes integrales y la estructura específica de los filtros.

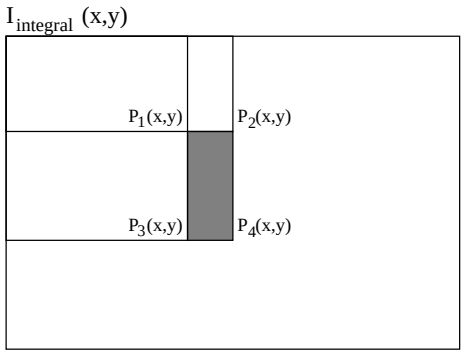

Figura 3.7: Imagen integral: El rectángulo sombreado puede ser calculado con 4 operaciones: $P_1 + P_4 - P_2 - P_3$

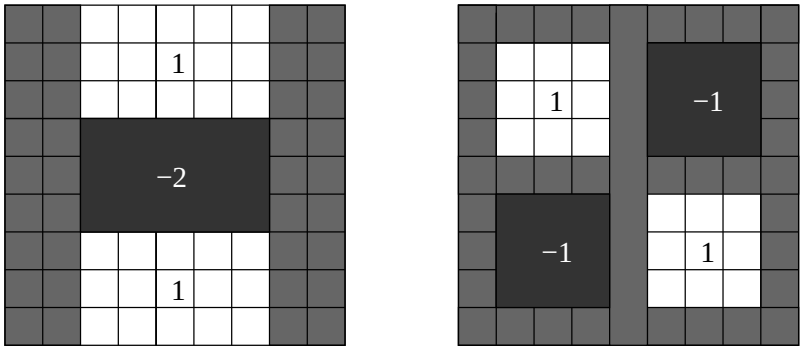

Figura 3.8: Aproximación de las segundas derivadas Gaussianas en la dirección y (izquierda) y xy (derecha), haciendo uso de los filtros de caja. Las regiones grises son iguales a cero.

Específicamente, los filtros aplicados son de tamaño $9\times9, 15\times15, 21\times21, 27\times27$, etc. Para cada nueva octava, el tamaño del paso entre filtros es incrementado el doble, siendo 6 para la primera octava, 12 para la segunda, 24 para la tercera, etc. Es decir, por ejemplo:

Primera Octava: $9 \times 9, 15 \times 15, 21 \times 21, 27 \times 27, 33 \times 33$.
Segunda Octava: $39 \times 39, 51 \times 51, 63 \times 63, 75 \times 75, 87 \times 87$.

Simultáneamente el intervalo de muestras para la extracción de puntos de interés puede ser el doble.

Debido a que la disposición de los coeficientes de los filtros permanecen constantes después de ser escalados, la aproximación de las derivadas Gaussianas se escalan de manera acorde, por lo tanto, como el filtro de 9×9 corresponde a σ =1.2, el filtro de 27×27 corresponde a $\sigma = 1.2 \times 3 = 3.6 = s$.

Para localizar los puntos de interés se extraen los máximos en una región de $3 \times 3 \times 3$ alrededor de el punto de interés.

Descripción de los puntos de interés

Para lograr que la descripción de los puntos sea invariante a la rotación, el primer paso consiste en obtener una orientación dominante del punto de interés basándose en la información obtenida en una región circular alrededor de él. Después se construye una región cuadrada alineada con la orientación obtenida previamente y ahí se obtiene el descriptor.

La orientación dominante se obtiene primeramente calculando las respuestas tipo Haar-wavelets con una longitud $4s$ en las direcciones x y y (ver figura 3.9) de muestras tomadas en una vecindad circular de radio $6s$ alrededor del punto de interés. Las respuestas son pesadas con una función Gaussiana centrada en el punto de interés. Debido al uso de la imagen integral, solo 6 operaciones son necesarias para calcular la respuesta en la dirección x o y en cualquier escala.

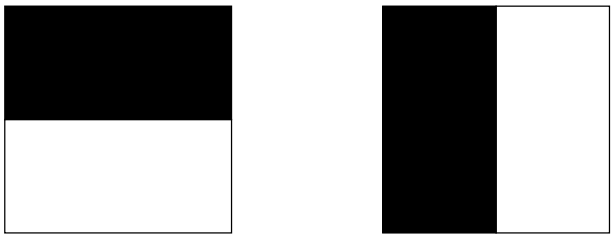

Figura 3.9: Respuestas Haar-wavelets en dirección x y y

Las respuestas son representadas como vectores donde la respuesta en la dirección x está a lo largo de las abscisas y la respuesta en y a lo largo de las ordenadas. Para obtener la orientación dominante, se tiene que obtener la orientación en una ventana deslizante cubriendo un ángulo de $\pi/3$ sobre el rango de 2π (ver figura 3.10). El tamaño de la ventana deslizante es un parámetro determinado experimentalmente.

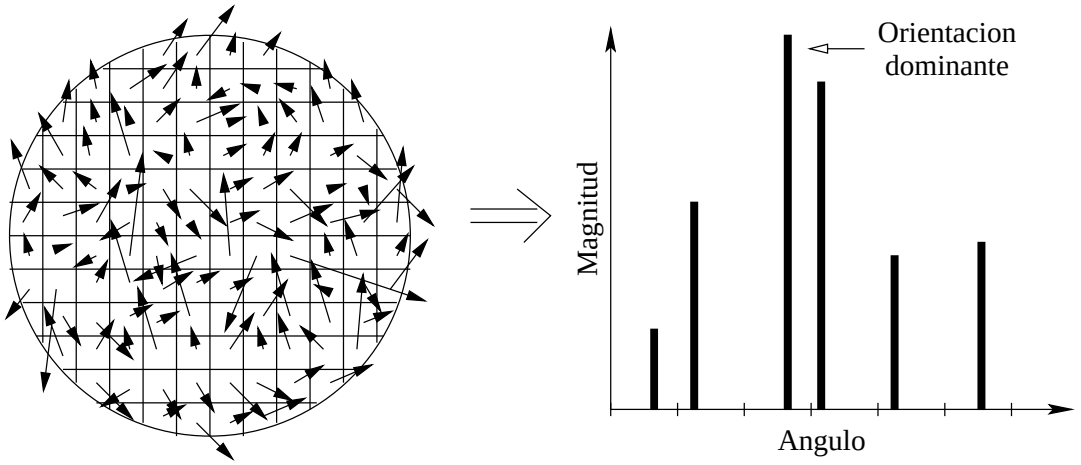

Figura 3.10: Orientación dominante: a)Respuestas calculadas en la región circular de tamaño $6s$ alrededor del punto de interés, b)Gráfica que muestra la orientación dominante calculada a partir de ventanas de tamaño $\pi/3$

La manera en la que se obtiene la orientación de cada ventana es sumando los vectores que se encuentran dentro de ella. El vector más largo de las 6 ventanas resultantes es el que fija la orientación dominante del punto de interés.

Una vez obtenida la orientación dominante, el siguiente paso es la construcción del descriptor. Para extraer el descriptor se define una región cuadrada de tamaño $20s$ centrada alrededor del punto de interés y orientada con respecto a la orientación dominante obtenida previamente, esta región es dividida en 4×4 subregiones cuadradas. En cada subregión se toman muestras regularmente espaciadas y se calculan las respuestas tipo Haar wavelet en las direcciones x y y, la longitud de estos filtros es de $2s$. Por simplicidad las respuestas son llamadas d_x y d_y, estas a su vez son pesadas con una función Gaussiana ($\sigma = 3.3s$) centrada en el punto de interés, ésto es para lograr robustez a deformaciones geométricas y errores de localización.

Las respuestas sobre cada subregión son sumadas obteniendo un vector por cada una de ellas. Con estos vectores y la suma del valor absoluto de las respuestas sobre cada subregión se obtiene la entrada total del descriptor por lo que cada subregión contribuye al descriptor con un vector v de 4 dimensiones $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ obteniendo un descriptor de longitud 64. La estructura del descriptor es la siguiente:

$$D = (\sum d_{x_i}, \sum d_{y_i}, \sum |d_{x_i}|, \sum |d_{y_i}|, ...) \tag{3.27}$$

donde $i = 1, ..., 16$. Una vez construido el descriptor, es convertido a un vector unitario para lograr invariancia al contraste [25].

Capítulo 4

Desarrollo del Sistema de Seguimiento de Objetos

En este capítulo se explica detalladamente el desarrollo y la implementación del seguidor de objetos desarrollado en la tesis. En capítulos anteriores se presentaron los antecedentes necesarios para la mejor comprensión de todos los aspectos que engloba el sistema desarrollado, así como el estado del arte en cuanto a las diferentes aproximaciones que se podría haber optado.

Se mencionó que en un sistema típico de seguimiento de objetos, las dos estructuras básicas en que podría dividirse el sistema son: (1) la representación del objeto y (2) el filtrado o estimación del siguiente estado. A continuación se hará una descripción del filtro usado en este trabajo.

Existen varios aspectos donde los modelos determinísticos en sistemas dinámicos no son adecuados para un buen análisis y diseño de un sistema. Entre ellos se encuentran, principalmente, la gran cantidad de variables o parámetros que se tienen que modelar y no son modeladas debido a que algunas de ellas no pueden ser expresadas determinísticamente debido a las características de los sistemas dinámicos. Incluir muchas variables produciría un modelo computacionalmente muy costoso y además se tiene que considerar que la extracción de los datos por medio de sensores no provee de una perfecta descripción del proceso. Todo esto incrementaría la incertidumbre al modelo, por lo que se tendría optar por el uso de modelos estocásticos, que son de gran importancia para obtener mejores resultados en el modelado del comportamiento del sistema.

4.1. El filtro de Kalman

El filtro de Kalman es un algoritmo recursivo óptimo para estimar el estado de un proceso. Las razones de su optimalidad proviene de la incorporación de toda la información que pueda ser proporcionada, es decir:

- Conocimiento del sistema y del dispositivo.

- Información estadística sobre los errores en las mediciones y ruido en el sistema.

- Además se puede incorporar información acerca de las condiciones iniciales del sistema.

Es recursivo debido a que usa información anterior para estimar el siguiente estado, pero una de las características importantes de éste filtro es el uso solamente de información de un estado de tiempo anterior, es decir, no se guarda toda la información previa del sistema sino solamente se usa el estado previo y las mediciones actuales para estimar el estado posterior.

Para entender mejor tanto las ecuaciones como las ideas básicas de este filtro es conveniente comprender el concepto de espacio-estado para el modelado de sistemas dinámicos. Un sistema dinámico se define como aquel en el que existe una dependencia del sistema con respecto al tiempo, es decir, el estado futuro depende de una regla o modelo matemático que (puede ser una ecuación diferencial) relaciona el estado presente con el estado futuro.

Existen muchas aplicaciones en donde el ruido en las mediciones no es tomado en consideración para generar el modelo, por ejemplo, en el desarrollo de algunos juegos por computadora. Sin embargo, es muy importante tenerlo en cuenta ya que se puede tener una idea más real de lo que sucede en el sistema. Si se toma en cuenta el ruido entonces el modelo se complica un poco mas ya que el ruido es generalmente aleatorio, lo que nos lleva a modelos estocásticos para representar el problema.

4.1.1. Concepto de espacio-estado

Los modelos de espacio-estado son esencialmente una notación para problemas de estimación y control. Es un modelo matemático de un sistema físico, representado como un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales. Espacio-estado se refiere al espacio cuyos ejes son las variables de estado en donde el estado del sistema puede ser representado con un vector dentro de ese espacio. El uso de ecuaciones diferenciales obtenidas del modelo físico, generalmente arrojan un conjunto de variables de estado como son las posiciones o velocidades aunque este conjunto no es único [1].

La representación general del modelo espacio-estado de un sistema lineal con p entradas, q salidas y n variables de estado está dada por:

a) Tiempo Continuo

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) + w(t), \tag{4.1}$$

$$\mathbf{y}(t) = C(t)\mathbf{x}(t) + v(t). \tag{4.2}$$

b) Tiempo Discreto

$$\mathbf{x}_{k+1} = A(k)\mathbf{x}_k + B(k)\mathbf{u}_k + w_k, \tag{4.3}$$

$$\mathbf{y}_k = C(k)\mathbf{x}_k + v_k. \tag{4.4}$$

donde:

- $\mathbf{x} \in \mathbf{R}^n$ representa al vector de estado,

- $\mathbf{y} \in \mathbf{R}^q$ representa al vector de salida,

- $\mathbf{u} \in \mathbf{R}^p$ representa al vector de entrada (o control),

- w y v representan el error o ruido en el proceso y medición respectivamente,

- A es la matriz de estado, $dim[A] = n \times n$,

- B es la matriz de entrada (o control), $dim[B] = n \times q$,

- C es la matriz de mediciones, $dim[C] = q \times n$,

Debido a las características del sistema desarrollado en la tesis, a partir de este momento se usara el tiempo discreto. La ec. 4.3 representa el estado posterior como la combinación lineal del estado anterior, un posible vector de control y ruido del proceso. La ec. 4.4 describe la las mediciones u observaciones derivadas del estado interno. Estas 2 ecuaciones generalmente son llamadas modelo del proceso y modelo de medición, respectivamente.

4.1.2. Ecuaciones del filtro de Kalman

El filtro de Kalman (KF) esencialmente es un conjunto de ecuaciones matemáticas que implementan un estimador óptimo en donde básicamente se tienen dos etapas: predictor y corrector. Es óptimo ya que al incorporar todas los datos disponibles, además del conocimiento previo del sistema y del dispositivo de medición, cuando se cumplen algunas condiciones asumidas previamente, el filtro estima las variables deseadas de tal manera que minimiza el error estadísticamente [51].

Las condiciones que se asumen son: que el sistema pueda ser descrito con un modelo lineal y que el ruido tanto del sistema como de las mediciones sea blanco (es decir, que no esté correlacionado en el tiempo y tenga la misma potencia para cualquier frecuencia) y Gaussiano.

El filtro de Kalman trata de estimar el estado $x \in \mathbf{R}^n$ de un proceso discreto gobernado por la siguiente ecuación[24]:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}, \tag{4.5}$$

con una medición $z \in \mathbf{R}^m$ dada por:

$$z_k = Hx_k + v_k \tag{4.6}$$

w_k y v_k son variables aleatorias que representan respectivamente el ruido del proceso y de la medición, se asume que son independientes, con media cero y con distribuciones de probabilidad normales,

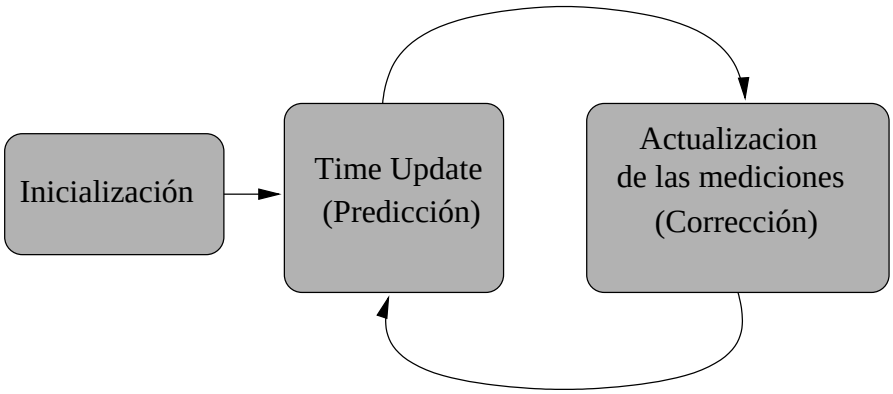

Figura 4.1: Fases del filtro de Kalman (inicialización, corrección y predicción)

$$p(w) \sim N(0, Q), \tag{4.7}$$

$$p(v) \sim N(0, P). \tag{4.8}$$

En donde:

A - matriz de $n \times n$ que relaciona el estado en un paso previo de tiempo $k-1$ con el estado en el tiempo actual k,
B - matriz de $n \times l$ que relaciona la entrada opcional de control $u \in \mathbf{R}^l$ con el estado x,
H - matriz de $m \times n$ que relaciona el estado x_k con la medición z_k,
Q - matriz de covariancia de la perturbación del proceso,
P - matriz covariancia de la perturbación en la medición.

En la práctica estas matrices podrían cambiar en el tiempo, pero por simplicidad se asumen son constantes.

El algoritmo del filtro de Kalman tiene 2 fases distintas: predicción y corrección. La fase de predicción usa el estimado del estado en un tiempo anterior para producir un estimado del estado en el tiempo actual. En la fase de corrección se usa la información de las mediciones del estado actual para refinar la predicción y así llegar a una estimación más precisa (ver figura 4.1).

Las ecuaciones que describen cada fase son las siguientes [51]:

a) Predicción

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{4.9}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.10}$$

b) Corrección

$$K_k = \frac{P_k^- H^T}{(H P_k^- H^T + R)} \tag{4.11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4.12}$$

$$P_k = (I - K_k H) P_k^- \tag{4.13}$$

En donde:

$\hat{x}_k^-$ - Estimación *a priori* del estado k,

$\hat{x}_k$ - Estimación *a posteriori* del estado k dada la medición z_k,

P_k^- - Estimación *a priori* de la covariancia de error en el estado k,

K_k - Ganancia que minimiza la covariancia del error de la nueva estimación del estado.

Para mayores detalles en la obtención de éstas ecuaciones ver [51][45][56]

4.2. Filtro de Kalman Extendido

De la sección 4.1 se puede decir que el KF estima el estado $x \in \mathbf{R}^n$ de un proceso discreto gobernado por una ecuación diferencial lineal estocástica. Cuando el proceso a ser estimado es no-lineal entonces se hace uso del EKF (Extended Kalman Filter), es un filtro de Kalman que linealiza la media y la covarianza a partir de transformaciones lineales.

4.2.1. Ecuaciones del filtro de Kalman Extendido

Se asume que el proceso tiene también el vector de estado $x \in \mathbf{R}^n$, gobernado por la siguiente ecuación diferencial estocástica no-lineal [51]:

$$x_k = f(x_{k-1}, u_k, w_{k-1}), \tag{4.14}$$

con la medición $z \in \mathbf{R}^m$:

$$z_k = h(x_k, v_k), \tag{4.15}$$

En la práctica los valores individuales de w_k y v_k no se conocen en cada paso de tiempo, sin embargo, se pueden aproximar el vector de estado y medición omitiendo estos valores, es decir,

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_k, 0), \tag{4.16}$$

con la medición $z \in \mathbf{R}^m$:

$$\tilde{z}_k = h(\tilde{x}_k, 0), \tag{4.17}$$

Las nuevas ecuaciones que linearizan el estimado de las ecuaciones 4.16 y 4.17 son:

$$x_k \approx \tilde{x}_k + A_k(x_{k-1} - \tilde{x}_{k-1}) + W_k w_{k-1}, \tag{4.18}$$
$$z_k \approx \tilde{z}_k + H_k(x_k - \tilde{x}_k) + V_k v_k, \tag{4.19}$$

donde:

- $\tilde{x}_k$ y $\tilde{z}_k$ son la aproximación del vector de estado actual y de medición dadas las ecuaciones 4.16 y 4.17.
- $\hat{x}_k$ es la estimación *a posteriori* del estado en el tiempo k,
- A matriz Jacobiana de derivadas parciales de f con respecto de x,
- W matriz Jacobiana de derivadas parciales de f con respecto de w,
- H matriz Jacobiana de derivadas parciales de h con respecto de x,
- V matriz Jacobiana de derivadas parciales de h con respecto de v,

El conjunto de ecuaciones completo se muestra a continuación, notar que se sustituye $\tilde{x}_k$ por $\hat{x}_k^-$ para tener una consistencia con la notación del filtro de Kalman en donde el signo $-$ significa "a priori",

a) Ecuaciones para la fase de predicción

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0), \tag{4.20}$$
$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T. \tag{4.21}$$

b) Ecuaciones para la fase de corrección

$$K_k = \frac{P_k^- H_k^T}{(H_k P_k^- H_k^T + V_k R_k V_k^T)} \tag{4.22}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \tag{4.23}$$
$$P_k = (I - K_k H_k) P_k^- \tag{4.24}$$

donde:

- Q_k es la covariancia del ruido del proceso en el tiempo k,
- H_k y V_k son los jacobianos de la medición en el tiempo k,
- R_k es la covariancia del ruido de la medición en el tiempo k.

4.3. La transformada Unscented

A pesar de que el EKF provee de una manera la estimación de sistemas no-lineales que mantiene la forma elegante y computacionalmente eficiente del algoritmo del KF, sufre de diversas limitantes [30]:

- Las transformaciones lineales son confiables solamente si la propagación del error puede ser aproximada por una función lineal. Si esta condición no se cumple entonces la linearización puede ser extremadamente pobre.

- La linearización puede ser aplicada sólo si la matriz Jacobiana existe. Sin embargo esto no es siempre el caso.

- Computacionalmente, el cálculo de las matrices Jacobianas puede ser muy complejo.

La transformada Unscented (UT) surge para interponerse a estas limitaciones, es decir, provee de un mecanismo más directo para transformar la información de la media y la covariancia. La idea básica de la UT es que es más fácil aproximar una distribución de probabilidad que aproximar una función o transformación arbitraria no-lineal.

La UT calcula el efecto de una función no-lineal hacia la media y la covariancia. Opera tomando determinísticamente un conjunto de muestras (llamadas puntos-sigma) las cuales son propagadas a través de la función no-lineal resultando en una nube de puntos transformados (ver figura 4.2, figura tomada de [30]). La estadística de los puntos transformados puede ser calculada para formar un estimado de la media y la covariancia.

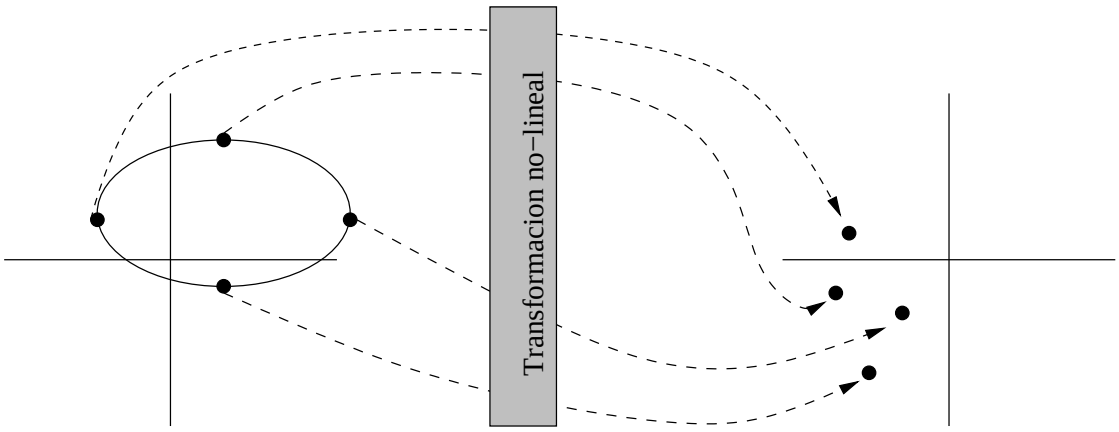

Figura 4.2: Principio de la Transformada Unscented

La UT tiene propiedades importantes que la hacen muy atractiva para su uso en problemas no-lineales

- Debido a que el algoritmo trabaja con un número finito de puntos, puede ser implementada en una librería de tipo "black box", donde dado un modelo y definiendo las entradas y salidas apropiadas, se puede crear una rutina estándar que puede predecir las variables necesarias para una transformación dada.

- No es necesario el cálculo de Jacobianos y el costo computacional es el mismo que el del EKF.

Dada una variable aleatoria n-dimensional con media $\bar{x}$ y covariancia $\mathrm{P_{xx}}$, ésta se puede aproximar con $2n+1$ puntos con sus respectivos pesos W_i, es decir, se tiene un conjunto de puntos-sigma $\mathcal{S}$ que consiste de $p+1$ vectores así como de sus pesos asociados, tal que $\mathcal{S} = i = 0, 1, ..., p : x_i, W_i$,

$$\begin{aligned} x_0 &= \bar{x} W_0 = \kappa/(n+\kappa) \\ x_i &= \bar{x} + (\sqrt{(n+\kappa)}\ \mathrm{P_{xx}})_i W_i = 1/2(n+\kappa) \\ x_{i+n} &= \bar{x} - (\sqrt{(n+\kappa)}\ \mathrm{P_{xx}})_i W_{i+n} = 1/2(n+\kappa) \end{aligned} \tag{4.25}$$

donde $\kappa \in \mathbf{R}$, $(\sqrt{(n+\kappa)}\mathrm{P_{xx}})_i$ es la i-esima fila o columna de la matriz raíz cuadrada de $(n+\kappa)\mathrm{P_{xx}}$.

Una vez obtenidos los puntos y_i transformados,

$$y_i = h[x_i]. \tag{4.26}$$

la media $\bar{y}$ y la covarianza $\mathrm{P_{yy}}$ se obtienen de la siguiente manera:

$$\bar{y} = \sum_{i=0}^{2n} W_i y_i. \tag{4.27}$$

$$\mathrm{P_{yy}} = \sum_{i=0}^{2n} W_i [y_i - \bar{y}][y_i - \bar{y}]^T \tag{4.28}$$

κ provee de un grado de libertad extra para un ajuste fino hacia los momentos de grado alto de la aproximación, y puede ser usado para reducir los errores de predicción. Cuando $x(k)$ se asume Gaussiana, una heurística es usar $n + \kappa = 3$.

En [30] se puede consultar mejor el algoritmo de selección de puntos-sigma, así como el uso la serie de Taylor para demostrar que la transformada Unscented es mejor que el uso de linearización en aplicaciones de filtrado y estimación, debido a que puede predecir la media y la covariancia con una precisión de segundo orden, por lo cual cualquier filtro que use la transformada Unscented tiene el mismo desempeño que el Filtro de Gauss truncado hasta el segundo orden con la ventaja de que no se calculan ni Jacobianos ni Hessianos [58]

4.3.1. Unscented Kalman Filter

Una vez visto el Filtro de Kalman, se puede decir que es un algoritmo que cuenta con dos pasos recursivos, los cuales son:

- Predicción. Predice el estado en el tiempo k usando el estimado de del estado en el tiempo $k-1$, es decir, $\hat{x}_k^-$ así como la covariancia P_k^- (ver ecs. 4.9 y 4.10).
- Corrección. En esta etapa, se mezcla la medición z_k con la predicción $\hat{x}_k^-$ para dar un nuevo estimado $\hat{x}_k$ así como P_k (ver ecs. 4.11 y 4.12).

El UKF es una extensión de la UT aplicado al algoritmo recursivo del Filtro de Kalman. En donde el vector de estado es redefinido como la concatenación del estado original con las variables de ruido $x_k^a = [x_k^T, v_k^T, n_k^T]^T$, x_k^a es una notación para definir el vector de estado aumentado. Después, un algoritmo como el mostrado en la ec 4.25 es usado para calcular la matriz sigma X_k^a.

El algoritmo de predicción usando la UT es el siguiente:

1) Se inicializan valores

$$\hat{x}_0^a = [\hat{x}_0^T, 0, 0]^T$$
$$\mathbf{P}_0^a = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix}$$

2) Se calculan los puntos sigma (ver ec 4.25) for X_{k-1}^a

3) Predicción

$$\begin{aligned} X_{k|k-1}^x &= F[X_{k-1}^x, X_{k-1}^v] \\ \hat{x}_k^- &= \sum_{i=0}^{2n} W_i X_{i,k|k-1}^x \\ P_k^- &= \sum_{i=0}^{2n} W_i [X_{i,k|k-1}^x - \hat{x}_k^-][X_{i,k|k-1}^x - \hat{x}_k^-]^T \\ Y_{k|k-1} &= H[X_{k|k-1}^x, X_{k-1}^n] \\ \hat{y}_k^- &= \sum_{i=0}^{2n} W_i Y_{i,k|k-1} \end{aligned} \tag{4.29}$$

4) Corrección

$$\begin{aligned} P_{y_k y_k} &= \sum_{i=0}^{2n} W_i [Y_{i,k|k-1} - \hat{y}_k^-][Y_{i,k|k-1} - \hat{y}_k^-]^T, \\ P_{x_k y_k} &= \sum_{i=0}^{2n} W_i [X_{i,k|k-1} - \hat{x}_k^-][Y_{i,k|k-1} - \hat{y}_k^-]^T \end{aligned} \tag{4.30}$$

4.4. Seguidor de Objetos

4.4.1. Descripción del algoritmo

En la figura 4.3 se puede observar un diagrama a bloques del seguidor de objetos desarrollado en éste trabajo de tesis (para una mejor visualización del algoritmo, ver el apéndice D).

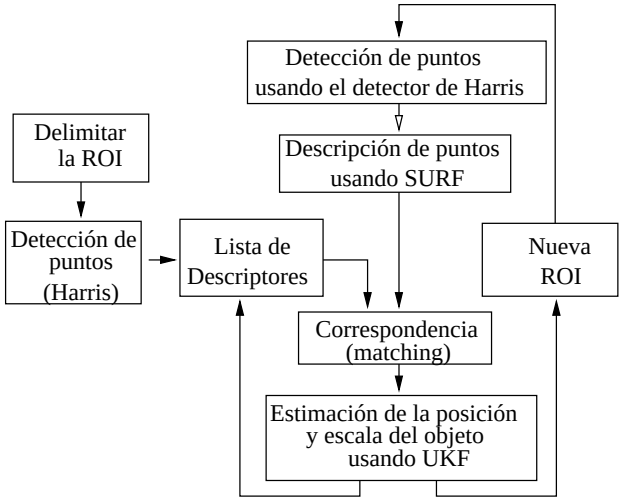

Figura 4.3: Diagrama a bloques del sistema

Este seguidor de objetos podría dividirse en dos fases principales. La primer fase del algoritmo consiste en la representación del objeto, esta representación puede hacerse de diferentes maneras por ejemplo: color del objeto, contorno o intensidad. En este trabajo se usan puntos característicos o puntos interesantes (ver Sección 3.3), esto permite que la definición del objeto no asuma una forma fija, obteniendo así grandes ventajas, ya que se usa un modelo predefinido para cada objeto que se quiera representar. Para seleccionar el objeto simplemente se delinea o selecciona la región en la imagen que se quiere representar, esto genera una ROI (Region of Interest) en donde los puntos característicos son extraídos (ver figura 4.4).

Los puntos de interés son extraídos usando el detector de Harris [13] y como es sabido, este detector no extrae puntos en diferentes escalas, para resolver la invariancia a escala se construye una lista de descriptores en muchas escalas (resoluciones) para cada uno de los puntos detectados, esta idea se ha probado que trabaja efectivamente en un algoritmo de SLAM (*Simultaneous Localization and Mapping*) usando los descriptores de SIFT (ver Sección 3.3.2) [14].

Un posible problema de esta aproximación es tener muchos puntos representando el objeto, haciendo el proceso de correspondencia (*matching*) mas costoso. Esto es resuelto en las siguientes fases del algoritmo durante el proceso de filtrado y predicción, en donde los descriptores son construidos a una escala fija y se obtiene la correspondencia sólo en una región de la lista de descriptores predicha por el UKF.

Figura 4.4: Región de interés para una sola característica mostrando la orientación y el tamaño de la ventana del descriptor en una escala fija

La descripción de los puntos interesantes una vez extraídos por el detector de Harris, se realiza con una aproximación denominada SURF (ver Sección 3.3.4) [25], la cual ha sido probada que aproxima o mejora esquemas previamente propuestos, y el hecho de que puede ser calculada mas rápidamente la hace mas adecuada para este sistema.

Una vez que los puntos característicos son extraídos y comparados con aquellos extraídos en el primer cuadro (*frame*) el siguiente paso es calcular el centro del objeto. En el primer cuadro, éste puede ser obtenido directamente ya que de antemano se sabe la posición espacial de la ROI, en los siguientes cuadros durante el seguimiento del objeto, el centro puede ser calculado de dos maneras, la primera de ellas es considerando las coordenadas relativas de los puntos calculados en el primer cuadro con el centro del objeto, la segunda es calculando la homografía, se utiliza la primera sólo cuando no se obtiene una buena homografía.

En la segunda fase del algoritmo, el UKF estima cuadro por cuadro la posición espacial del centro del objeto dentro de la imagen, así como la escala del objeto. Esto es muy importante para el desempeño del seguidor, debido a que los puntos característicos son solamente extraídos dentro de una área de la imagen limitada por la incertidumbre en la escala y el centro predicho por el UKF. La escala y la incertidumbre en la escala predichas por el filtro son usadas también para buscar dentro de una región en la lista de descriptores en donde el sistema va a buscar por posibles candidatos para hacer correspondencia. Si la cámara pierde el objeto, la incertidumbre en la escala y en la posición se incrementa hasta que la región de búsqueda sea toda la imagen, y la búsqueda de descriptores sea en toda la lista.

Como se mencionó anteriormente, la región de búsqueda dentro de la lista de descriptores está delimitada por la escala y la incertidumbre predicha por el filtro, es decir, la región de búsqueda está representada por las siguientes ecuaciones:

$$R_{max} = Sc + 3\sigma_{Sc} \qquad (4.31)$$
$$R_{min} = Sc - 3\sigma_{Sc}$$

donde, σ_{Sc} es la desviación estándar de la estimación de la escala Sc, la región de búsqueda queda delimitada por $[R_{min}, R_{max}]$, asegurando encontrar el candidato correcto con alta probabilidad (ver figura 4.5).

4.4.2. Representación del Objeto

Se decidió extraer los puntos característicos usando el detector de Harris el cuál muestra mucha repetibilidad y distintividad. En la evaluación de puntos de interés presentada en [12] es demostrado que el detector de Harris se desempeña mejor que otras aproximaciones existentes. Este detector es conocido no solamente por detectar esquinas sino puntos en la imagen donde la señal cambia en dos dimensiones, esto es logrado usando la matriz de segundo momento o matriz de auto-correlación.

Después de que los puntos de interés son extraídos de la ROI, el siguiente paso es describir cada uno de los puntos usando características locales, es decir, una vecindad alrededor del punto. Con la ventaja de que la información extraída de cada punto es local, por lo que no se necesita segmentar la imagen, permitiendo ocluir hasta cierto punto al objeto y seguir detectando puntos pertenecientes a el. La descripción de estos puntos es alcanzado usando descriptor confiable, robusto y rápido llamado SURF [25], el cuál hace uso de imágenes integrales [48] y respuestas tipo Haar-wavelets (ver Seccion 3.3.4).

Todo el algoritmo de SURF hace uso de la imagen integral tanto para detectar características como para construir el descriptor. En éste trabajo de tesis solo se hace uso del descriptor debido a su buen desempeño y rápida obtención, en [25] se demuestra que el algoritmo de SURF aproxima o mejora previos esquemas con respecto a repetibilidad, distintividad y robustez, además puede ser calculado y comparado más rápidamente. Estas características lo hacen ideal para nuestra aplicación.

4.4.3. Descriptores Multi-escala

En lugar de usar una representación espacio-escala para lograr la invariancia a los cambios de escala, inicialmente se construye una lista de descriptores a múltiples escalas (resoluciones) para cada uno de los puntos de interés. Esto se logra fuera de línea, es decir, durante el primer cuadro después de haber seleccionado la ROI. Este esquema es útil en dos maneras: primero, con esta aproximación se logra la invariancia a la escala y segundo, se tiene un uso eficiente de los recursos computacionales.

En los cuadros subsecuentes, los descriptores son solamente construidos a una escala fija, por lo que se hace uso de la lista de descriptores para comparar cada punto detectado y encontrar el más parecido (ver figura 4.5).

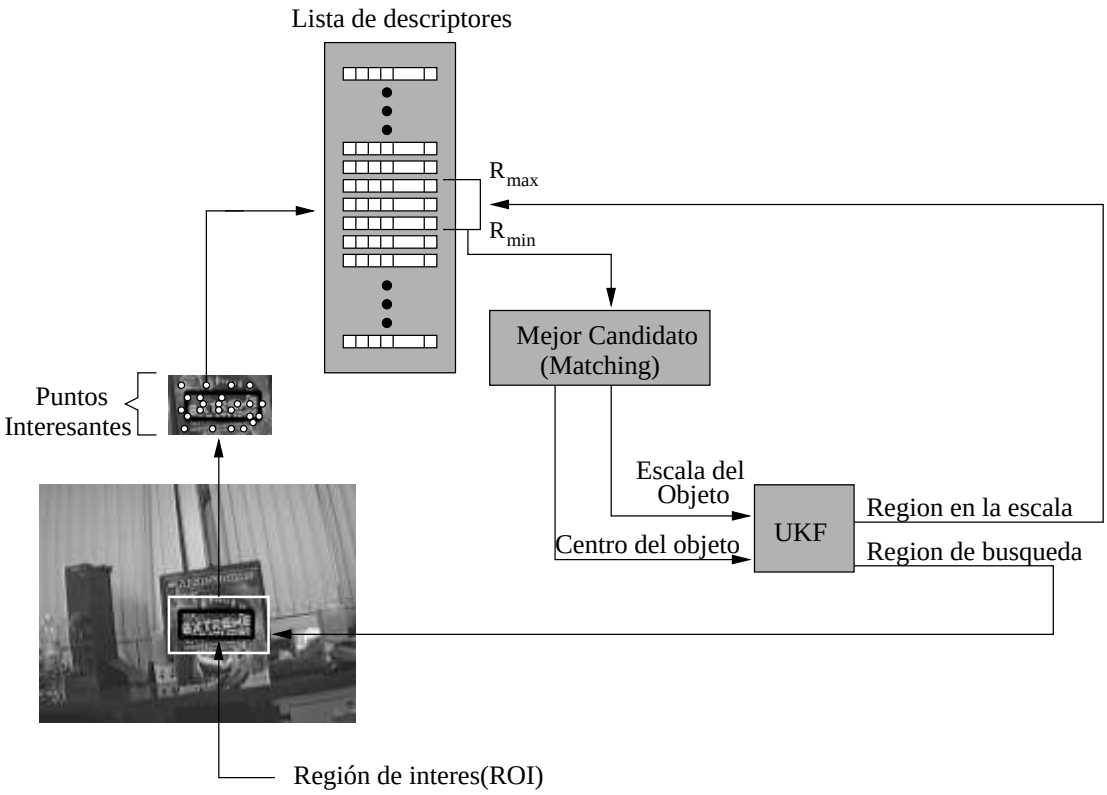

Figura 4.5: Vista esquemática del algoritmo

No solamente la región donde el descriptor es calculado es dependiente de la escala sino también la distancia entre las muestras, lo que significa que el número de muestras es fijo aunque cambie la escala, solamente se necesita incrementar o decrementar el tamaño de la ventana y la longitud entre los intervalos de muestreo acorde con la escala a la cuál el descriptor va a ser extraído (ver figura 4.6) [14].

En el primer cuadro, se conoce la posición espacial de la ROI(objeto a seguir), y el centro del objeto P_c. Para determinar el centro del objeto en los siguientes cuadros, se guarda la posición relativa (x, y) de cada punto interesante al centro del objeto. Así, es posible hacer uso de estas medidas y la escala del objeto para poder obtener el centro. Esto puede ser observado en la ecuación 4.32.

$$Pc(x, y) = \frac{\sum_{i=0}^{N}(M_i(x, y)/Sc_i) + p_i(x, y)}{N} \tag{4.32}$$

Donde N es el número de puntos que hacen match, Sc_i es la escala del punto i, $M_i(x, y)$ es la distancia del punto interesante al centro del objeto y $p_i(x, y)$ es la posición espacial de cada punto con respecto a la imágen.

4.4.4. Filtrado y estimación

En esta etapa del algoritmo, se utilizó el UKF para predecir tanto la posición (x, y) del objeto dentro de la imagen, como la escala del objeto. Como se mencionó en

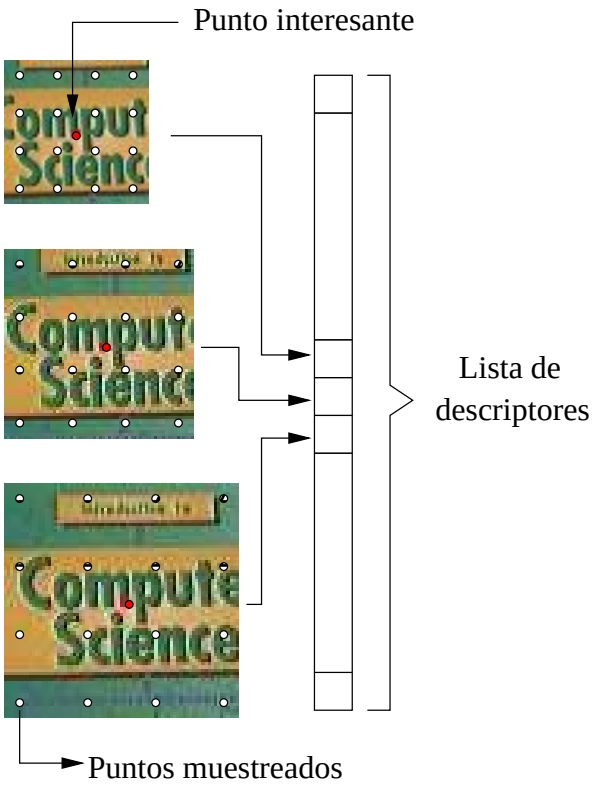

Figura 4.6: Obtención de muestras para el cálculo del descriptor

la sección 3.3.1, para hacer uso de el algoritmo del UKF se necesita especificar un modelo del proceso $F(\mathbf{x})$ el cuál es básicamente un modelo de movimiento de la cámara y como crece se incrementa la incertidumbre en la posición y escala. Para poder reducir la incertidumbre, se describe también un modelo de observación o medición $H(\mathbf{x})$. En donde $\mathbf{x}$ es el vector de estado del filtro, definido por la posición y la escala del objeto, es decir: $p = [\mathbf{r}, Sc]$, tal que $\mathbf{r} = [x, y]^T$ y Sc es la escala del objeto.

Construir un modelo de movimiento para una cámara con movimientos rápidos, la cual puede ser portada por una persona o por un robot, es diferente a modelar el movimiento por ejemplo de un robot móvil en un plano. La principal diferencia radica en el hecho de que en el caso del robot, uno tiene el control de las entradas de control del robot (p.ej, "moverse 1m con un ángulo de 0.5rad"). En el caso de la cámara, no se tiene la información de los movimientos de la persona o de la cámara con anterioridad.

Para este trabajo, se uso un modelo de velocidad constante. La implicación de este modelo no significa que la cámara se mueve con la misma velocidad en todo momento sino que se impone cierta certeza de que la cámara tiene movimientos suaves, es decir, se descartan aceleraciones muy grandes ya que son relativamente improbables.

El modelo del proceso es el siguiente:

$$F(\mathbf{x}, V_r, V_{Sc}) = [r + V_r, Sc + V_{Sc}, V_r, V_{Sc}]^T \tag{4.33}$$

en donde, $V_r = [V_x, V_y]$; V_x, V_y y V_{Sc} es la velocidad en la coordenada x, y y en la escala. La ecuación para el modelo de medición queda definida como:

$$H(\mathbf{x}) = [r + n_r, Sc + n_s]^T \tag{4.34}$$

en donde, n_r y n_s es ruido con una distribución normal en la posición y en la escala respectivamente.

4.4.5. Correspondencia (*matching*)

Para poder comparar entre los puntos de interés que representan al objeto, es decir, los puntos extraídos de la región seleccionada durante el primer cuadro, con aquellos detectados en los cuadros sub-secuentes, es necesario hacer uso de por lo menos un algoritmo que nos de una medida de similitud.

En este trabajo se utiliza un algoritmo de matching muy sencillo pero eficiente ya que los puntos detectados incorrectos son casi nulos. La idea fue tomada del trabajo de [25] debido a la sencillez en la implementación y los buenos resultados arrojados durante los experimentos.

Para hacer la comparación entre vectores, se obtiene la distancia euclidiana. La distancia euclidiana entre 2 puntos $p1(x1, y1)$ y $p2(x2, y2)$ se define con la ec 4.35.

$$d(p1, p2) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{4.35}$$

Haciendo uso de la ec 4.35 para vectores de tamaño 64 y aplicando el siguiente algoritmo es como se obtiene el matching:

```
for(i=0; i<numPuntosOrig; i++)
{
  nn = 1000000;  //Vecino más cercano
  snn = 1000000; //Segundo vecino más cercano
  for(j=0; j<numPuntos; j++)
  {
      sum = 0.0;
      //Se calcula la distancia en el espacio del descriptor
      //La dimensión del descriptor es de 64 elementos

      for(k=0; k<65; k++)
        sum = sum + pow((original[i].desc[k]-nuevo[j].desc[k]),2);

      dist = sqrt(sum);

      //Se hace uso del vecino más cercano y del segundo
```

```
      //vecino más cercano

      if(dist<snn && dist>nn)
        snn = dist;

     if(dist < nn)
      {
        snn = nn;
        nn = dist;
        ind = j;
      }
  }

  //Si la distancia es 7 veces menor a la del segundo vecino
  //mas cercano, entonces se tiene una buena correspondencia
  if(nn<0.7*snn && nn<0.1) //Threshold
  {
    //Se guardan los matches
  }
}
```

donde:

- *numPuntosOrig* es el número de puntos detectados en el primer cuadro,
- *numPuntos* es el número de puntos detectados en cada cuadro,
- *nn* y *snn* primero y segundo vecino más cercanos,
- *original[].desc[]* y *nuevo[].desc[]* son respectivamente estructuras que guardan los descriptores de cada punto interesante encontrados fuera de linea y en cada cuadro sub-secuente.

4.4.6. Cálculo de la Homografía

El cálculo de la homografía correspondiente es de gran ayuda para tener una mejor vista del objeto en seguimiento. En un espacio de 2 dimensiones, una homografía se define como un mapeo entre un punto $p_o(x, y)$ en una imagen origen y el punto $p_d(x', y')$ en la imagen transformada.

Existen diferentes clases de mapeos, los principales son: afines, bilineales y proyectivos.

- Afin. Este tipo de mapeo incluye escala, translación, rotación e inclinaciones. Las características más importantes son la preservación de líneas paralelas y los puntos equi-espaciados a lo largo de una línea. Por ejemplo, un triángulo es mapeado a un triangulo, un rectángulo a un paralelogramo.

- Bilineal. En este tipo de mapeo, un cuadrado puede ser mapeado a cualquier cuadrilátero. La transformación del sistema coordenado origen al destino preserva lineas verticales y horizontales, y puntos equi-espaciados a lo largo de estas líneas, pero no preserva líneas diagonales.

- Proyectivo. Este mapeo comparte algunas de las características del mapeo afin pero en general, no se preservan los puntos equi-espaciados.

Debido a que esta tesis es enfocada en cambios de escala y rotación , y debido a sus características, se utiliza el mapeo afin para el cálculo de la homografía. Como se menciono en el capítulo 1, formalmente una transformación afin es una transformación lineal mas una translación. Una transformación $T(x)$ es lineal si $T(x+y) = T(x) + T(y)$ y $T(\alpha x) = \alpha T(x)$ para cualquier escalar α. Es afin si existe una transformación lineal $L(x)$ mas una constante c tal que $T(x) = L(x) + c, \forall x$. En general una mapeo afin en 2 dimensiones puede ser expresado algebraicamente de la siguiente manera [50]:

$$p_o = p_d M_{od} \tag{4.36}$$

es decir,

$$(x' \ y' \ 1) = (x \ y \ 1) \begin{pmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{pmatrix} \tag{4.37}$$

en donde M_{od} denota la matriz de transformación del sistema coordenado inicial al destino. Como se puede observar en la ec 4.37, la matriz de 3×3 tiene 6 grados de libertad y puede ser definida geométricamente especificando 3 puntos en el sistema coordenado origen y en el destino, como se muestra en la siguiente ecuación:

$$M_d = M_o M_{od} \tag{4.38}$$

$$\begin{bmatrix} x'_0 & y'_0 & 1 \\ x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix} \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{bmatrix} \tag{4.39}$$

Debido a que durante el primer cuadro es seleccionada la ROI interactivamente, se sabe de antemano los 4 puntos que conforman el objeto a seguir, es decir, las 4 esquinas del rectángulo delineado. Durante el seguimiento del objeto, si en cada cuadro se tienen por lo menos 3 correspondencias entonces es posible obtener la matriz de mapeo M_{od} y así obtener una homografía. La matriz M_{od} se obtiene a partir de 4.39. Teniendo 3 puntos origen y destino se obtienen 6 ecuaciones que definen la matriz M_{od} (ver ec 4.40).

$$\begin{aligned} x'_k &= ax_k + by_k + c \\ y'_k &= dx_k + ey_k + f \end{aligned} \tag{4.40}$$

para k = 0,1,2. Esto puede ser reescrito en un sistema de 6×6:

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix} \tag{4.41}$$

Este sistema lineal puede ser resuelto con cualquier método para obtener los coeficientes de mapeo $a - f$. En la mayoría de los casos se tienen más de 3 puntos que hacen correspondencia por lo que para ésto se usa el algoritmo de RANSAC (*Random Sample Consensus* [44]) para obtener la mejor homografía posible, además de descartar puntos que sean colineales ya que pueden llevar a aberraciones en la homografía.

RANSAC es un algoritmo para estimar parámetros de un modelo matemático a partir de un conjunto de datos observados, el cual contiene inliers (puntos que pertenecen al modelo) y outliers (puntos que no pertenecen al modelo). La entrada del algoritmo son un conjunto de datos u observaciones y un modelo para el cual estas observaciones pueden ser evaluadas, y la salida es el mejor modelo que representa al conjunto de datos. El algoritmo es el siguiente:

```
Dado:
datos: Observaciónes (que en este caso son puntos que hacen
       correspondencia),
modelo: Modelo con el que se prueban los datos (ec 3.39),
n: Número mínimo de datos requeridos para el modelo (n=3),
k: Número máximo de iteraciónes permitidas para el algorítmo,
t: Umbral para saber cuando un punto es considerado como inlier
d: Numero de datos inliers requeridos para considerar un buen modelo,

while(iteracion < k)
{
   error = Numero muy grande;
   puntosPrueba = n puntos seleccionados aleatoriamente;
   modeloPrueba = modelo generado a partir de puntosPrueba;
   inliers = 0;
   for(Todos los datos excluyendo puntosPrueba)
     {
        datoPrueba = Probar dato en modeloPrueba;
        if(datoPrueba < t)
           añadir dato a inliers;
     }
   if(num(inliers) > d)
   {
```

```
      modelo = modeloPrueba;
      errorModelo = medida de que tan bueno es el modelo;
      if(errorModelo < error)
      {
        mejorModelo = modelo;
        error = errorModelo;
      }
    }
    iteracion++;
}
return mejorModelo;
```

Ya que se tiene un modelo de la homografía, entonces es posible hacer uso de ésta para calcular el nuevo centro del objeto. Por lo que si no se tiene una homografía adecuada entonces se hace uso de la ec 4.32 para calcular el centro. En la figura 4.7 se muestra la salida final del sistema seguidor de objetos.

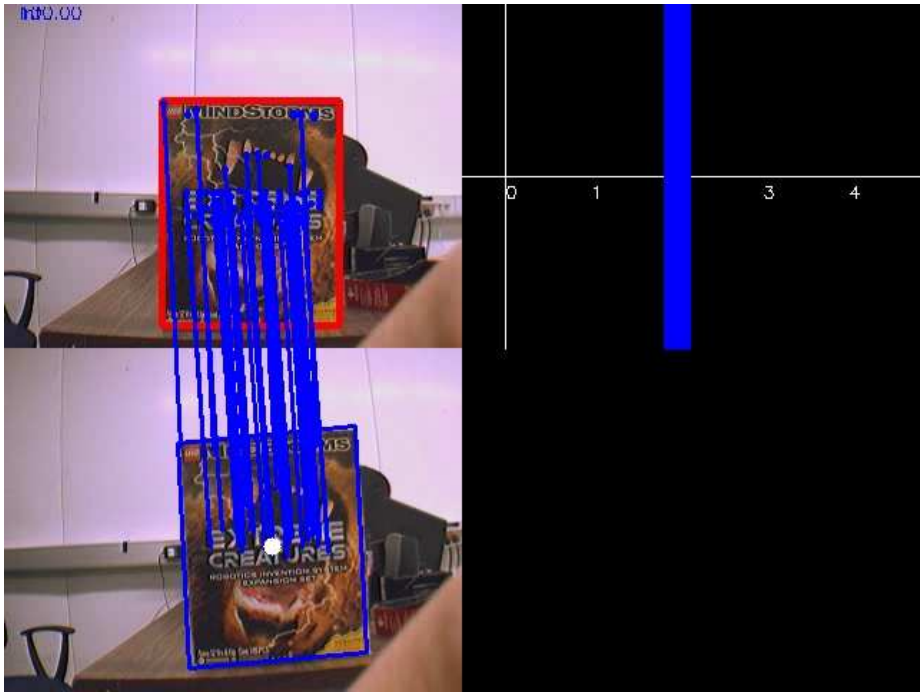

Figura 4.7: Salida final del sistema seguidor de objetos: La esquina superior izquierda muestra el primer cuadro, el objeto a seguir enmarcado en un cuadro rojo y los puntos detectados. La esquina inferior izquierda muestra la secuencia de video en donde el marco azul representa la homografía del objeto y las líneas azules indican los puntos que hacen correspondencia. La imágen superior derecha indica la escala y el ancho del rectángulo azul representa la incertidumbre en la escala arrojada por el UKF

Capítulo 5

Desempeño del sistema

En éste capítulo se muestra el desempeño del sistema evaluándolo bajo diferentes experimentos, estos experimentos se refieren principalmente a la comparación entre el sistema desarrollado en esta tesis (ver figura 4.5), es decir:

- Detector de puntos de Harris,
- construcción de descriptores a múltiples escalas (resoluciones) usando el descriptor de SURF,
- estimación de la escala y la posición del objeto usando el UKF.

y el algoritmo de SURF usando la librería provista por sus desarrolladores [67].

Se mostraran gráficas donde se comparara la velocidad de procesamiento y la cantidad de puntos detectados a lo largo del video. Cada secuencia de video consta de aproximadamente 200 a 300 cuadros. En el eje coordenado "x" se indica el numero de cuadro. En el eje coordenado "y" se muestra para el inciso a) la velocidad del algoritmo indicando la tasa de cuadros, representados en cps (cuadros por segundo o *frames per second*); y en el inciso b) se indica la cantidad de puntos detectados.

5.1. Software y hardware

El seguidor de objetos se programo en lenguaje C++ bajo la plataforma Linux. Se hace uso de la librería "OpenCV", comúnmente utilizada en aplicaciones de visión por computadora. En éste trabajo, la librería es usada para la extracción de imágenes, creación de videos y aplicación de algunos filtros (esta librería puede ser utilizada gratuitamente) [28].

El UKF es utilizado usando la librería "*Bristol Tracking Library*" desarrollada en el grupo de visión en tiempo real de la Universidad de Bristol, G.B [2]. Cabe mencionar que éste trabajo de tesis fue desarrollado conjuntamente en el grupo mencionado anteriormente bajo la supervisión del Dr. Walterio Mayol-Cuevas.

Los experimentos fueron realizados usando una computadora Dell inspiron 8500 con procesador Pentium 4 a 2.2 GHz. Se utilizó una cámara firewire "Unibrain" (ver figura 5.1) con las siguientes especificaciones:

Figura 5.1: Cámara Unibrain utilizada en el sistema

- 400Mbps 1394a, IIDC v1.04 DCAM
- Video VGA descomprimido hasta 30 cps
- Pixeles cuadrados, 1/4" CCD
- Resolución máxima de 640×480

5.2. Experimentos y resultados

Por razones de velocidad de procesamiento, la resolución de las imágenes utilizadas para probar el sistema son de tamaño 320×240.

En ésta sección se describen experimentos con 3 objetos diferentes (ver figura 5.2), probando el desempeño del sistema bajo cambios de escala, rotación y secuencias que incluyan oclusiones parciales y totales. Los objetos son: Caja 1 (objeto 1), Caja 2 (objeto 2), Taza (objeto 3).

Figura 5.2: Objetos utilizados para probar el desempeño del sistema: a) Caja 1, b) Caja 2, c) Taza

5.2.1. Invariancia a escala

Para determinar el desempeño del sistema a cambios de escala, los objetos se alejaron y se acercaron de la cámara por intervalos de $10cm$. En la secuencia 1, los objetos solamente se alejaron, determinando así la distancia máxima a la cual son detectados correctamente, en las siguientes figuras se muestran gráficas del desempeño del sistema evaluándolo en la secuencia 1 y comparándolo con el algoritmo de detección y descripción de SURF.

En la figura 5.3 se pueden observar gráficas del desempeño del objeto 1, en el inciso *a)* se puede apreciar que la tasa de cuadros de el sistema desarrollado en esta tesis, durante toda la secuencia es mejor, con una media de $\mu = 11.36\ cps$ comparado con $\mu_{SURF} = 6.58\ cps$ del algoritmo de SURF. Del inciso *b)* se puede observar en la gráfica, que el numero de puntos encontrados en cada alejamiento son comparativamente iguales en las dos aproximaciones obteniendo a lo largo de toda la secuencia una media de: $\mu = 32.97$ de 133 puntos detectados en el primer cuadro, y $\mu_{SURF} = 27.29$ de 130 puntos detectados en el primer cuadro. Los puntos detectados en el primer cuadro son los que originalmente caracterizan al objeto.

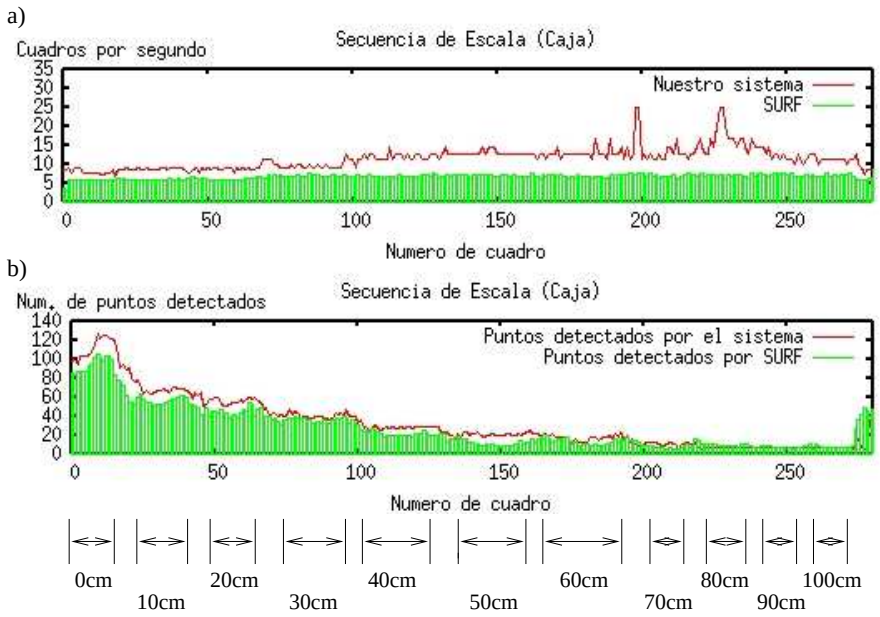

Figura 5.3: Desempeño del objeto 1 a cambios de escala en la secuencia 1. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos encontrados

En la figura 5.4 se muestra el desempeño del objeto 2, el cuál tiene menos puntos que lo caracterizan en el primer cuadro comparado con el objeto anterior, es decir: 46 puntos detectados por nuestro sistema y 62 por SURF. De ésta gráfica también se puede observar mejor como la tasa de cuadros de nuestro sistema es mucho mejor obteniendo una media de $\mu = 19.46\ cps$ comparado con $\mu_{SURF} = 7.04\ cps$. Se puede observar que la distribución de puntos encontrados a lo largo de la secuencia por las dos aproximaciones se comporta de manera similar.

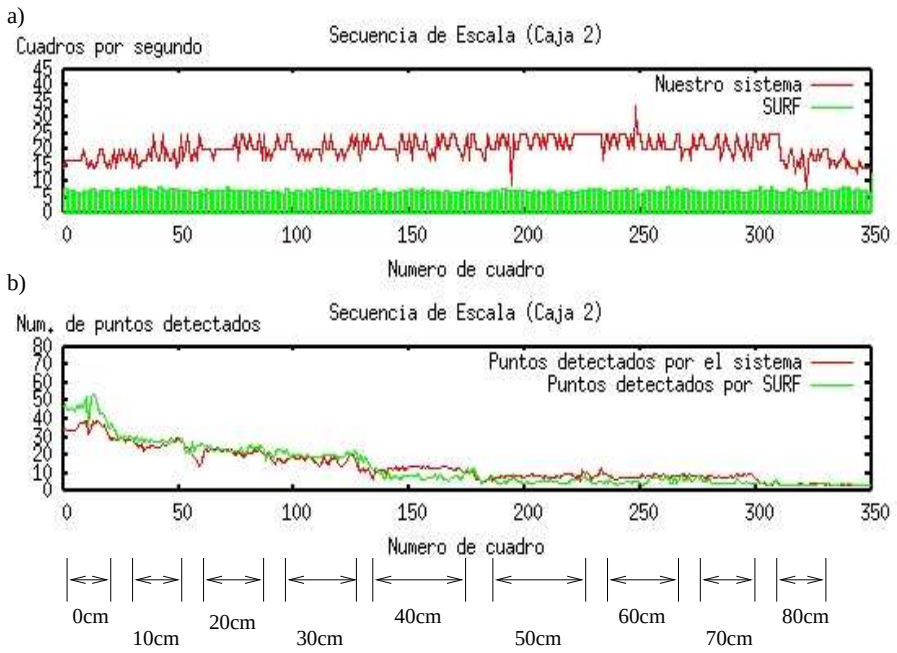

Figura 5.4: Desempeño del objeto 2 a cambios de escala en la secuencia 1. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

En la figura 5.5 se muestra el desempeño del objeto 3, este es el que tiene las dimensiones mas pequeñas comparado con los tres objetos en prueba, es de esperarse que el objeto sea reconocido a una distancia menor que los dos objetos anteriores. Del inciso *a)* se observa una velocidad de procesamiento mayor de nuestro sistema comparado con SURF obteniendo una media de $\mu = 15.12\ cps$ comparado con $\mu_{SURF} = 7.99\ cps$. Los puntos encontrados a lo largo del video se observan en el inciso *b)* obteniendo una media de $\mu = 8.69$ comparado con $\mu_{SURF} = 7.26$.

La figura 5.6 representa imágenes de muestra tomadas de la secuencia 1 de cada uno de los objetos, en donde los incisos a), b) y c) representan a los objetos 1, 2 y 3 respectivamente. Para tener una mejor visualización de algunas imágenes de prueba, referirse al apéndice A.

En la secuencia 2, los objetos se alejaron y se acercaron. Cabe destacar que la extracción de los puntos de interés que caracterizaron a los objetos durante la secuencia, se hizo con una distancia mayor a la de la secuencia 1, para así tener la habilidad de acercar los objetos hacia la cámara y que no se salieran de la imagen.

De las figuras 5.7, 5.8 y 5.9 inciso *a)* se puede comprobar que a lo largo del video, la tasa de cuadros es mejor para los tres objetos, obteniendo para el objeto 1 una media de $\mu = 14{,}15 cps$ comparado con $\mu_{SURF} = 6.42\ cps$, para el objeto 2 se obtuvo $\mu = 13.77\ cps$ y $\mu_{SURF} = 7.87\ cps$, y por último, para el objeto 3 se obtuvo una media de $\mu = 14.82\ cps$ comparado con $\mu_{SURF} = 7.62\ cps$. Con respecto a los puntos encontrados, tomando en consideración que los puntos que caracterizan a cada uno de los objetos con las 2 aproximaciones, son diferentes. La cantidad de

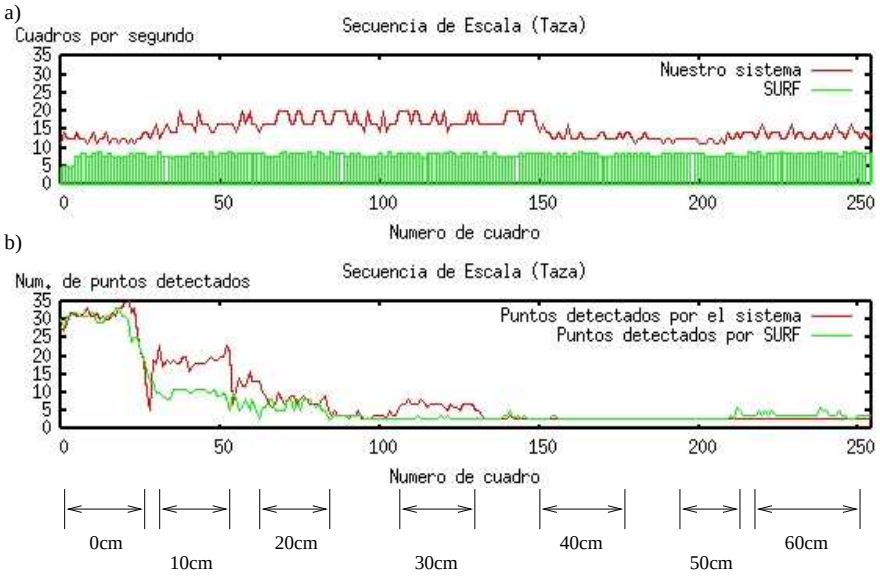

Figura 5.5: Desempeño del objeto 3 a cambios de escala en la secuencia 1. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

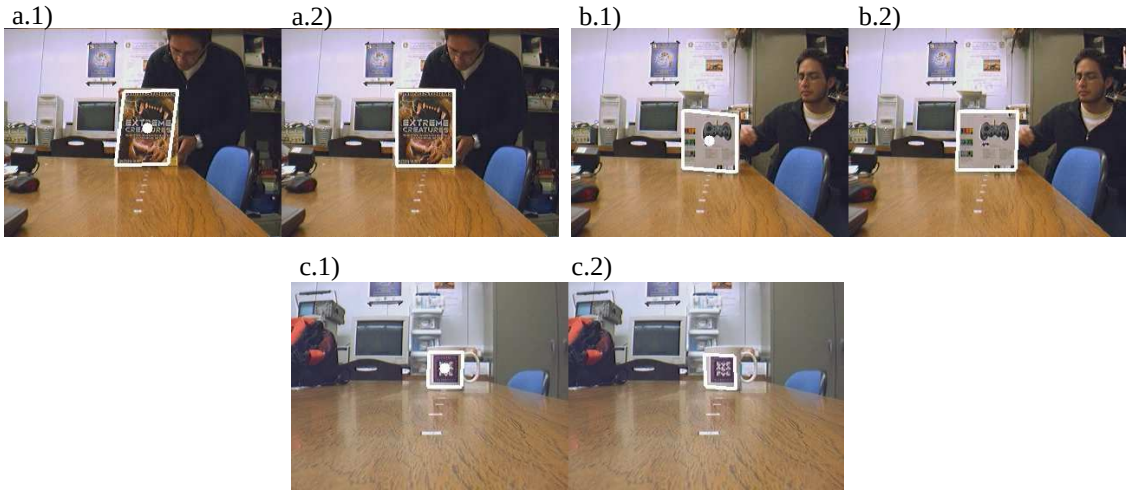

Figura 5.6: Esta figura representa imágenes de muestra de cada uno de los objetos, tomadas de la secuencia 1 durante el seguimiento, en donde el rectángulo blanco representa la homografía calculada. Los incisos a.1), b.1) y c.1) representan la salida del sistema desarrollado en ésta tesis, y los incisos a.2), b.2) y c.2) representan al algoritmo de SURF

puntos encontrados se comporta de manera similar a lo largo de la secuencia.

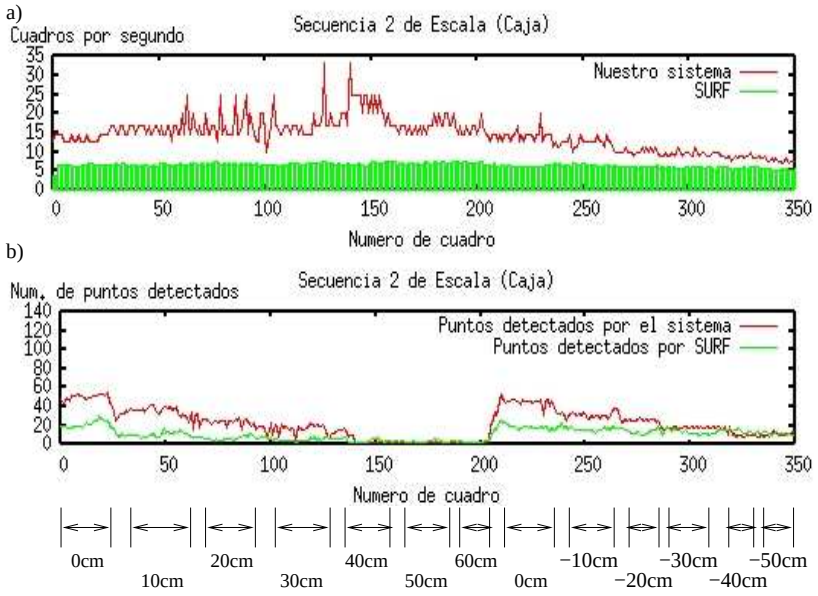

Figura 5.7: Desempeño del objeto 1 a cambios de escala en la secuencia 2. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

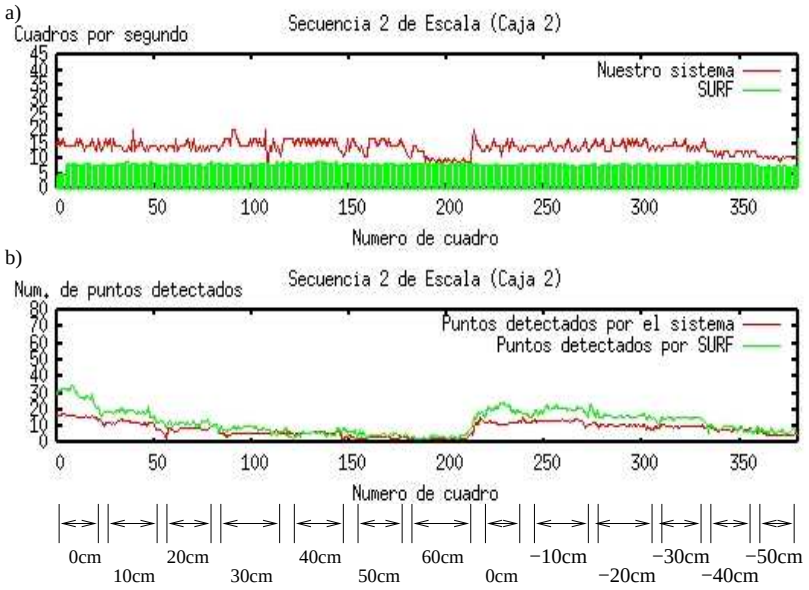

Figura 5.8: Desempeño del objeto 2 a cambios de escala en la secuencia 2. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

La figura 5.10 representa imágenes de muestra tomadas de cada uno de los objetos durante la secuencia 2. Para tener una mejor visualización de algunas imágenes

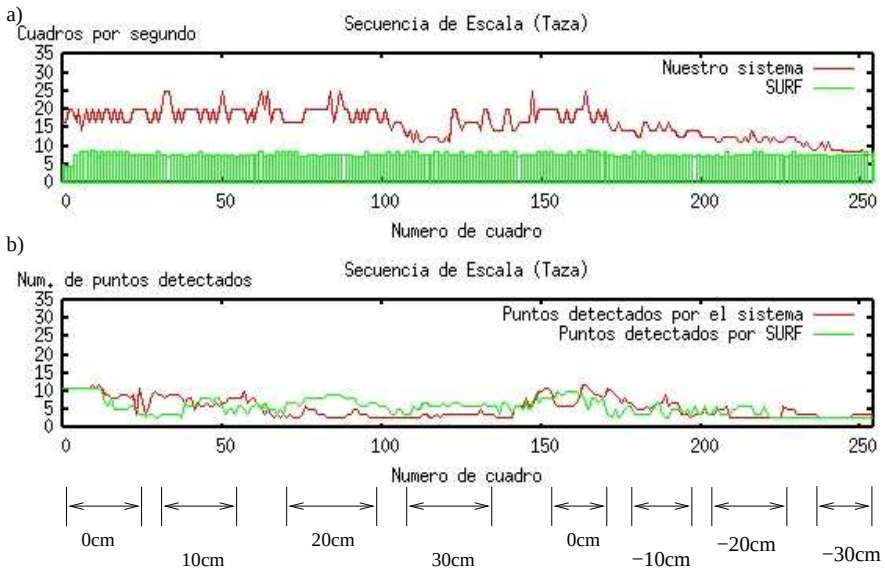

Figura 5.9: Desempeño del objeto 3 a cambios de escala en la secuencia 2. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

de prueba, referirse al apéndice A.

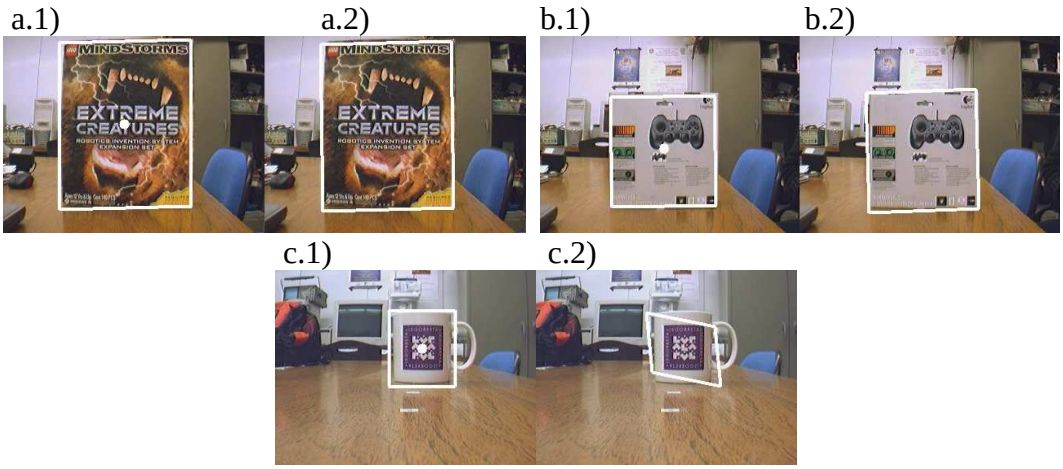

Figura 5.10: Esta figura representa imágenes de muestra de cada uno de los objetos, tomadas de la secuencia 2 durante el seguimiento, en donde el rectángulo blanco representa la homografía calculada. Los incisos a.1), b.1) y c.1) representan la salida del sistema desarrollado en ésta tesis, y los incisos a.2), b.2) y c.2) representan al algoritmo de SURF

5.2.2. Invariancia a la rotación

En esta prueba durante el video, los objetos se rotaron por intervalos de 5° determinando la invariancia a la rotación, en el apéndice B se pueden observar

imágenes muestra de los resultados. En la figura 5.11 inciso *b)*, se observa que el objeto 1 se comporta mejor con la aproximación de SURF, al cambio en la rotación del objeto, es decir, el número de puntos detectados a lo largo de la secuencia es mejor. También se puede observar del inciso *a)* que la tasa de cuadros es mejor para nuestro sistema.

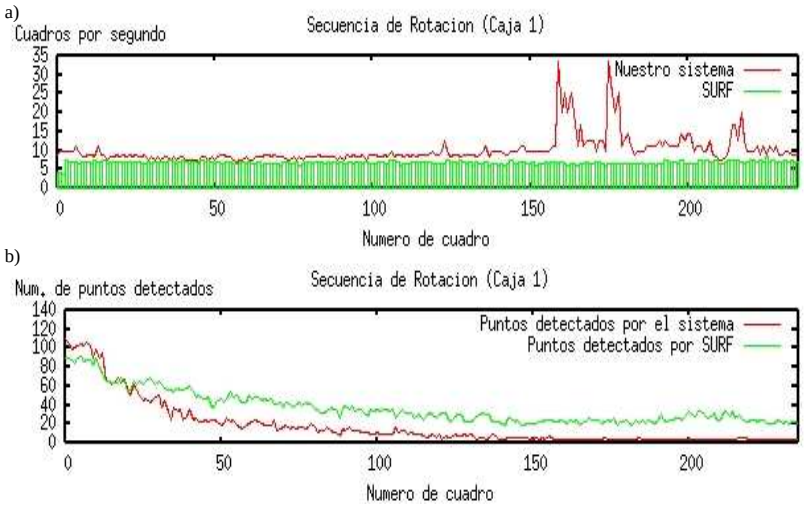

Figura 5.11: Desempeño del objeto 1 con respecto a la rotación del objeto 2. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

En la tabla 5.1 se hace una indicación de los cuadros correspondientes a cada ángulo donde el objeto 1 se encuentra en la secuencia.

Ángulo	**Cuadros**	**Ángulo**	**Cuadros**
0	0 - 11	50	124 - 132
5	15 - 23	55	134 - 142
10	27 - 32	60	145 - 154
15	36 - 41	65	159 - 167
20	43 - 51	70	172 - 180
25	58 - 65	75	185 - 192
30	70 - 80	80	197 - 202
35	84 - 93	85	205 - 210
40	98 - 107	90	212 - 238
45	112 - 120		

Cuadro 5.1: Relación de ángulo - numero de cuadros del objeto 1

Contrario a la figura anterior, en la figura 5.12 se observa al objeto 2 con un mejor desempeño por sistema desarrollado, con respecto al algoritmo de SURF. Primeramente, en el inciso *a)* la tasa de cuadros obtenido por el sistema a lo largo

de toda la secuencia es mucho mejor, con una media de $\mu = 19.92\ cps$, comparado con $\mu_{SURF} = 8.97\ cps$. En el inciso *b)* se tiene una media de $\mu = 9.00$ y $\mu = 12.55$ con respecto a los puntos encontrados a partir de 39 puntos detectados por el sistema y 41 por el algoritmo de SURF en el primer cuadro.

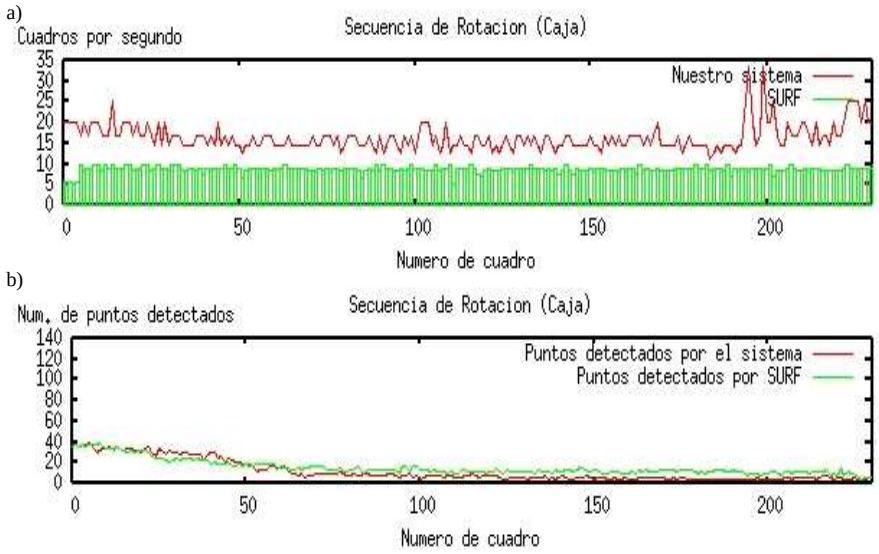

Figura 5.12: Desempeño del sistema con respecto a la rotación del objeto 2. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

La tabla 5.2 describe una relación entre los cuadros correspondientes a cada ángulo probado con el objeto 2.

Ángulo	**Cuadros**	**Ángulo**	**Cuadros**
0	0 - 25	50	199 - 209
5	33 - 42	55	213 - 221
10	51 - 61	60	228 - 236
15	67 - 79	65	240 - 249
20	89 - 99	70	255 - 265
25	110 - 120	75	271 - 280
30	127 - 139	80	286 - 296
35	145 - 154	85	304 - 314
40	162 - 173	90	320 - 365
45	180 - 190		

Cuadro 5.2: Relación de ángulo - numero de cuadros del objeto 2

El la figura 5.13 se muestra el desempeño del objeto 3 con respecto a la rotación. Se observa una buena tasa de cuadros (cps) por parte del sistema en el inciso *a)*, pero es notorio de la gráfica del inciso *b)* que el algoritmo de SURF se desempeña

mejor a medida en que se inclina más el objeto, a pesar de que después de los 65 grados la homografía computada es muy mala debido a la poca cantidad de puntos encontrados correctamente.

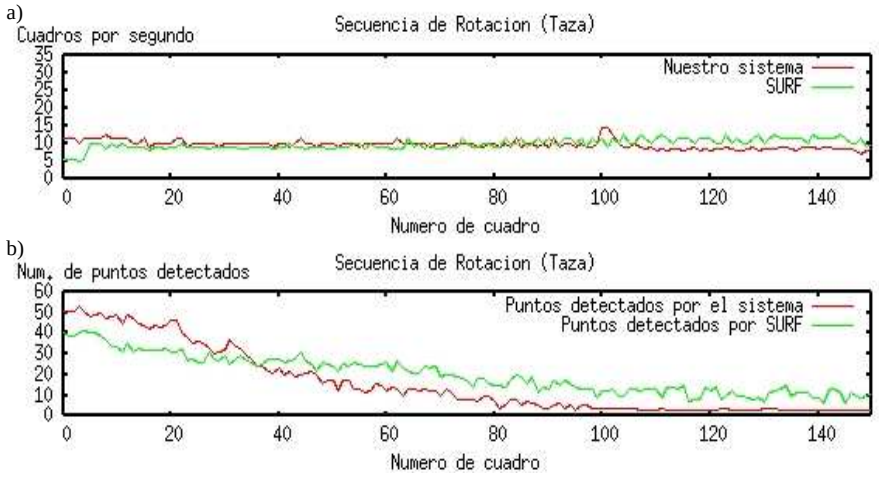

Figura 5.13: Desempeño del sistema con respecto a la rotación del objeto 3. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

La tabla 5.3 describe una relación entre los cuadros correspondientes a cada ángulo probado con el objeto 3.

Ángulo	**Cuadros**	**Ángulo**	**Cuadros**
0	0 - 10	50	125 - 131
5	18 - 24	55	136 - 143
10	29 - 36	60	147 - 152
15	41 - 48	65	156 - 160
20	53 - 60		
25	66 - 74		
30	79 - 85		
35	90 - 98		
40	101 - 109		
45	114 - 120		

Cuadro 5.3: Relación de ángulo - numero de cuadros del objeto 2

La figura 5.14 representa imágenes de muestra tomadas de cada uno de los objetos durante la secuencia de rotación, para tener una mejor visualización de algunas imágenes de prueba, referirse al apéndice B.

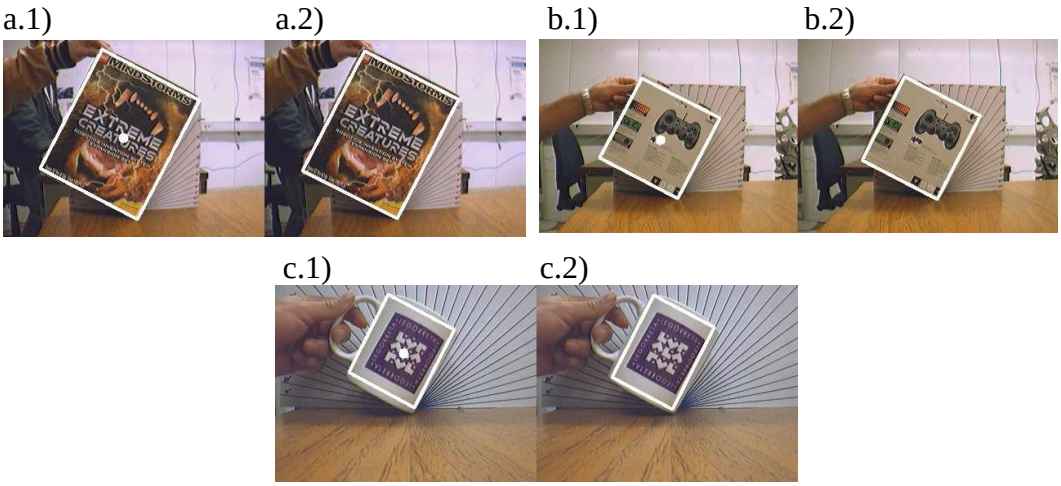

Figura 5.14: Esta figura representa imágenes de muestra de cada uno de los objetos, tomadas de la secuencia de rotación a 30 grados, en donde el rectángulo blanco representa la homografía calculada. Los incisos a.1), b.1) y c.1) representan la salida del sistema desarrollado en ésta tesis, y los incisos a.2), b.2) y c.2) representan al algoritmo de SURF

5.2.3. Seguimiento de objetos a través de secuencias aleatorias

En esta sección se presenta el desempeño del sistema a través del seguimiento de los 3 objetos presentados anteriormente, a lo largo de secuencias aleatorias, las cuales incluyen cambios de escala, rotación, oclusiones parciales y perdida total del objeto en seguimiento. En el apéndice C se observan imágenes muestra de cada objeto durante la secuencia.

En la figura 5.15 inciso *a)* se analiza el seguimiento del objeto 1, y se puede constatar un mejor desempeño con respecto a la tasa de cuadros, obteniendo una media de $\mu = 11.83\ cps$ contra $\mu_{SURF} = 6.54\ cps$. Con respecto a los puntos encontrados a lo largo de la secuencia, se obtuvo una media de $\mu = 33.8$ y $\mu_{SURF} = 36.6$ a partir de 124 y 118 puntos detectados respectivamente en el primer cuadro.

El desempeño del objeto 2 puede ser observado en la figura 5.16, se sigue observando una mejora con respecto a la velocidad del sistema propuesto en este trabajo, resultando así en una media de $\mu = 12.78cps$ y $\mu_{SURF} = 8.63\ cps$. De 40 puntos detectados por el sistema como caracterización del objeto, se obtuvo una media de $\mu = 12{,}24$ puntos encontrados a lo largo de la secuencia. Con el algoritmo de SURF se caracterizó al objeto con 46 puntos y se obtuvo una media de $\mu_{SURF} = 14{,}89$ puntos encontrados durante la secuencia.

En el último objeto de prueba se puede observar de la figura 5.17 inciso *a)*, un desempeño con respecto a la velocidad de procesamiento casi en tiempo real obteniendo una media de $\mu = 22.57\ cps$ comparado con $\mu_{SURF} = 7.92\ cps$. Con respecto a los puntos encontrados, se obtuvo una media de $\mu = 11.82$ para el sistema, a partir de 66 puntos representando al objeto; comparado con $\mu_{SURF} =$

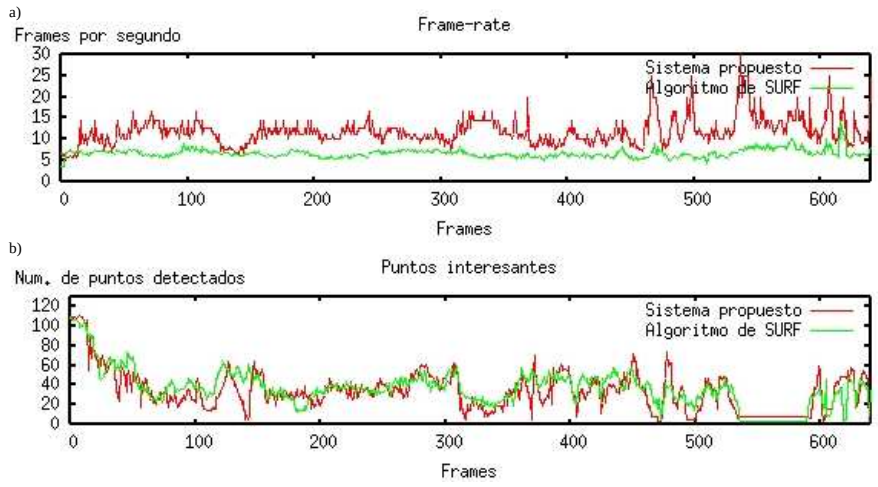

Figura 5.15: Desempeño del sistema con respecto a una secuencia aleatoria con el objeto 1. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

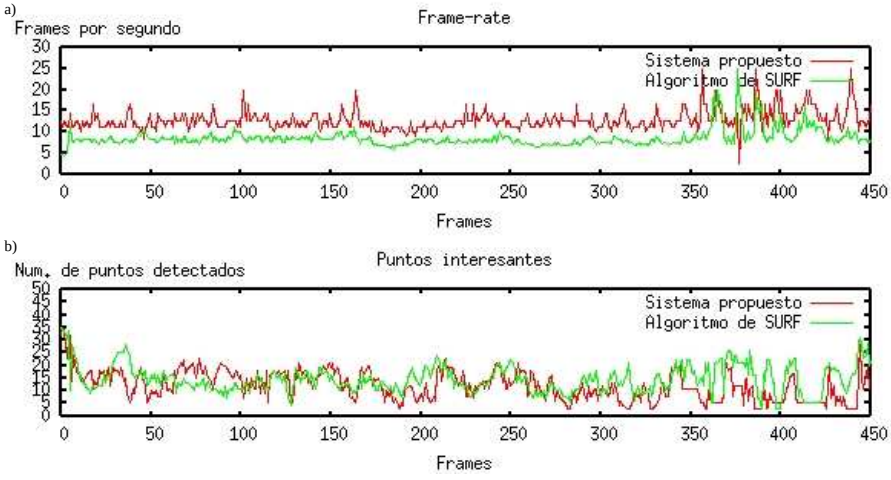

Figura 5.16: Desempeño del sistema con respecto a una secuencia aleatoria con el objeto 2. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

12.83 para el algoritmo de SURF, a partir de 59 puntos representando al objeto.

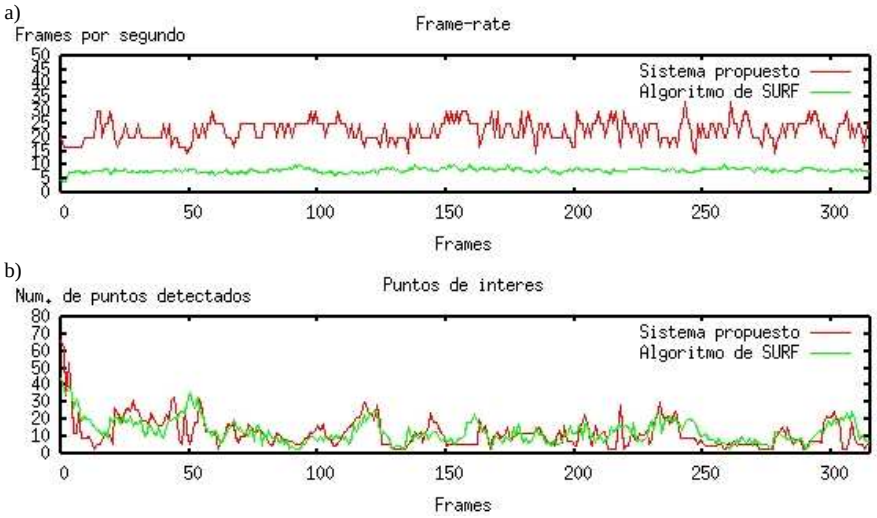

Figura 5.17: Desempeño del sistema con respecto a una secuencia aleatoria con el objeto 3. La gráfica de la parte superior muestra la tasa de cuadros (cps), la gráfica de la parte inferior muestra los puntos detectados

Capítulo 6

Conclusiones

En el capítulo anterior se realizaron una serie de experimentos que permitieron probar el desempeño del sistema desarrollado bajo diferentes situaciones las cuales incluyeron secuencias con cambios de escala, rotaciones de los objetos y secuencias aleatorias con oclusiones parciales y totales, logrando así una demostración detallada del desempeño del sistema.

Los experimentos mostraron que el algoritmo propuesto para este sistema seguidor de objetos resultó ser robusto a oclusiones parciales, escala y rotaciones de aproximadamente 0 a 60 grados para diferentes objetos, permitiendo así un seguimiento del objeto bajo movimientos normales y sin ningún modelo previo del movimiento.

Para poder lograr esto, el objeto primeramente es representado por puntos interesantes, estas características dotaron al objeto de una representación distintiva, lo que permitió diferenciarlos uno de otro. El uso de estos puntos añade al objeto una representación local ya que para poder describirlo, sólo se toman muestras alrededor de su vecindad, haciendo que el objeto pueda ser identificado aun cuando sólo se encuentren pocos puntos de aquellos que representan originalmente al objeto, lo cual hace que el objeto pueda ser ocluido parcialmente sin dejar de reconocer al objeto. El hecho de que no sea necesario segmentar la imagen añade más velocidad al algoritmo.

La descripción de puntos usando el descriptor de SURF demostró ser una idea eficaz aun y cuando el descriptor no estaba diseñado para el detector usado en este sistema (detector de puntos interesantes de Harris). El uso de los puntos interesantes de Harris descritos con SURF no había sido probado anteriormente en la literatura y resulto ser una manera eficaz para la descripción del objeto.

Uno de los inconvenientes del uso del detector de Harris dentro de un seguidor de objetos es que no detecta puntos a diferentes escalas. Esto no fue una limitante ya que se optó por una descripción con múltiples escalas (resoluciones) por cada punto detectado, es decir, el descriptor fue construido sobre diferentes tamaños de la vecindad del punto de interés pero con el mismo número de muestras dependientes de la escala correspondiente. Esta idea resulto en un cómputo más rápido debido principalmente a dos factores: primero, con esta aproximación se evita construcción de una representación espacio-escala en cada cuadro para la detección de puntos en

diferentes escalas; segundo, la construcción de la lista de descriptores multi-escala se hace fuera de línea por lo que durante el seguimiento del objeto, los descriptores son extraídos a una escala fija.

Se puede afirmar que la estimación de la posición del objeto y de la escala del mismo con el uso de un filtro ayuda de gran manera a la velocidad del computo de cada cuadro debido a que la búsqueda del objeto solo se realiza en una porción de la imagen delimitada por la incertidumbre predicha por el filtro. Es cierto que al hacer mas pequeña la región de búsqueda, se puede sacrificar la detección de puntos de interés correctos principalmente cuando el objeto es rotado debido a que las características de la función usada de la librería de openCV que calcula los puntos de interés de Harris no permite el cálculo en una región rotada. Aun así, los puntos encontrados correctos bastan para una buena detección del objeto.

También es cierto que tener muchos descriptores para un punto podría hacerse más pobre el desempeño del sistema provocando un decremento en la velocidad del algoritmo. Con la ayuda del filtro arrojando una predicción de la escala del objeto así como una incertidumbre en ésta, hace que la búsqueda de posibles candidatos se limite hacia sólo una pequeña parte de la lista de descriptores reduciendo así el costo computacional del algoritmo

Se concluye que el sistema desarrollado alcanzó los objetivos propuestos, en donde las aportaciones principales fueron las siguientes: 1) el uso de puntos de interés extraídos con el detector de Harris y descritos con el algoritmo de SURF, 2) el modo en el que se logra la invariancia a escala, para lo cual, los puntos que representan al objeto durante el primer cuadro son descritos a múltiples escalas y 3) el uso del UKF para predecir la escala y la posición del objeto, logrando reducir el costo computacional de ésta aproximación.

Durante los experimentos se observó que hay momentos en los que el detector de Harris deja de detectar puntos y el detector de SURF sigue encontrando puntos, además de que los puntos detectados por SURF generalmente están distribuidos de una manera más uniforme. Es decir, el algoritmo de SURF tanto detección como descripción es una aproximación bastante buena, y como sus autores mencionan, aproxima o mejora esquemas previamente propuestos con respecto a la repetibilidad, distintividad y robustez.

6.1. Trabajos Futuros

Se propone trabajar en un descriptor más robusto con respecto a la rotación, de los experimentos se observa que el número de puntos detectados a medida que los objetos son rotados decae hasta el momento en que deja de ser reconocido. El mismo comportamiento aplica para el algoritmo de SURF pero se observa un poco mas definido en el sistema desarrollado en ésta tesis.

Se probará al sistema para reconocer diferentes objetos en diferentes ambientes, además de que se incorporaran diferentes métodos como el de cuantización vectorial,

con la posibilidad de encontrar al mejor candidato más rápidamente y así acelerar la velocidad del sistema.

Se harán experimentos con el sistema incorporándolo a un robot móvil y haciendo seguimiento de objetos y personas en un ambiente dinámico. Hasta el momento, sólo se conoce la posición espacial del objeto en 2 dimensiones, para obtener la profundidad a la que éste se encuentra localizado, se propone el uso de estéreo-visión además de el uso de un sensor láser para detectar la distancia a la cual se encuentra el objeto y tener una mayor certidumbre.

Se experimentará tratando de incorporar más características a la representación del objeto, como el color o la textura. Tratando de no degradar el desempeño del sistema.

Apéndice A

Invariancia a Escala

En éste apéndice se muestran imágenes tomadas a diferentes distancias durante el seguimiento de los objetos sobre las secuencias 1 y 2, determinando la calidad de nuestro sistema con respecto a cambios de escala.

Cada cuadro está dividido en 4 imágenes en donde se observa lo siguiente: en las imágenes superiores se coloco el primer cuadro y en las imágenes inferiores la salida de ambos sistemas en un determinado cuadro, en donde la columna izquierda muestra la salida de nuestro sistema y en la derecha la de SURF.

Es posible observar que en algúnos cuadros no aparece la homografía, esto es porque no se encontraron más de 3 puntos que hagan correspondencia con los puntos representativos del objeto.

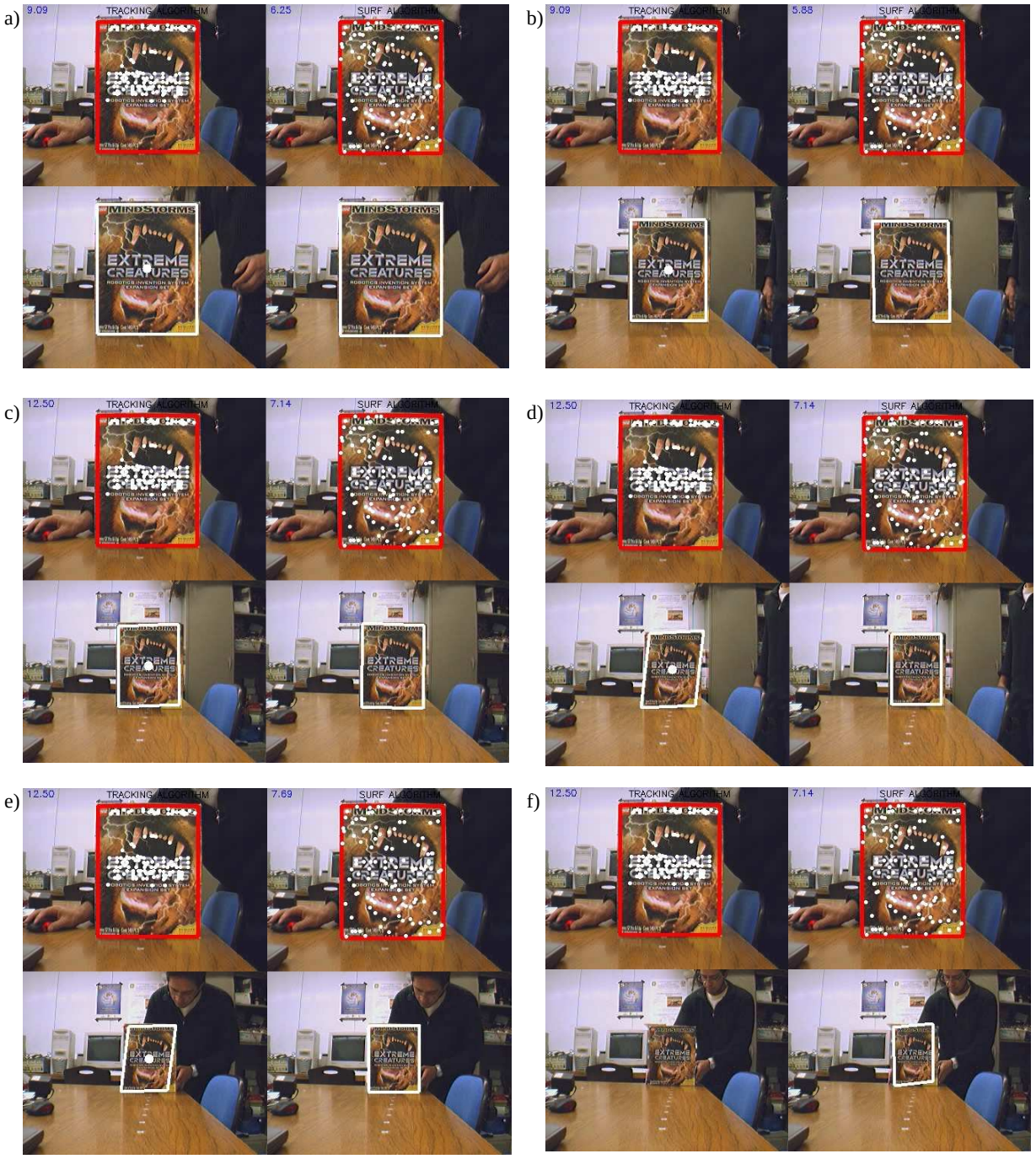

Figura A.1: Invariancia a cambios de escala: caja 1, secuencia 1. En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunas distancias representativas para los experimentos: a)0cm, b)20cm, c)40cm, d)60cm, e)80cm, f) 100cm

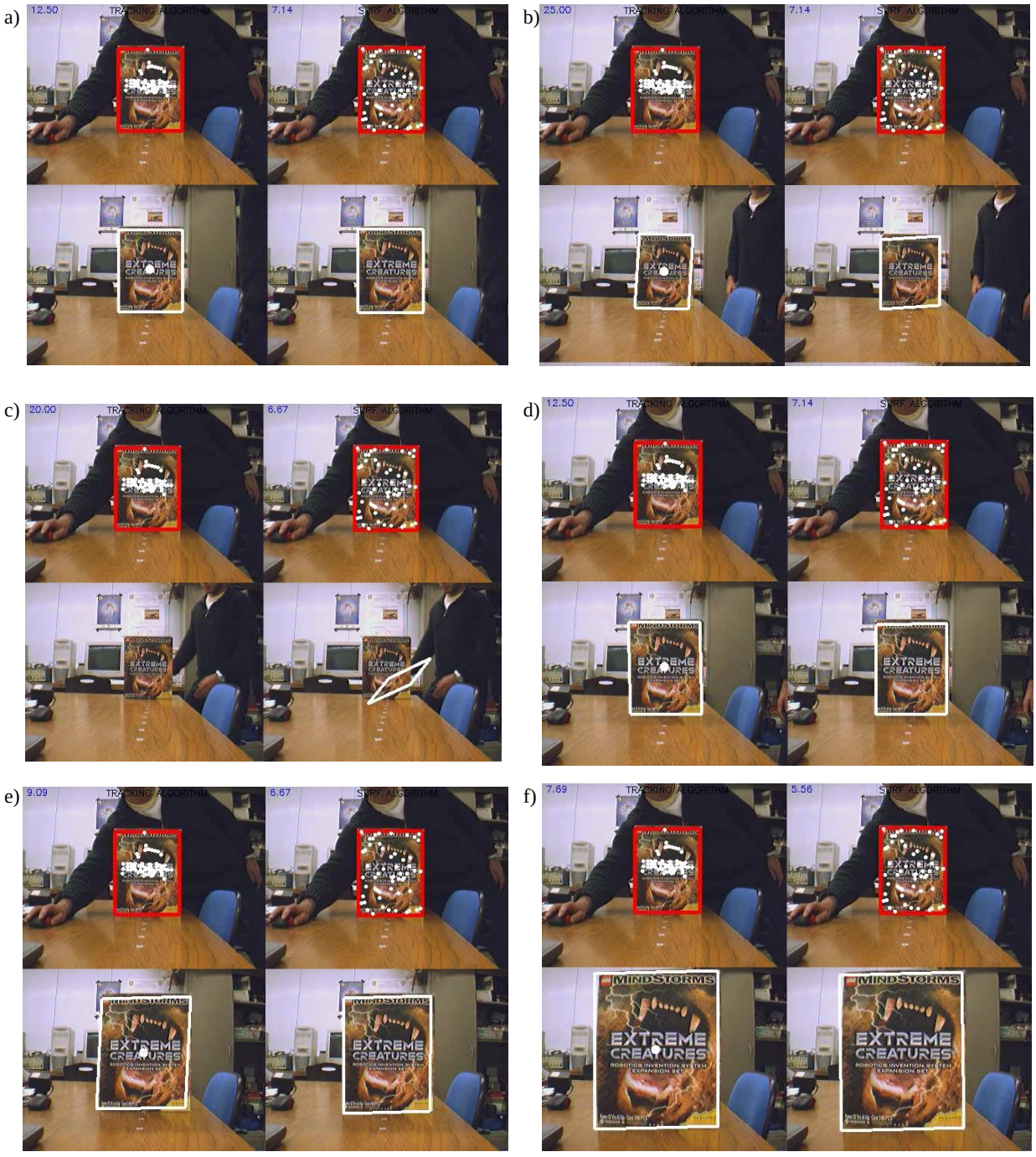

Figura A.2: Invariancia a cambios de escala: caja 1, secuencia 2. En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunas distancias representativas para los experimentos: a)0cm, b)20cm, c)40cm, d)-10cm, e)-30cm, f) -50cm

Figura A.3: Invariancia a cambios de escala: caja 2, secuencia 1.En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunas distancias representativas para los experimentos: a)0cm, b)20cm, c)40cm, d)60cm, e)80cm, f) 100cm

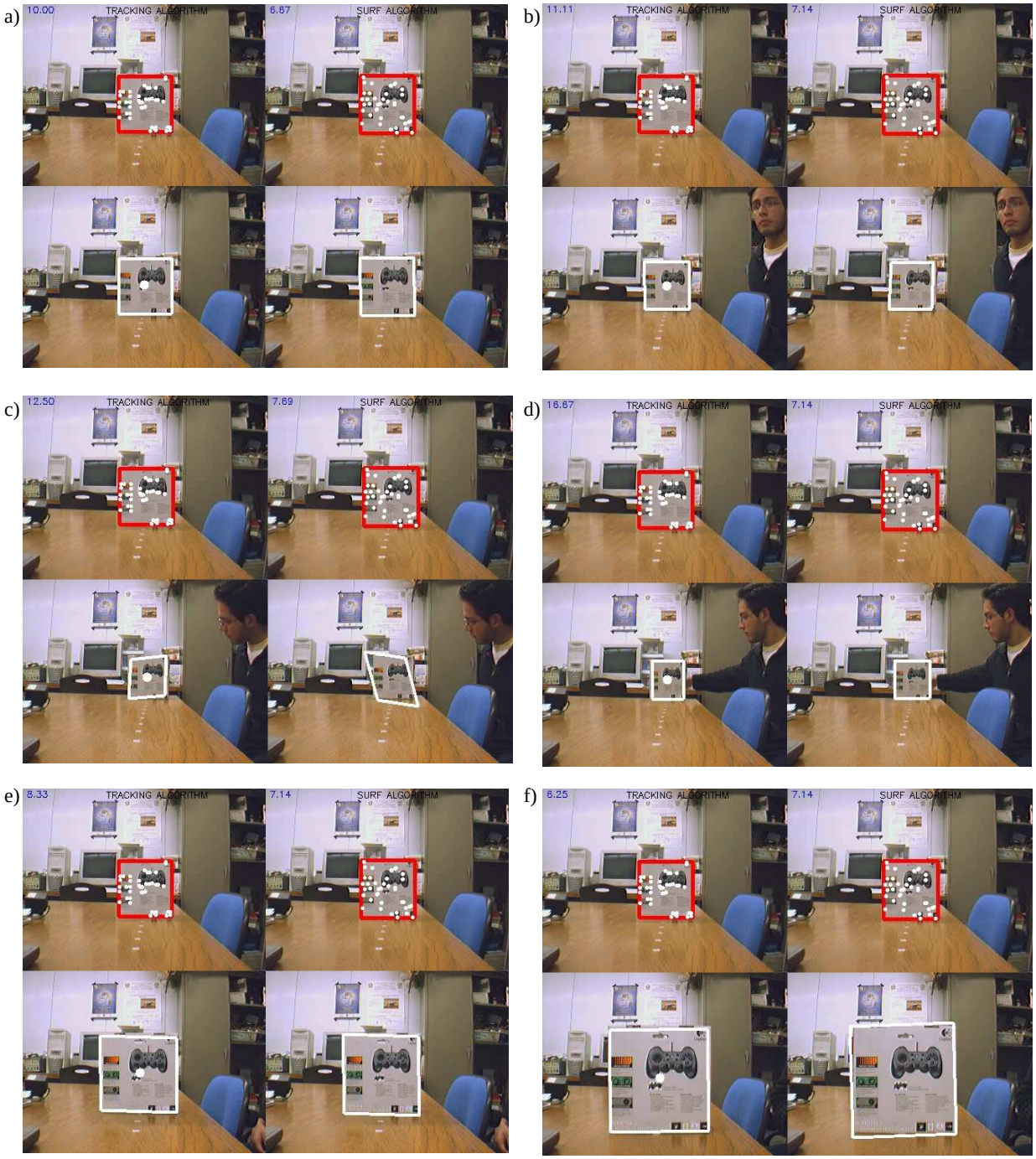

Figura A.4: Invariancia a cambios de escala: caja 2, secuencia 2. En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunas distancias representativas para los experimentos: a)0cm, b)20cm, c)40cm, d)-10cm, e)-30cm, f) -50cm

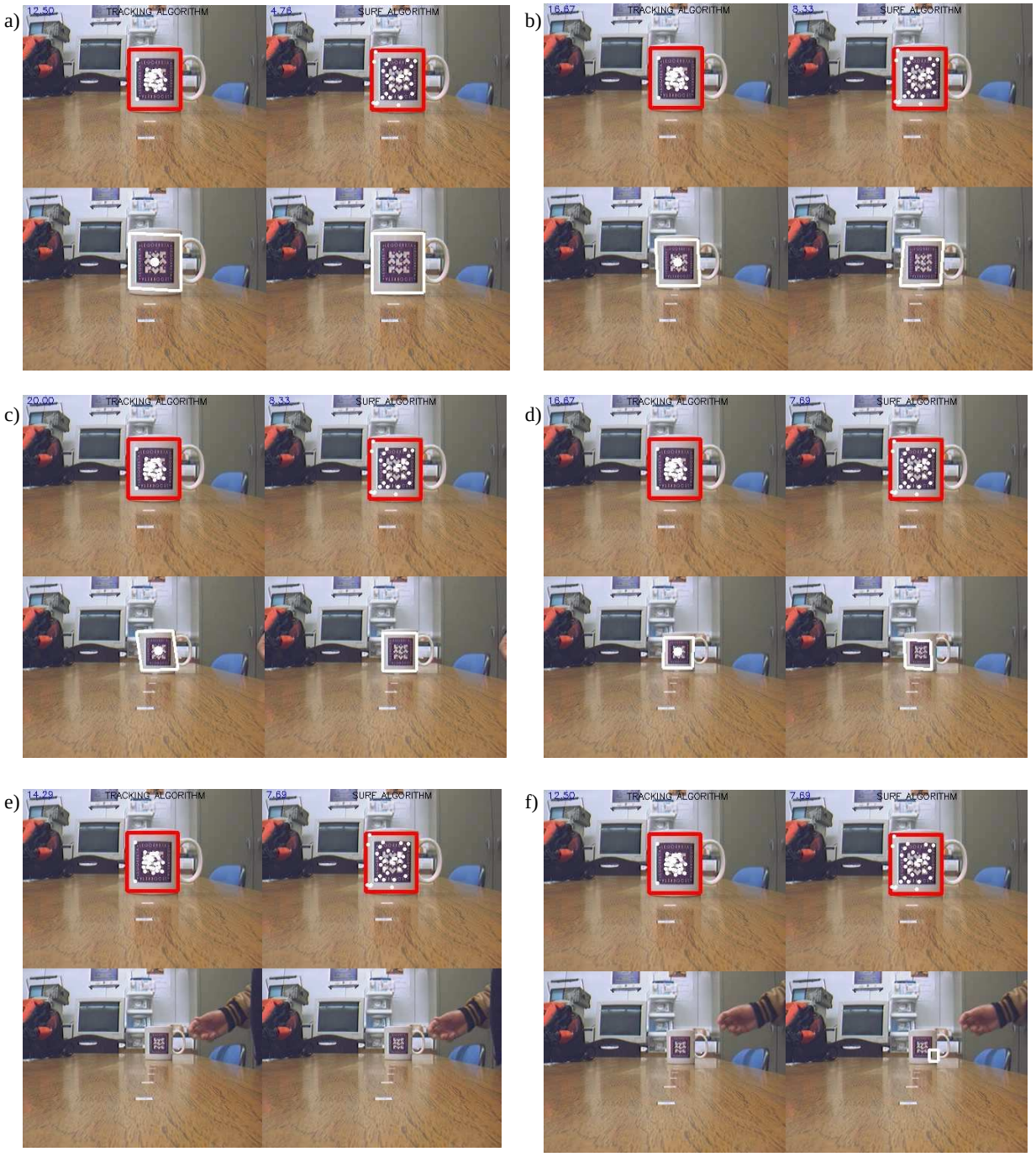

Figura A.5: Invariancia a cambios de escala: caja 2, secuencia 1.En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunas distancias representativas para los experimentos: a)0cm, b)10cm, c)20cm, d)30cm, e)40cm, f) 50cm

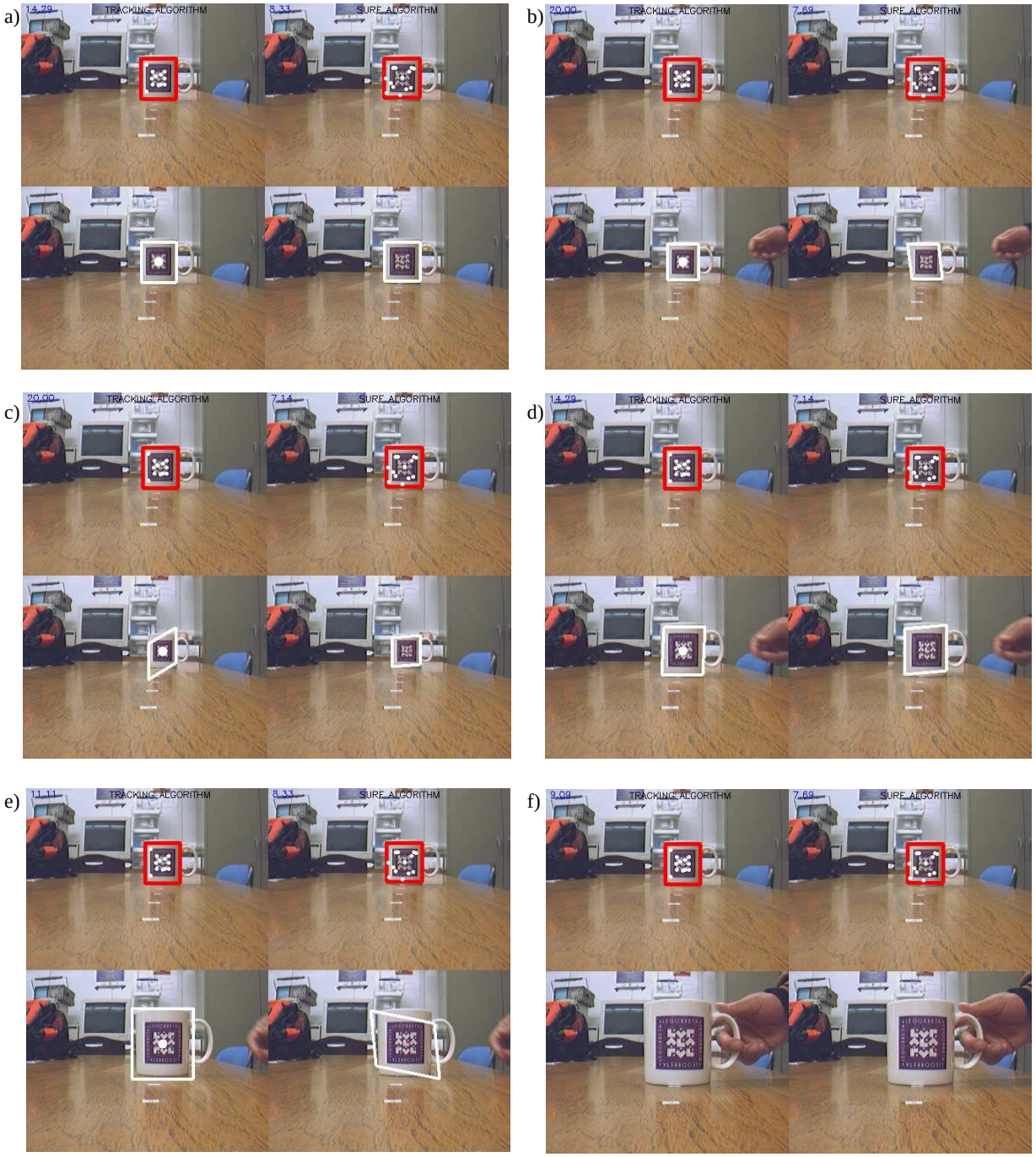

Figura A.6: Invariancia a cambios de escala: caja 2, secuencia 2. En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunas distancias representativas para los experimentos: a)0cm, b)10cm, c)20cm, d)-10cm, e)-20cm, f) -30cm

Apéndice B

Invariancia a rotación

En este apéndice se muestran cuadros tomados a diferentes ángulos durante la secuencia de rotación, para determinar la calidad de nuestro sistema con respecto a cambios en la rotación del objeto.

Cada cuadro está dividido en 4 imágenes en donde se observa lo siguiente: en las imágenes superiores se coloco el primer cuadro y en las imágenes inferiores la salida de ambos sistemas en un determinado cuadro, en donde la columna izquierda muestra la salida de nuestro sistema y en la derecha la de SURF.

Es posible observar que en algúnos cuadros no aparece la homografía, esto es porque no se encontraron más de 3 puntos que hagan correspondencia con los puntos representativos del objeto.

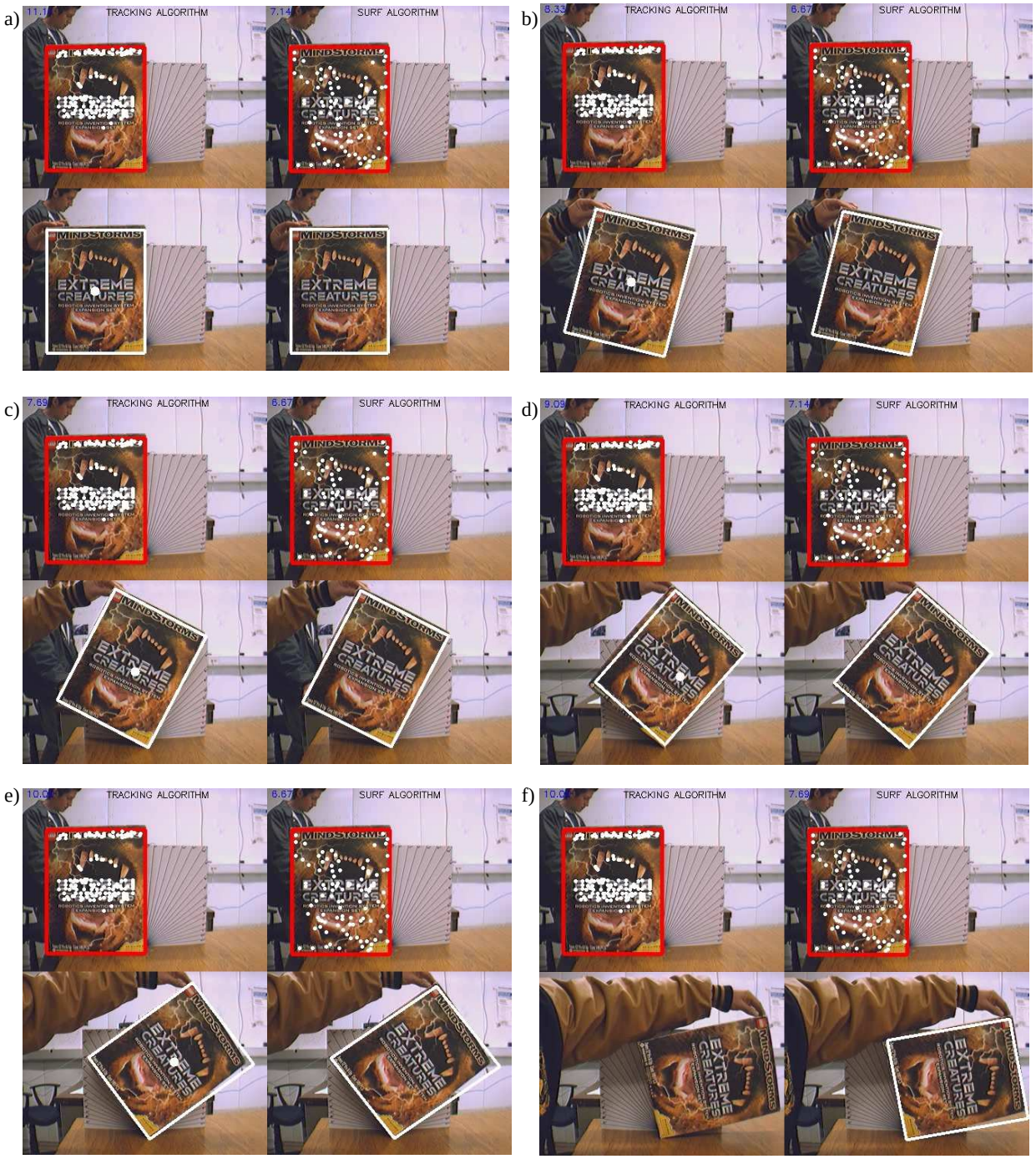

Figura B.1: Invariancia a cambios en rotación: objeto 1. En los incisos: a) - f) se muestran imagenes tomadas del video de prueba en algunos ángulos (grados) representativas para los experimentos: a)0, b)15, c)30, d)45, e)65, f) 80

Figura B.2: Invariancia a cambios en rotación: objeto 2. En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunos ángulos (grados) representativas para los experimentos: a)0, b)15, c)30, d)45, e)65, f) 80

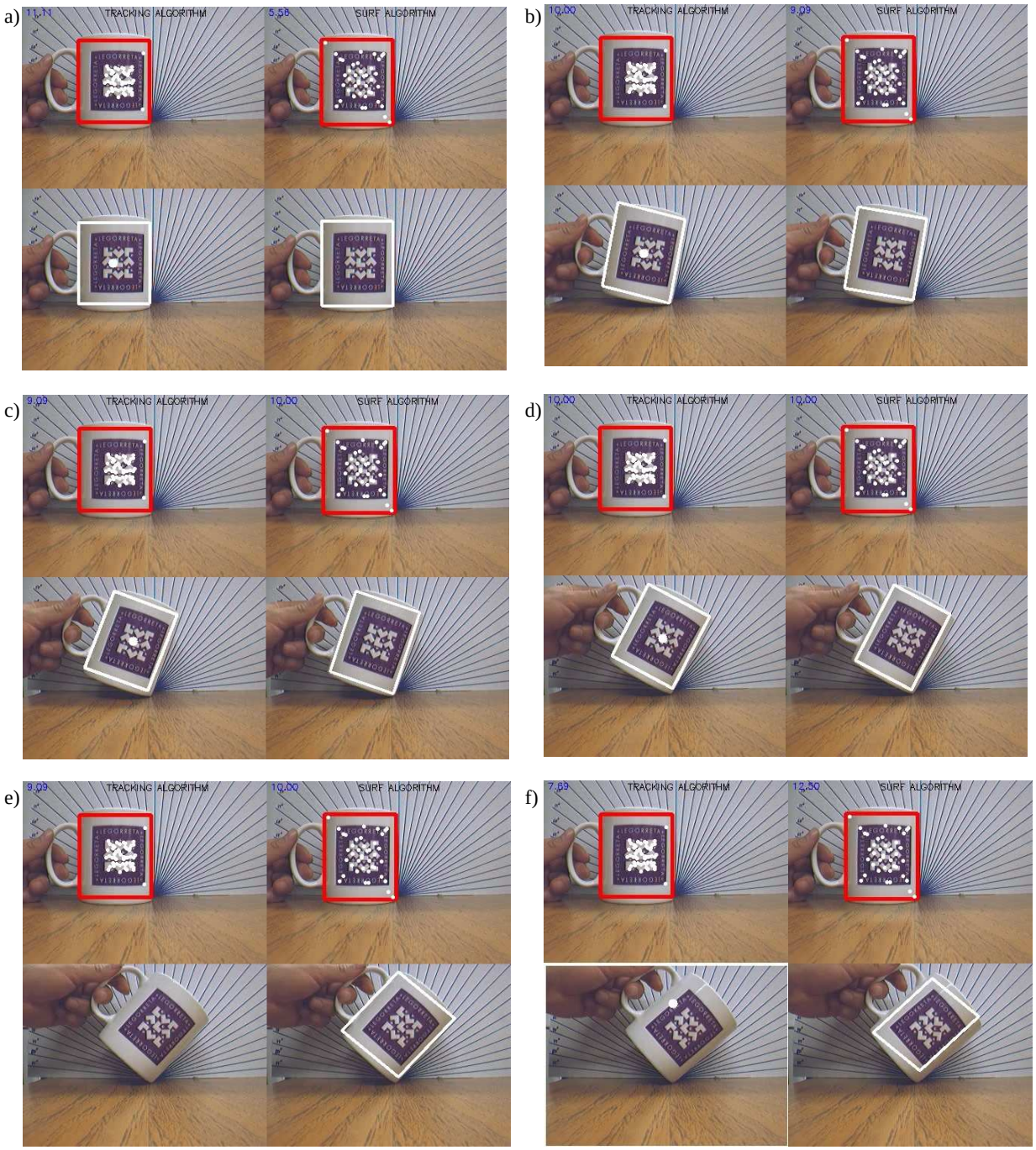

Figura B.3: Invariancia a cambios en rotación: objeto 3. En los incisos: a) - f) se muestran imágenes tomadas del video de prueba en algunos ángulos (grados) representativas para los experimentos: a)0, b)10, c)20, d)30, e)40, f) 50

Apéndice C

Secuencia Aleatoria

En este apéndice se muestran cuadros tomados durante la secuencia aleatoria para los 3 objetos, los cuadros muestran cambios de escala, rotación y oclusiones parciales.

Cada cuadro está dividido en 4 imágenes en donde se observa lo siguiente: en las imágenes superiores se coloco el primer cuadro y en las imágenes inferiores la salida de ambos sistemas en un determinado cuadro, en donde la columna izquierda muestra la salida de nuestro sistema y en la derecha la de SURF.

Es posible observar que en algúnos cuadros no aparece la homografía, esto es porque no se encontraron más de 3 puntos que hagan correspondencia con los puntos representativos del objeto.

Figura C.1: Secuencia aleatoria: objeto 1. En los incisos: a) - f) se muestran cuadros tomados del video de prueba: a)29, b)110, c)188, d)223, e)255, f) 406

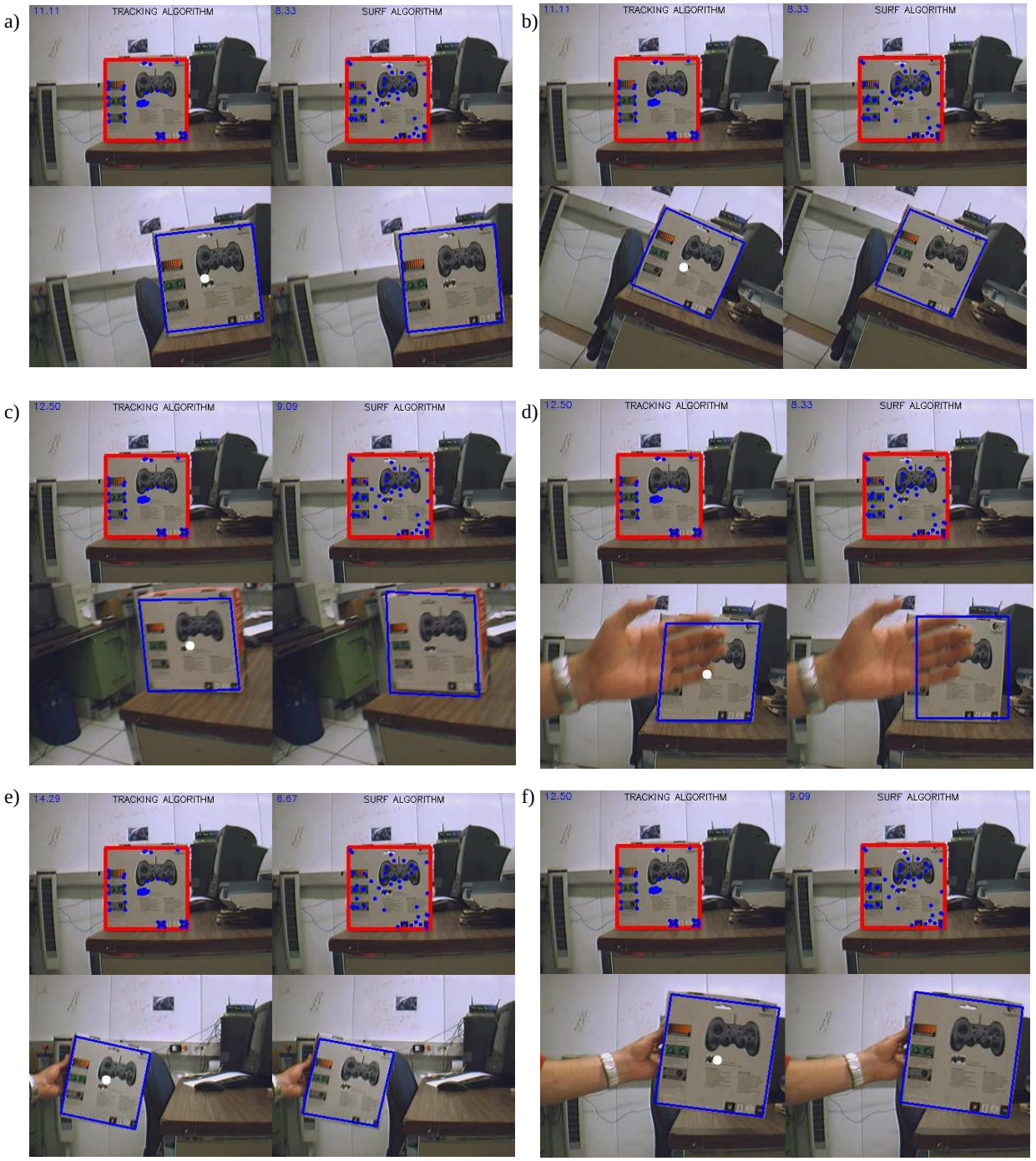

Figura C.2: Secuencia aleatoria: objeto 2. En los incisos: a) - f) se muestran cuadros tomados del video de prueba: a)43, b)53, c)96, d)129, e)279, f) 309

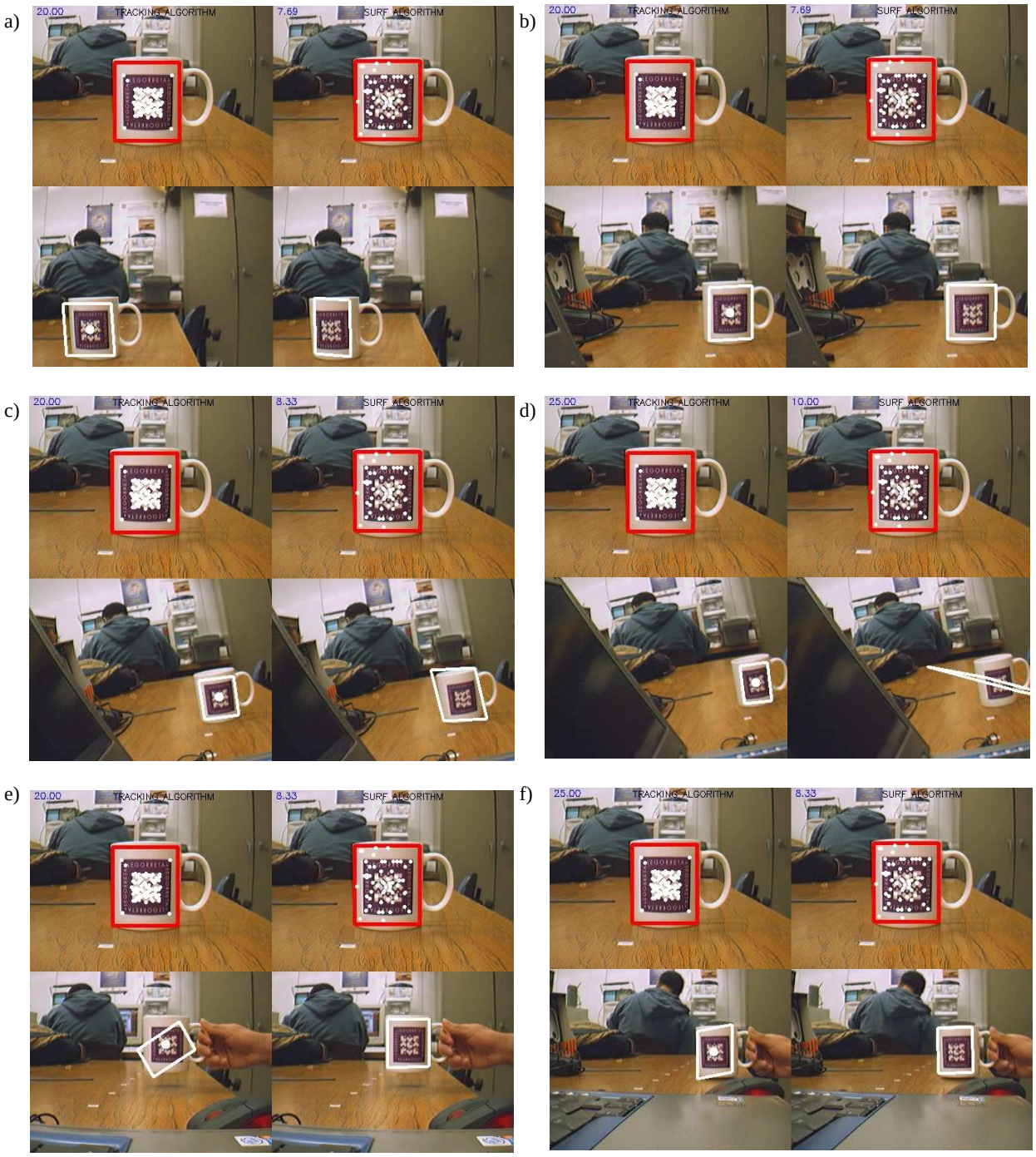

Figura C.3: Secuencia aleatoria: objeto 3. En los incisos: a) - f) se muestran cuadros tomados del video de prueba: a)17, b)29, c)88, d)94, e)126, f) 221

Apéndice D

Algoritmo

En este apéndice se muestra el diagrama a bloques detallado del algoritmo desarrollado para nuestro sistema

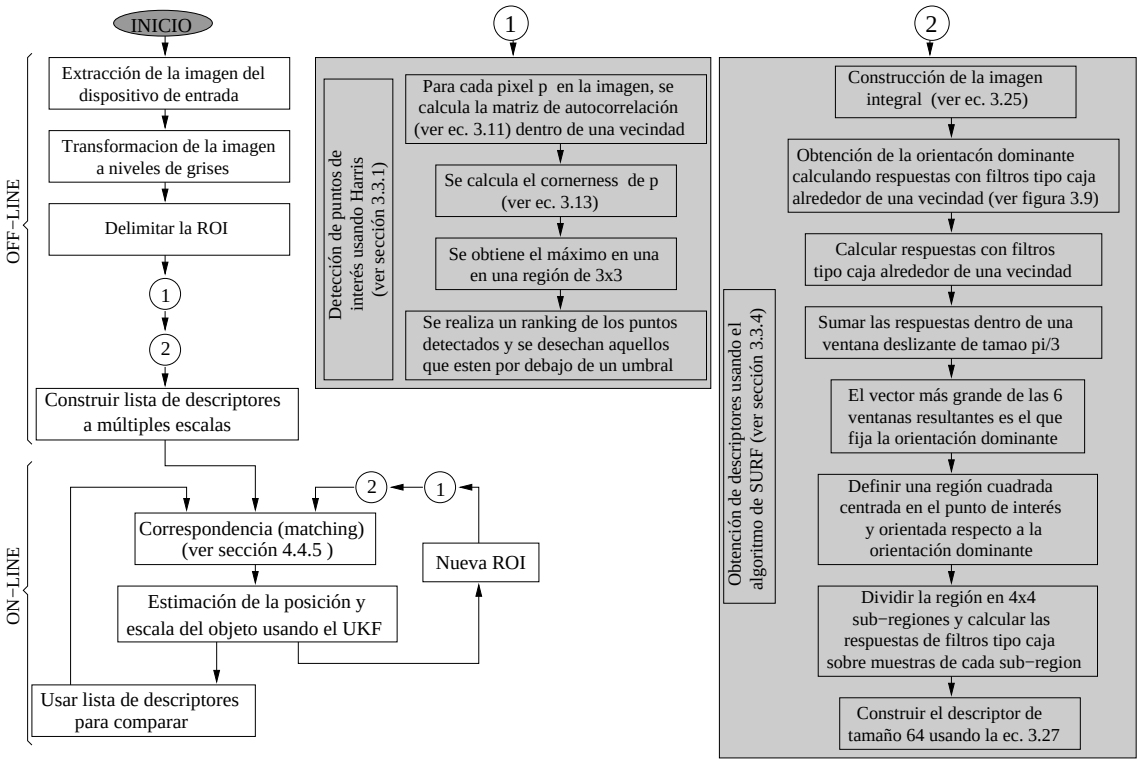

Figura D.1: Diagrama a bloques detallado del sistema

Bibliografía

[1] State space. http://en.wikipedia.org/wiki/State_space_(controls).

[2] U. of bristol, real time vision group. http://www.cs.bris.ac.uk/Research/Vision/Realtime/.

[3] Bakowski A and Jones G. A. Video surveillance tracking using colour region adjacency graphs. In *Seventh International Conference on Image Processing and Its Applications*, volume 2, pages 794 – 798, Jul 1999.

[4] Baumberg A. Reliable feature matching across widely separated views. In *Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[5] Blake A, Isanc M, and Reynard D. Learning to track the visual motion of contours. *Artificial Intelligence*, 78:101–133, 1995.

[6] Blake A and Isard M. *Active Contours*. Springer-London, 1998.

[7] Blake A, Curwen R, and Zisserman A. A framework for spatiotemporal control in the tracking of visual contours. *Int. J. Computer Vision*, 11(2):127–145, 1993.

[8] Witkin A.P. Scale-space filtering. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 1019–1023, 1983.

[9] Toll B. Geometric transformations. http://www.css.taylor.edu/btoll/s99/424/res/mtu/Notes/geometry/geo-tran.htm.

[10] Rasmussen C, Toyama K, and Hager G.D. Traking objects by color alone. Technical report, Yale University, Jun 1996.

[11] Schmid C and Mohr R. Local grayvalue invariants for image retrieval. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 530–534, May 1997.

[12] Schmid C, Mohr R, and Bauckhage C. Evaluation of interest point detectors. *International Journal of Computer Vision*, 2(37):151–172, 2000.

[13] Harris C.J and Stephens M. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conferences*, pages 147–151, 1988.

[14] Chekhlov D, Pupilli M, Mayol-Cuevas W, and Calway A. Real-time and robust monocular slam using predictive multi-resolution descriptors. In *2nd International Symposium on Visual Computing*, Nov 2006.

[15] Comaniciu D, Ramesh V, and Meer P. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, pages 142–149, 2000.

[16] Comaniciu D, Ramesh V, and Meer P. Kernel-based object tracking. In *IEEE Transactions on Pattern Annalysis and Machine Intelligence*, volume 25, pages 564–577, 2003.

[17] Koller D, Weber J, and Malik J. Robust multiple car tracking with occlusion reasoning. In *Proc. of Europe Conference on Computer Vision*, pages 189–196, 1994.

[18] Lowe D. Object recognition from local scale invariant features. In *International Conference on Computer Vision*, 1999.

[19] Lowe D. Distinctive image features from scale-invariant keypoints. *Int, J. Computer Vision*, 2(60):91–110, 2004.

[20] Rembold D, Zimmermann U, Langle T, and Worn H. Detection and handling of moving objects. In *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE*, volume 3, pages 1332–1337, Sep 1998.

[21] Serby D, Meier E.K, and Gool L.V. Probabilistic object tracking using multiple features. In *International Conference on Pattern Recognition (ICPR04)*, pages 184–187, August 2004.

[22] Heitger F, Rosenthaler L, von der Heydt R, Petterhans E, and Kuebler O. Simulation of neural contour mechanism:from simple to end-stopped cells. *Vision Research*, 32(5):963–981, 1992.

[23] Tang F and Tao H. Object tracking with dynamic feature graph. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 25 – 32, 2005.

[24] Welch G and Bishop G. *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill, 2001.

[25] Bay H, Tuytelaars T, and Van Gool L. *SURF*: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.

[26] Moravec H.P. Towards automatic visual obstacule avoidance. In *5th International Joint Conference on artificial Intelligence*, page 584, 1977.

[27] Haritaoglu I, Harwood D, and Davis L.S. W4: Who? when? where? what?. a real time system for detecting and tracking people. In *3rd Int. Conf. on Face and Gesture Recognition*, 1998.

[28] Intel. Computer vision library. http://www.intel.com/technology/computing/opencv/index.htm.

[29] Cottier J. Extraction et appariements robustes des points d'intéret de deux images non étalonnées. Technical report, LIFIA-IMAG-INRIA, Rhone-Alpes, 1994.

[30] Julier J and Uhlmann K. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 93, pages 401–422, 2004.

[31] Noble J.A. Finding corners. *Image and Vision Computing*, 6(2):121–128, May 1988.

[32] Koenderink J.J. The structure of images. *Biological Cybernetics*, 50(5):363–396, 1984.

[33] Derpanis K. The harris corner detector, October 2004.

[34] Mikolajczyk K. *Detection of local features invariant to affine transformations*. PhD thesis, Institut National Polytechnique de Grenoble, Jul 2002.

[35] Mikolajczyk K and Schmid C. Indexing based on scale invariant interest points. In *8th International Conference on Computer Vision*, pages 525–531, 2001.

[36] Pahlavan K and Eklundh J.O. A head-eye system- analysis and design. In *CVGIP: Image Understanding*, volume 56, pages 41–56, Jul 1992.

[37] Brethes L, Menezes P, Lerasle F, and Hayet J. Face tracking and hand gesture recognition form human-robot interaction. In *Proceedings of the International Conference on Robotics and Automation*, pages 1901–1906, April 2004.

[38] Florack L. *The Syntactical Structure of Scalar Images*. PhD thesis, Universiteit Utrecht, 1993.

[39] Kitchen L and Rosenfeld A. Gray-level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[40] Ledwich L and Williams S. Reduced sift features for image retrieval and indoor localization. In *Australian Conference on Robotics and Automation*, 2004.

[41] Silvio Levy. Projective transformations. http://www.geom.uiuc.edu/docs/reference/CRC-formulas/node16.html, Oct 1995.

[42] Bicego M, Lagorio A, Grosso E, and Tistarelli M. On the use of sift features for face authentication. In *2006 Conference on Computer Vision and Pattern Recognition Workshop*, pages 17 – 22, Jun 2006.

[43] Brown M and Lowe D. Invariant features from interest point groups. In *13th British Machine Vision Conference (BMVC2002)*, pages 253–262, 2002.

[44] Fischler M.A and Bolles R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[45] Jacobs O. L .R. *Introduction to Control Theory*. Oxford University Press, 2nd edition, 1993.

[46] Perez P, Hue C, Vermaak J, and Gagnet M. Color-based probabilistic tracking. In *ECCV*, pages 661–675, 2002.

[47] Tissainayagam P and Suter D. Object tracking in image sequences using point features. In *Pattern Recognition*, volume 38, pages 105–113, 2005.

[48] Viola P and Jones M. Rapid object detection using boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[49] Burt P. J and Adelson E.H. The laplacian pyramid as a compact image code. In *IEEE Transactions on Communications*, volume 9, pages 532–540, Jul 1983.

[50] Heckbert Paul S. Fundamentals of texture mapping and image warping. Master's thesis, University of California, Berkeley. Dept. of Electrical Engineering and Computer Science, Jun 1989.

[51] Maybeck Peter S. *Stochastic models, estimation and control*, volume 1. Academic Press, 1979.

[52] Beaudet P.R. Rotationally invariant image operators. In *4th International Joint Conference on Pattern Recognition*, pages 579–583, 1978.

[53] Cipolla R and Yamamoto M. Stereoscopic tracking of bodies in motion. *Image and Vision Computing*, 8(1):85–90, 1990.

[54] Horaud R, Skordas T, and Veillon F. Finding geometric and relational structures in an image. In *1st European Conference on Computer Vision*, pages 374–384, 1990.

[55] Rosales R and Sclaroff S. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *CVPR Workshop on the Interpretation of Visual Motion*, 1998.

[56] Brown R.G and Hwang P. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley and Sons, Inc, second edition edition, 1992.

[57] Se S, Lowe D, and Little J. Global localization using distinctive visual features. In *International Conference on Intelligent Robots and Systems*, 2002.

[58] Julier S.J and Uhlmann J.K. A new extension of the kalman filter to nonlinear systems. In *AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[59] Lindeberg T. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis ans Machine Intelligence*, 12(1):234–254, 1990.

[60] Lindeberg T. On the axiomatic foundations of linear scale-space:combinig semi-group structure with causality vs. scale invariance. Technical report, KTH, Stockholm, Sweden, Aug 1994.

[61] Lindeberg T. *Scale-Space Theory in Computer Vision*. Kluwer Academic, 1994.

[62] Lindeberg T. Linear spatio-temporal scale space. In *1st International Conference on Scale-Space Theory in Computer Vision*. Springer Verlag, New York, Jul 1997.

[63] Lindeberg T. Feature detection with automatic scale selection. *International Journal on Computer Vision*, 30(2):79–116, 1998.

[64] Moller T, Machiraju R, Muller K, and Yagel R. Evaluation and design of filters using a taylor series expansion. In *IEEE Transactions on Visualization and Computer Graphics*, volume 3, pages 184–199, 1997.

[65] Tuytelars T. Tutorial on local invariant features. In *9th European Conference on Computer Vision, ECCV*, May 2006.

[66] Forstner W. A framework for low level feature extraction. In *3rd European Conference on Computer Vision*, pages 383–394, 1994.

[67] Computer Vision Laboratory. ETH Zurich. Surf. http://www.vision.ee.ethz.ch/surf/download.html.