Universidad Rey Juan Carlos

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso Académico 2009/2010

Proyecto de Fin de Carrera

Scanner-Cam

— Autor —

Juan Rinconada Pérez

—Tutores—

Juan José Pantrigo Fernández

Antonio Sanz Montemayor

6 de septiembre de 2010

Agradecimientos

Gracias a mis tutores Antonio Sanz y Juan José Pantrigo por toda la ayuda y consejo prestados durante el desarrollo del proyecto.

A mis padres por siempre animarme a seguir estudiando.

A mi abuelo, que me ha dado alojamiento, apoyo y batallitas sin pensárselo dos veces.

A mis compañeros por aguantarme en las largas tardes en los laboratorios, especialmente Juan.

Y por último, a mis amigos por distraerme.

Resumen

Los nuevos avances en la tecnología nos permiten disponer de cámaras en cada vez más situaciones, por lo que no es de extrañar que queramos potenciar esta tecnología añadiendo nuevas funcionalidades a este dispositivo.

En este proyecto se aborda la posibilidad de utilizar una cámara como si fuera un escáner. Un escáner recoge información parcial de un documento y la integra en una única imagen. Este sistema permite aprovechar al máximo la resolución disponible.

La idea principal es recorrer el documento moviendo la cámara por encima, pudiendo acercarse para aumentar la resolución de cada zona parcial y colocarla en la posición correspondiente de la imagen escaneada. La dificultad fundamental es la auto-localización de la cámara respecto al documento. Mediante técnicas de visión artificial se consigue la detección y seguimiento del documento permitiendo su posterior escaneo.

Se ha conseguido escanear un documento tolerando cambios en la posición y distancia relativas de la cámara al documento y obteniendo una imagen final con múltiples resoluciones según el acercamiento a cada área escaneada, mejorando en consecuencia lo que obtendríamos con una sola imagen de todo el documento.

Índice general

Agradecimientos i

Resumen iii

1. Introducción 1

1.1. Descripción del problema 1

1.2. Herramientas conceptuales 2

1.2.1. Representación de una secuencia de vídeo 2

1.2.2. Umbralización de imágenes 2

1.2.3. Ajuste de Patrones 4

1.2.4. Filtro de partículas 4

1.3. Herramientas utilizadas 9

1.3.1. Visual Studio 2008 9

1.3.2. OpenCV 10

1.3.3. Macromedia Flash 8 10

1.4. Estado del arte 10

1.4.1. Descripción del problema 10

1.4.2. Auto-localización usando un filtro de partículas 11

2. Objetivo 13

2.1. Descripción del objetivo principal 13

2.2. Objetivos parciales 13

2.3. Etapas de desarrollo 14

3. Descripción Informática 17

3.1. Requisitos 17

3.1.1. Requisitos funcionales 17

3.1.2. Requisitos no funcionales 17

3.2. Descripción de la aplicación 18

3.2.1. Obtención y preparación de imágenes 18

3.2.2. Inicialización y actualización del modelo 18
3.2.3. Filtro de partículas 24
3.2.4. Escaneo 27

4. Resultados Experimentales **31**
4.1. Resultados 31
4.1.1. Documento 1 31
4.1.2. Documento 2 33
4.1.3. Documento 3 38
4.1.4. Documento 4 40
4.1.5. Documento 5 48
4.2. Análisis de los resultados 55

5. Conclusiones y trabajos futuros **57**
5.1. Conclusiones 57
5.2. Trabajos futuros 58

Anexo 1: Funcionamiento de la aplicación **59**

Bibliografía **61**

Índice de figuras

1.1. Secuencia de vídeo 2
1.2. Imagen normal e imagen umbralizada 3
1.3. Cálculo de pesos 6
1.4. Selección de pesos acumulados 8
1.5. Diagrama del filtro de partículas 9

3.1. Esquema del modelo 19
3.2. Diagrama UML del modelo 19
3.3. Plantillas de las esquinas 20
3.4. Ejemplo de detección de esquinas errónea y correcta 20
3.5. Usando malla y usando Good Features to track 21
3.6. Muestras a escala 1.0 y a escala 1.7 22
3.7. Sustitución de características perdidas 23
3.8. Diagrama del filtro implementado 24
3.9. Tabla de porcentaje de pesos 26
3.10. Nubes de partículas con grados de incertidumbre de 0.0, 0.2 y 0.7 27
3.11. Esquema del proceso de escaneo 29

4.1. Captura 1 del documento 1 31
4.2. Captura 2 del documento 1 32
4.3. Captura 3 del documento 1 32
4.4. Captura 4 del documento 1 33
4.5. Captura 1 del documento 2 34
4.6. Captura 2 del documento 2 34
4.7. Captura 3 del documento 2 35
4.8. Captura 4 del documento 2 36
4.9. Captura 5 del documento 2 37
4.10. Capturas 1 y 2 del documento 3 38
4.11. Capturas 3 y 4 del documento 3 39

4.12. Captura 1 del documento 4 40
4.13. Captura 2 del documento 4 41
4.14. Captura 3 del documento 4 42
4.15. Captura 4 del documento 4 43
4.16. Captura 5 del documento 4 44
4.17. Captura 6 del documento 4 45
4.18. Escaneo 1 al inicio y al fin 46
4.19. Detalle de escaneo 1 47
4.20. Capturas 1 y 2 del documento 5 48
4.21. Captura 3 del documento 5 49
4.22. Captura 3 del documento 5 50
4.23. Captura 4 del documento 5 51
4.24. Escaneo 2 al inicio y al fin 52
4.25. Detalle de escaneo 2 53
4.26. Captura de documento 5 con 100 partículas 54

Capítulo 1

Introducción

1.1. Descripción del problema

La visión artificial es una disciplina cuyo propósito es la extracción de información del mundo a partir de imágenes procesadas por un computador [JV03].

Para realizar esta tarea el computador dispone de información representada en forma de imágenes digitales que proporcionan datos acerca del color, luminosidad o estructura del mundo real.

Normalmente se adquieren las imágenes mediante un dispositivo óptico como una cámara de vídeo. Luego se aplican una serie de operaciones y filtros a la imagen. Después se aíslan las características interesantes de la imagen y por último mediante un análisis de estas características se sacan unas conclusiones y se realiza una clasificación de los objetos segmentados.

El objetivo de este proyecto es valerse de los métodos de la visión artificial para a partir de imágenes de una secuencia de vídeo detectar y escanear un documento. Una vez localizado se extraen e integran zonas parciales del documento en una imagen final de mayor calidad, en función de la distancia de la cámara al documento.

Para alcanzar este objetivo será necesario conocer las técnicas de visión artificial necesarias para la detección y seguimiento de objetos y familiarizarse con las herramientas de programación que hacen posible su implementación.

1.2. Herramientas conceptuales

1.2.1. Representación de una secuencia de vídeo

Una secuencia de vídeo es una sucesión de imágenes llamadas fotogramas, donde cada uno representa la escena en un instante determinado. Cada fotograma consta de un conjunto de píxeles situados en forma de matriz, donde el número de columnas de píxeles corresponde al ancho de la imagen y el número de filas corresponde al alto. Los valores de los píxeles representan el color y la intensidad de la luz en un punto determinado de la imagen.

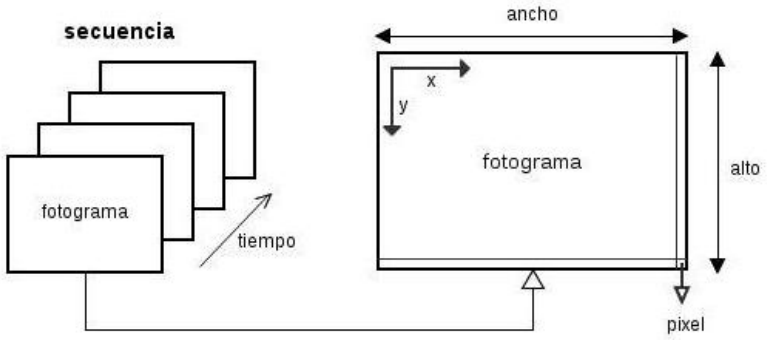

Figura 1.1: Secuencia de vídeo

El fotograma de un instante determinado nos muestra el estado de la escena que variará dependiendo de cambios en la posición de los objetos de en la escena, la posición de la cámara y la intensidad de la luz. En el caso de este proyecto se centra la atención en los movimientos de la cámara.

1.2.2. Umbralización de imágenes

Una imagen en escala de grises tiene una nivel de intensidad para cada uno e sus píxeles comprendido en un rango de valores. Mediante la umbralización se obtiene una imagen con sólo dos valores posibles de intensidad, por ejemplo 0 y 1. La binarización puede resultar útil a la hora de realizar operaciones posteriores sobre la imagen reduciendo notablemente el coste computacional.

Usualmente, para aplicar esta técnica se establece un umbral de intensidad y se compara con el valor de intensidad de cada píxel obteniendo una imagen binaria como resultado. A partir de la imagen f, usando un umbral U se obtiene la imagen g, del siquiente modo:

$$g(x,y) = \begin{cases} 1 \text{ si } f(x,y) \geq U \\ 0 \text{ en cualquier otro caso} \end{cases}$$

Determinar un buen valor umbral es un aspecto esencial para un correcto funcionamiento de la umbralización. La luminosidad de la imagen es un parámetro a tener en cuenta para elegir el valor umbral. Lo que significa que un umbral fijo puede producir malos resultados sobre una imagen donde no se conocen las condiciones de antemano o en una secuencia de vídeo donde varía la iluminación.

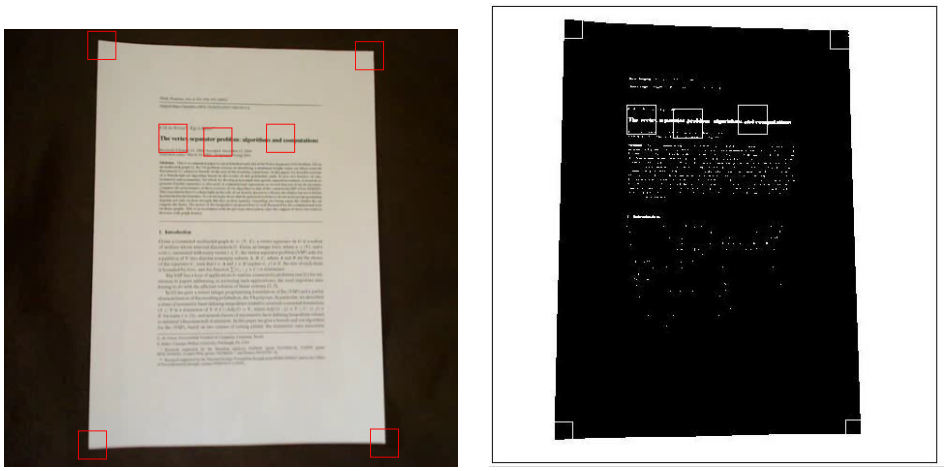

Figura 1.2: Imagen normal e imagen umbralizada

1.2.3. Ajuste de Patrones

Ajuste de Patrones o *Template matching* [Bru09] es una técnica utilizada para encontrar partes de una imagen que encajen con una plantilla. En visión artificial puede usarse como una manera de detectar bordes en una imagen [Yan85]. La idea es recorrer la imagen desde la esquina superior izquierda hasta la inferior derecha comparando píxel a píxel la plantilla con la región de la imagen correspondiente a la posición actual. Ésta, puede realizarse comparando la intensidad de los píxeles usando la suma de diferencias absolutas (SAD). Si llamamos $S(x, y)$ a la imagen y $T(x, y)$ a la plantilla donde (x, y) representan las posiciones de los píxeles de cada imagen y cada píxel tiene una intensidad $I_s(x_s, y_s)$ y $I_t(x_t, y_t)$ respectivamente. La diferencia absoluta de la intensidad de los píxeles se define como:

$$Diff(x_s, y_s, x_t, y_t) = |I_s(x_s, y_s) - I_t(x_t, y_t)|$$

$$SAD(x, y) = \sum_{i=0}^{T_{rows}} \sum_{j=0}^{T_{cols}} Diff(x + i, y + j, i, j)$$

Esta operación compara las intensidades de los píxeles en una imagen parcial de imagen de búsqueda, para adquirir las medidas de toda la imagen se itera transladando el origen de coordenadas de la plantilla a través de toda la imagen de búsqueda. La representación matemática de aplicar la suma de diferencias absolutas a cada píxel es la siguiente:

$$SAD(x, y) = \sum_{i=0}^{S_{rows}} \sum_{j=0}^{S_{cols}}$$

1.2.4. Filtro de partículas

El filtro de partículas es un método empleado para estimar el estado de un sistema que varía a lo largo del tiempo. En concreto se usa en la visión artificial para el seguimiento de objetos en una secuencia. Fue propuesto en 1993 por N. Gordon, D. Salmond y A. Smith [Mac02]. Las partículas son una serie de muestras que tienen asignado un peso.

$$p = \{(x, \pi)|x \in X, p \in [0, 1]\}$$

$$S_n = \{(x_1, p_1)...(x_n, p_n)\}$$

$$\sum_{i=1}^{n} \pi_i = 1$$

Donde x es el estado del espacio de estados X y π es el peso normalizado entre 0 y 1. Usando conjuntos de partículas se aproximan distribuciones de probabilidad. Con un conjunto suficientemente grande obtenemos una función de densidad de probabilidad muy cercana a la real. Se puede demostrar que para un conjunto infinito de muestras la función de densidad de probabilidad estimada coincide exactamente con la real.

Operaciones

Hay tres operaciones que se pueden aplicar sobre un conjunto de partículas: cálculo de pesos, aplicación de un modelo de dinámica y remuestreo [Sán03].

Cálculo de pesos

El cálculo de pesos es la multiplicación del conjunto de partículas por una función consiguiendo mantener la situación de las partículas dentro del espacio de estados pero modificando su peso.

$$S = \{(x_i, \pi_i)_{i=1}^{n}, n = 1, 2, ...\}$$

$$S' = \{(x'_i, \pi'_i)_{i=1}^{n}, n = 1, 2, ...\}$$

$$x'_i = x_i$$

$$\pi'_i = \alpha h(x_i)\pi_i$$

S es un conjunto de partículas y $h(x)$ una función continua, no negativa y acotada lejos de 0. El conjunto resultado tiene la misma disposición que el original pero con pesos proporcionales a la función aplicada.

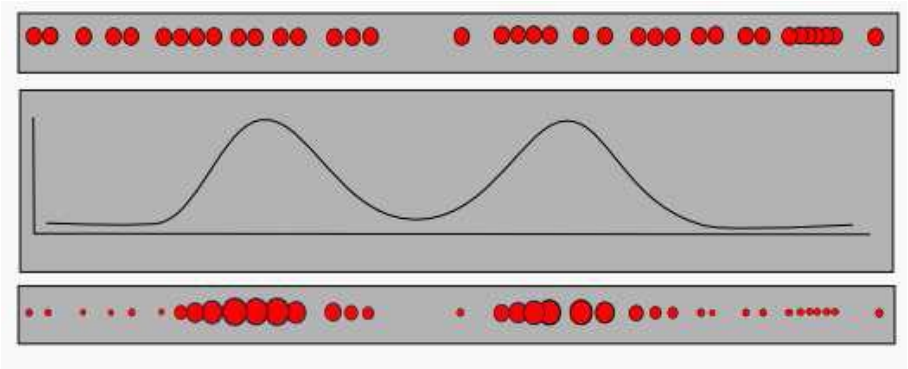

Figura 1.3: Cálculo de pesos

Aplicación de un modelo de dinámica

Con el modelo de dinámica se representa el conocimiento que tenemos del movimiento. Conociendo la posición de un objeto en un instante, se tiene alguna información sobre la posición en el instante siguiente. Esto se expresa como una distribución de probabilidades que aplicamos al conjunto de partículas:

$$S = \{(x_i, \pi_i)_{i=1}^{n}, n = 1, 2, ...\}$$

$$p = (x'|x)$$

$$S' = \{(x'_i, \pi'_i)_{i=1}^{n}, n = 1, 2, ...\}$$

$$x'_i \vdash p(x'_i|x_i)$$

$$\pi'_i = \pi_i$$

Si se conoce la dinámica del sistema, el nuevo conjunto de partículas habrá adaptado su estado a éste. Si se desconoce se puede proponer una distribución uniforme o basándonos

en la hipótesis de continuidad que entiende que la siguiente posición más probable está en las inmediaciones de la anterior, lo que representamos mediante una distribución normal.

Remuestreo

Una distribución de probabilidad se puede representar de varias formas según el uso que ese le de. La etapa de remuestreo persigue este objetivo. Aplicando el remuestreo a un conjunto de partículas S obtenemos:

$$S = \{(x_i, \pi_i)_{i=1}^{n}, n = 1, 2, ...\}$$

$$x_i' = x_j \text{ con probabilidad } \pi_j$$

$$\pi_i' = \frac{1}{n}$$

Donde la operación de selección es independiente en cada caso. Siguiendo un modelo determinista el remuestreo puede ser proporcional a peso de las partículas antiguas.

Descripción del algoritmo

El algoritmo consta de cuatro etapas en las que se usan las operaciones explicadas anteriormente.

- **Inicialización:** Se lanzan N partículas distribuidas aleatoriamente en el espacio de búsqueda.
- **Actualización:** Se define una función que determine proporcionalmente el peso de la partícula de forma que sea más alto si la situación de la partícula es cercana al objeto.
- **Estimación:** Se crea un nuevo conjunto de partículas con el mismo número de partículas que el conjunto anterior pero replicando los pesos proporcionalmente a su peso, obteniendo así un conjunto donde las partículas de pesos altos estarán repetidas más veces que las de pesos bajos.

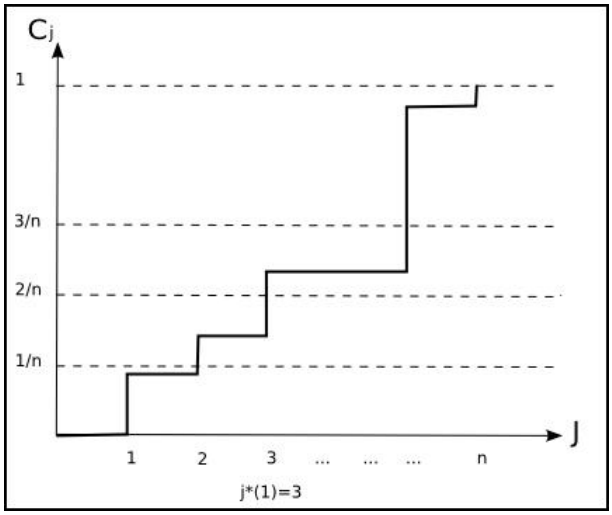

Figura 1.4: Selección de pesos acumulados

- **Predicción:** Siguiendo la hipótesis de continuidad se aplica una difusión de las partículas siguiendo una distribución normal de un radio determinado, adaptando así el sistema a los cambios de estado de las partículas.

A continuación se muestra un diagrama de flujo del funcionamiento del filtro:

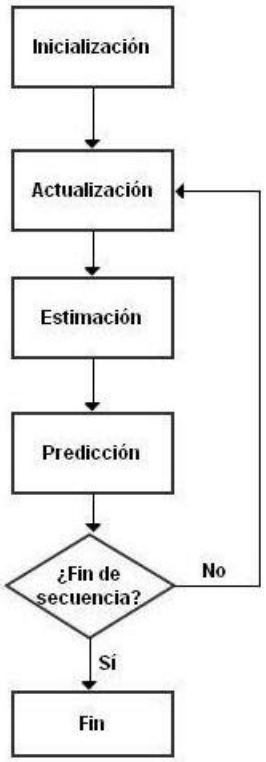

Figura 1.5: Diagrama del filtro de partículas

1.3. Herramientas utilizadas

1.3.1. Visual Studio 2008

Visual Studio es un entorno de desarrollo integrado para Windows con soporte para varios lenguajes como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET.

Visual Studio 2008 tiene una interfaz personalizada para cada uno de los lenguajes y contiene librerías y herramientas como el autocompletado de código, asistentes de configuración y esquemas de organización. También permite el uso de pruebas unitarias y acceso simplificado a bases de datos.

Para este proyecto se usa Visual Studio 2008 como herramienta de desarrollo en lenguaje C/C++.

1.3.2. OpenCV

OpenCV (Open source Computer Vision library) es una librería de código abierto para visión artificial desarrollada por Intel compatible con Linux, Mac OS X y Windows. También puede utilizar el sistema IPP (Integrated Performance Primitives) que aprovecha la optimización de las arquitecturas Intel al trabajar con imágenes a bajo nivel.

OpenCV ofrece un interfaz de alto nivel para las operaciones básicas de manipulación y procesado de imagen y vídeo así como funciones de visión artificial como filtrado, segmentación, análisis estructural, calibración de cámara, análisis de movimiento, reconocimiento de objetos e interfaz gráfica básica.

En el caso concreto de este proyecto interesan especialmente la manipulación de imágenes y vídeo y procesado básico de imágenes.

1.3.3. Macromedia Flash 8

Macromedia Flash es una aplicación multiplataforma usada para crear animaciones, vídeos y aplicaciones interactivas desarrollada por Adobe Systems.

Utiliza gráficos vectoriales, imágenes de mapas de bits, sonido, código de programa, flujo de vídeo y audio bidireccional. Ofrece herramientas avanzadas de dibujo y animación, efectos y codificación de vídeo.

Para este proyecto Macromeria Flash es útil en la creación de vídeos sintéticos para realizar pruebas de rendimiento de la aplicación.

1.4. Estado del arte

1.4.1. Descripción del problema

Para realizar este proyecto hay varios problemas que han de ser abordados, pero hay una diferencia notable en importancia, ya que las dificultades ligadas al escaneo en sí pueden desaparecer implementando un buen algoritmo de seguimiento y el problema de la medida, es un problema general de la visión computacional que tiene que ver con la intensidad de la luz y la calidad de las imágenes devueltas por la cámara. Por lo

tanto el verdadero problema al que hay que enfrentarse es el seguimiento, en concreto el seguimiento de la propia cámara, que también se puede llamar auto-localización de la cámara en un entorno.

1.4.2. Auto-localización usando un filtro de partículas

En los últimos años han aparecido varios algoritmos dirigidos al seguimiento de la posición de una cámara. Para lograr esta tarea se había explorado el mecanismo de "la estructura del movimiento", confiando en una pre-calibración de la escena. Recientemente han aparecido algoritmos más avanzados que determinan la estructura mientras se realiza el seguimiento. El sistema de Davison [Dav03] desarrolla esta idea mediante un filtro de Kalman que estima los parámetros del movimiento con una cámara calibrada y el mantenimiento de características representativas de la escena demostrando un muy buen rendimiento.

La aproximación mediante el filtro de Kalman ha demostrado buen rendimiento si se mantiene una linealidad entre intervalos de tiempo y con unas mediciones que no se desvíen demasiado de un modelo Gaussiano. Desviarse de este modelo acaba desembocando en pérdida del seguimiento que provoca una ambigüedad en la localización de características y muy difícilmente se consigue una recuperación de esta pérdida. El filtro de partículas proporciona una solución mediante una aproximación de muestreo Bayesiana, que lidia con estos problemas de una forma natural y más robusta. Además de ser más flexible respecto a modelo de observación y ser más sencillo de implementar.

Capítulo 2

Objetivo

2.1. Descripción del objetivo principal

El objetivo de este proyecto es desarrollar una aplicación que a partir de una secuencia de vídeo permita detectar y escanear un documento permitiendo movimiento de la cámara sobre este y pudiendo acercar la cámara para aumentar la resolución en zonas parciales que se integrarán en la parte correspondiente del imagen escaneada. Para ello el principal objetivo es resolver el problema de la auto-localización mediante un filtro de partículas.

2.2. Objetivos parciales

- Familiarizarse con el funcionamiento y representación de vídeo y las funciones básicas de la librería OpenCV.
- Entender las técnicas de comparación y filtrado de imágenes.
- Implementar un método de detección del documento y extracción de características para el seguimiento.
- Estudiar el método del filtro de partículas y su aplicación a la auto-localización.
- Desarrollar un filtro de partículas que funcione sobre un vídeo sencillo con movimiento de translación.
- Mejorar el filtro para que funcione con movimiento de acercamiento y alejamiento de la cámara.
- Aplicar cambios en las variables del filtro para obtener los mejores resultados posibles sobre un vídeo concreto.

- Desarrollar un proceso de escaneo que permita actualizar la imagen escaneada según un criterio y la coloque en la posición sobre el escaneo inicial.
- Generar secuencias de vídeo sintéticas y reales para probar la aplicación en diferentes situaciones.
- Estudiar la respuesta del sistema ante las distintas secuencias y experimentar con varios valores para los parámetros.

2.3. Etapas de desarrollo

Análisis

- Entender el problema de detección de características y buscar posibles soluciones.
- Estudiar el problema de la auto-localización y la aproximación a una solución mediante el filtro de partículas investigando trabajos anteriores.
- Decidir los casos de prueba que serán utilizados para probar la aplicación con movimiento de translación, acercamiento y pérdida de características.
- Analizar los resultados y e intentar mejorarlos aplicando otros métodos.

Diseño

- Proponer un esquema de la estructura general de la solución.
- Crear una adaptación del método del filtro de partículas al caso particular de seguimiento del documento.
- Decidir un criterio para la actualización del escaneo automático.

Implementación

- Aplicar las funciones de la librería OpenCV para todas las necesidades en que sea posible.
- Desarrollar un filtro de partículas capaz de seguir al documento.

- Modificar las características y parámetros del filtro para optimizar el funcionamiento en nuestro caso particular.
- Escribir la parte de la aplicación correspondiente al escaneo.

Pruebas

- Crear vídeos sintéticos y reales de situaciones características con distintos documentos.
- Ejecutar la aplicación variando los distintos parámetros.
- Analizar el acierto del filtro de partículas y el rendimiento global de la aplicación.

Capítulo 3

Descripción Informática

3.1. Requisitos

3.1.1. Requisitos funcionales

Los requisitos funcionales son características que definen qué debe realizar el sistema. Estos son los requisitos funcionales del proyecto:

- La aplicación debe ser capaz de detectar un documento dentro de una escena separando este del fondo de la imagen.
- Debe desechar una detección claramente incorrecta de un documento.
- Debe poder seguir un documento cuando el movimiento de la cámara cambie la posición de este en la escena.
- Ha de permitir cambios en la distancia de la cámara al documento.
- Debe recuperarse de algunas pérdidas en el seguimiento.
- Debe disponer de un criterio de auto-escaneo que actualize la imagen escaneada si la escena permite aumentar la resolución del área que se está escaneando actualmente.

3.1.2. Requisitos no funcionales

Los requisitos no funcionales describen cómo debe comportarse la aplicación, especificando las restricciones a las que está sujeta la aplicación para asegurar un correcto funcionamiento. Los requisitos no funcionales son los siguientes:

- Las imágenes deben proporcionarse a la aplicación por medio de una cámara o un vídeo.
- La secuencia de vídeo debe tener una iluminación que se mantenga constante.
- El fondo sobre el que se encuentre el documento debe tener un color oscuro, en el que el documento destaque y no se refleje la luz.

3.2. Descripción de la aplicación

3.2.1. Obtención y preparación de imágenes

Las imágenes se obtienen de un vídeo o una cámara usando de funciones de OpenCV. A cada imagen se le aplica un filtro de color para crear una nueva imagen en escala de grises. Ésta se convierte a binario usando una constante como umbral que puede variarse según la luminosidad de las imágenes. Manejar imágenes en escala de grises y binarias permite que las operaciones posteriores se realicen de forma más eficiente. Para ambas transformaciones se pueden utilizar funciones de OpenCV.

3.2.2. Inicialización y actualización del modelo

Un modelo es una estructura que utilizamos para realizar el cálculo de pesos en el filtro de partículas. Consta de una posición central, que indica su localización en la imagen, un conjunto de muestras, que a su vez tienen una posición relativa a al centro del modelo y una imagen que corresponde a una plantilla para comparar con la imagen real. Además el modelo tiene una escala para poder estimar distancia de la cámara al documento, cuando la cámara se acerque al documento la escala aumenta y al multiplicarse por las coordenadas relativas de las muestras el modelo se hará más grande, al alejarse la cámara, se obtiene el efecto contrario. El modelo también contiene un conjunto de estados que nos indican si una muestra esta fuera o dentro de la imagen actual.

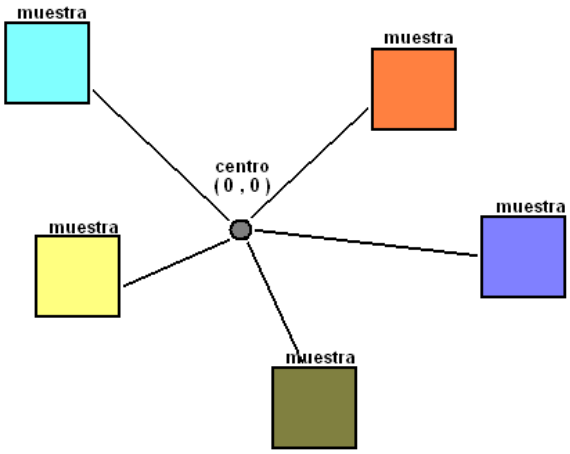

Figura 3.1: Esquema del modelo

El modelo requiere una configuración inicial, la posición central puede ser una posición cualquiera pero parece lógico usar el centro del primer fotograma como referencia. La escala comienza siendo 1.0. Para inicializar las muestras hay que decidir sus posiciones respecto al centro y para decidir cuáles serán estas posiciones y sus imágenes correspondientes primero hay que distinguir entre dos tipos de muestras: las cuatro esquinas del documento y algunas características interiores a éste.

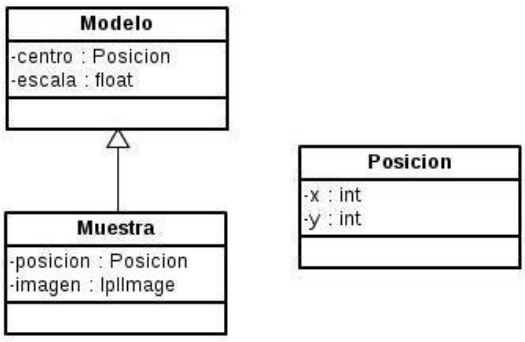

Figura 3.2: Diagrama UML del modelo

A continuación se describen las operaciones que pueden aplicarse al modelo:

- **Detección de esquinas:** Para encontrar las esquinas se utiliza el método de *Template matching* con cuatro imágenes que actúan como plantillas, una para cada esquina. Aplicando *Template matching* sobre el primer fotograma con cada una de las plantillas se obtienen las cuatro posiciones sobre la imagen con mayor probabilidad de ser nuestras esquinas. Como esta operación no asegura que los puntos obtenidos están en posiciones con sentido, una vez obtenidos los puntos se realiza una comprobación para desecharlos en caso de que estos formen una figura que no se corresponda con un rectángulo aproximado. Por ejemplo, si la esquina inferior izquierda está en un punto más alto que la superior izquierda. (Ver figura 3.4)

  Si la detección ha ido bien, la comparación de cada una de las esquinas con el fotograma actual devolverá cuatro puntos en los cuales se ha encontrado la mayor similitud con cada una de las plantillas, un rectángulo alrededor de cada punto delimita la imagen que a partir de ahora se tomará como plantilla, utilizando así imágenes tomadas del primer fotograma de la secuencia en lugar de las plantillas genéricas con las que se empieza.

Figura 3.3: Plantillas de las esquinas

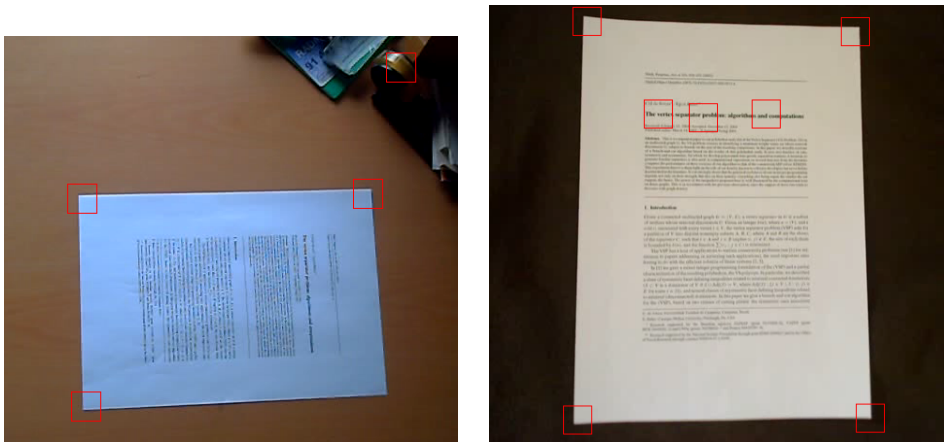

Figura 3.4: Ejemplo de detección de esquinas errónea y correcta

- **Obtención de muestras interiores:** Si la detección de esquinas ha sido satisfactoria se obtiene una región de interés, o sea un trozo de la imagen del primer fotograma delimitada por el rectángulo formado por las posiciones de la esquina superior izquierda y la inferior derecha. Primero se le aplica un margen de seguridad al rectángulo para asegurar que se encuentra en el interior del documento. Para el siguiente paso hay dos opciones, que se pueden ver como dos formas de aproximarse al problema. La opción más simple es usar una malla, o sea obtener las muestras simplemente a partir de posiciones uniformemente repartidas por la zona interior. La segunda opción es usar un método implementado por OpenCV llamado *Good Features To Track* que permite detectar puntos de la imagen que poseen buenas características para su seguimiento. Aunque cualquier enfoque puede ser igual de válido, en un principio la primera opción puede llevar a obtener muestras no muy características.

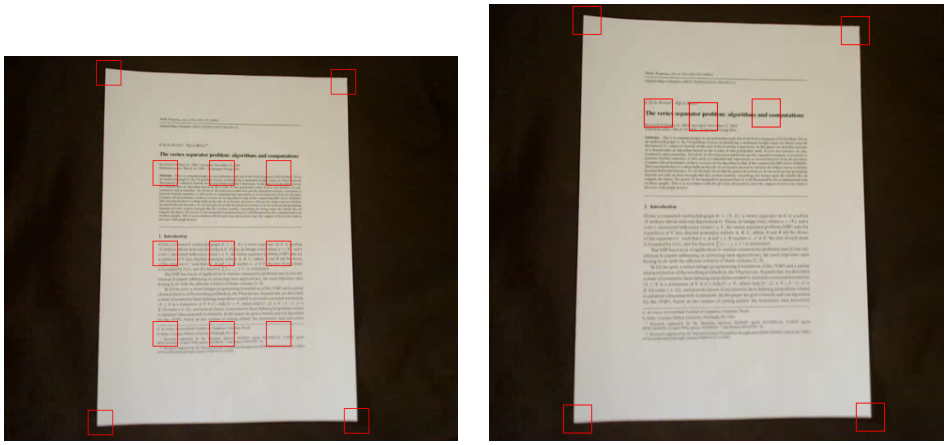

Figura 3.5: Usando malla y usando Good Features to track

La inicialización del modelo proporciona una buena aproximación de lo que es el documento al inicio del vídeo, pero conforme la posición del documento cambie, el modelo debe adaptarse a su nueva apariencia en la imagen. De los cambios en la posición y la escala se encarga el filtro de partículas, ya que utiliza el modelo como herramienta para el cálculo de pesos y por tanto estas características varían de una partícula a otra. En cambio otras características del modelo como el tamaño de las imágenes de las muestras y la pérdida de muestras porque se han salido del fotograma no son responsabilidad del filtro de partículas. Cuando el modelo varía su escala las muestras pueden adaptarse aplicando una redimensión, otra alternativa es la actualización, o sea, cambiar la imagen por la que actualmente ocupa la posición de

la muestra. Si alguna de las muestras se sale del fotograma se realiza una sustitución.

A continuación se describen estos tres métodos:

- **Redimensión de muestras:** Cada vez que el modelo cambie de escala, las imágenes de las muestras dejaran de corresponderse con la realidad, así que se deben adaptar sus dimensiones a las de la nueva apariencia del documento. Para ello se fija un valor umbral que indica si la escala ha variado lo suficiente como para realizar una redimensión. Esta comprobación se realiza fotograma a fotograma y cuando haya cambiado lo suficiente se multiplican las dimensiones iniciales por la nueva escala y se aplica una función de OpenCV que permite la redimensión de imágenes usando distintos métodos de interpolación que pueden variar en eficacia dependiendo del vídeo o las características de las muestras.

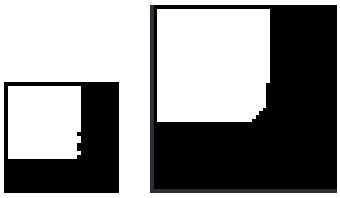

Figura 3.6: Muestras a escala 1.0 y a escala 1.7

- **Sustitución de muestras perdidas:** A veces, alguna de las muestras se sale de las fronteras de la imagen. Cuando esto pasa, se pierde una de las referencias y afecta negativamente al funcionamiento. Para que esto no ocurra existe un método de sustitución de muestras perdidas, que detecta cuando una muestra lleva un tiempo fuera de la imagen y crea una nueva muestra con una posición interior al documento para sustituirla usando *Good Features To Track* para obtener una buena posición. Es importante destacar que una muestra debe permanecer un tiempo fuera de la imagen para considerarse perdida ya que una sustitución precipitada podría crear excesivas muestras nuevas. Para controlar esto, se dispone de un historial de pérdidas que se actualiza y comprueba cada fotograma.

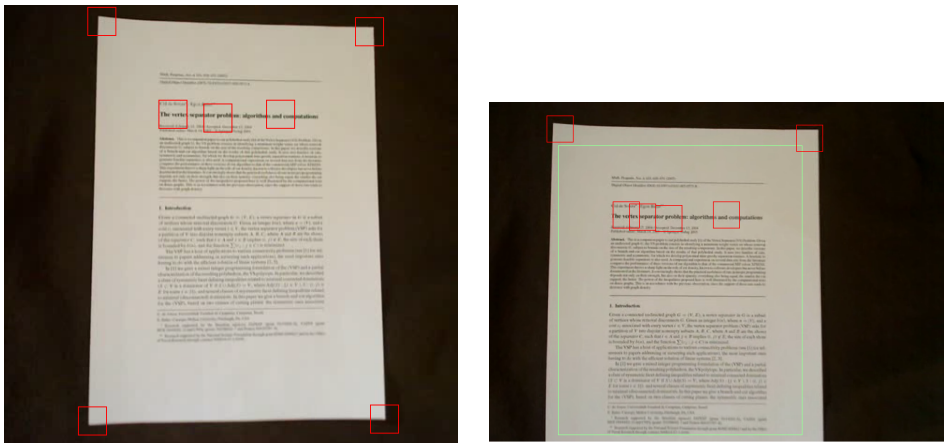

Figura 3.7: Sustitución de características perdidas

- **Actualización de muestras:** Cuando se aplica la redimensión, muchas veces sobre las muestras sus imágenes pueden variar mucho de se forma inicial y dejar de ser buenas plantillas para el cálculo de pesos del filtro de partículas. Esta actualización puede llevar a engaño, pues si el modelo no está bien colocado, la imagen actualizada puede no corresponder con la de la posición en que se encuentra la muestra. Para evitarlo se puede simplemente no realizar la actualización, sólo realizarla cuando la escala cambie un cierto valor o usar un historial para ver si, en el momento actual, el modelo tiene una alta probabilidad de estar bien colocado. Esto se hace almacenando los últimos pesos de las mejores partículas y, si el peso actual es el mejor, se puede decir que hay una alta probabilidad de que el modelo este bien colocado.

3.2.3. Filtro de partículas

El filtro de partículas implementado permite el seguimiento del documento como un objeto de dos dimensiones que puede moverse, cambiar de tamaño y salirse parcialmente de la imagen. Las partículas utilizadas tienen una posición, una escala y un peso. El filtro de partículas consta de cinco etapas: Inicialización, cálculo de pesos, obtención de la mejor partícula, selección y difusión.

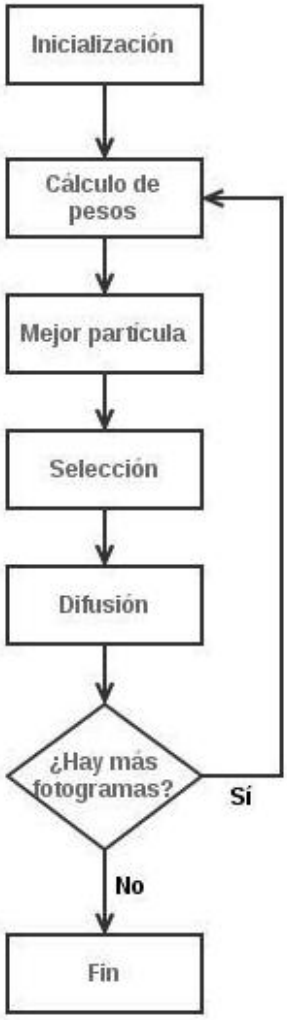

Figura 3.8: Diagrama del filtro implementado

A continuación se describen en detalle cada una de las cinco etapas del filtro:

- **Inicialización:** La inicialización de un filtro de partículas se puede hacer distribuyendo todas las partículas en posiciones aleatorias, pero en nuestro caso, al inicializar el modelo, hemos obtenido la posición más probable para que se encuentre el documento. Por tanto la inicialización se encarga de asignar a cada partícula una posición y escala aleatorias cercanas a esta posición. Los rangos que definen el grado de aleatoriedad son configurables tanto para la escala como para la posición de las partículas.

- **Cálculo de pesos:** En el cálculo de pesos cada partícula utiliza el modelo para obtener su peso en el fotograma actual. Primero se le asigna al modelo el centro y la escala de la partícula, después se comparan una a una las muestras con la imagen del fotograma que se encuentra en la posición correspondiente. Esta comparación devuelve un valor numérico que corresponde al peso de cada muestra. Cada peso se multiplica por un valor que pondera los pesos parciales dependiendo de si son muestras interiores o esquinas. Estos factores de ponderación se pueden configurar para adaptarse al vídeo utilizado. Para calcular la diferencia entre la plantilla de la muestra y la imagen actual del fotograma se aplica un método que actúa de manera distinta para las esquinas y para las muestras interiores.

  En las esquinas sólo interesa tener en cuenta los píxeles de color blanco, que son los que representan al documento, cuando los píxeles blancos de la plantilla coincidan en posición con los píxeles blancos de la imagen esto sumara un 1 por píxel. Por otro lado, en las zonas negras de la plantilla no se sumará nada cuando estos píxeles coincidan con los de la imagen porque forman parte del fondo y no se considera relevante ya que si gran parte de la imagen es negra puede llevar a una potenciación errónea del peso. En cambio sí se restará un 0.25 cuando la posición de un píxel negro de la plantilla corresponda con uno blanco en la imagen, evitando así asignar buenos pesos a imágenes con mucho blanco, por ejemplo en el interior del documento.

  Para las plantillas de las muestras interiores se realiza la operación contraria a la de las esquinas. En este caso sólo interesa sumar peso cuando coincidan los píxeles negros, o sea, los píxeles que forman parte de letras o figuras del documento, que son las zonas que dan características únicas. Así que se sumará 1 cuando el píxel de la plantilla y la imagen sean negros y se restará 0.25 cuando el negro de la plantilla coincida en posición con un blanco en la imagen. Los píxeles blancos de la plantilla

no son relevantes ya que la mayoría de los píxeles de las plantillas y las imágenes que se tomen en el interior del documento serán blancos y por tanto no presentan características discriminatorias entre ellos.

		Porcentaje de similitud		
Pantilla	Imagen	Comparación normal	Sólo suman los blancos	Sólo blancos y penalización
		0 %	0 %	- 75 %
		100 %	100 %	100 %
		25 %	100 %	25 %
		75 %	0 %	0 %

Figura 3.9: Tabla de porcentaje de pesos

- **Obtención de la mejor partícula:** Esta etapa no afecta al resto del funcionamiento del filtro de partículas, así que se puede no considerar como etapa. Consiste en obtener una partícula para tener una buena estimación de donde se encuentra el documento en un momento dado para poder utilizarlo, ya sea dibujar una indicación visual de que el filtro funciona o realizar un escaneo. La estimación de esta partícula se puede hacer de dos maneras: seleccionando simplemente la partícula de mayor peso o calculando la media de las posiciones y escalas de todas las partículas ponderadas por su peso.

- **Selección:** La selección pretende descartar la mayoría de las partículas con malos pesos y potenciar aquellas con buenos pesos. Una forma de hacer la selección es simplemente sustituir todas las partículas por la mejor. Una solución más robusta consiste en utilizar el algoritmo de la ruleta, un sorteo en el que las partículas de mayor peso tienen más probabilidad de ser seleccionadas. La selección implementada usa ambos métodos y recibe una variable que nos proporciona un grado de incertidumbre con respecto a la partícula de mayor peso. Cuanto mayor sea el grado de incertidumbre, menos confianza tenemos en que la partícula de mayor peso debe ser la seleccionada. Esto significa que el grado de incertidumbre nos indica el

porcentaje de partículas que debemos seleccionar usando el algoritmo de la ruleta, el resto se sustituyen directamente por la partícula de mayor peso.

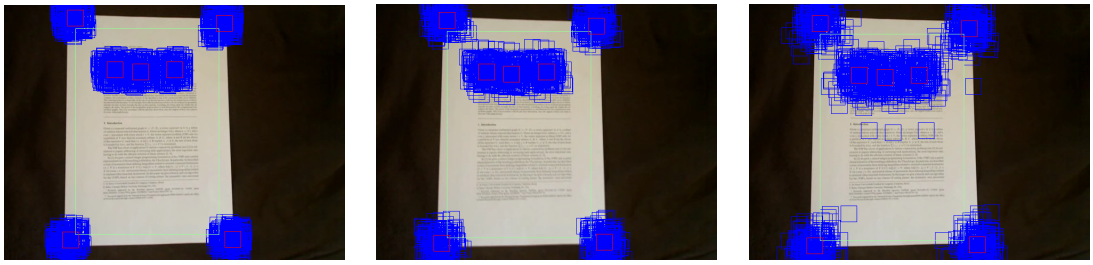

Figura 3.10: Nubes de partículas con grados de incertidumbre de 0.0, 0.2 y 0.7

- **Difusión:** La selección crea una nueva nube de partículas muy condensada en torno a un punto. Ya que el documento puede cambiar de posición y escala en el siguiente fotograma hay que aplicar un ligero movimiento a cada partícula sumando un valor aleatorio a la posición y escala actual. Este valor puede ser aleatorio o ser obtenido mediante una distribución Gaussiana.

3.2.4. Escaneo

El escaneo es la etapa de la aplicación que nos permite obtener los resultados finales. Una vez el filtro de partículas nos proporciona una posición del documento sobre la imagen, el escaneo extrae la zona del documento que es visible en un fotograma dado y la integra en una imagen correspondiente al escaneo total del documento. El escáner tiene tres métodos: inicialización, escaneo y guardado.

- **Inicialización:** La inicialización es un método que debe ser llamado necesariamente antes de empezar a escanear, mientras que los otros dos métodos pueden ser llamados en cualquier momento. Este método se encarga de crear una imagen en blanco de las dimensiones adecuadas, para lo que hay definido un ancho o alto máximo que se adapta a la dimensión más grande del documento. Si el documento que vamos a escanear se encuentra en vertical, la imagen inicial tendrá la altura máxima y un ancho proporcional a esa altura según la forma del documento. Si se encuentra en horizontal, la anchura será el valor máximo y la altura será proporcional. Esto significa que se intenta crear una imagen mayor que la imagen del documento inicial

para que al acercarse la cámara y ampliarse la imagen pueda mejorar la resolución, pero siempre manteniendo las proporciones del documento.

- **Escaneo:** Este método recibe como información la posición del documento en la imagen y el área del documento que está visible en ese momento. Si el área visible es el documento completo, el escaneo consiste en redimensionar la imagen del documento a las dimensiones de la imagen del escaneo que se creó inicialmente y copiar la imagen encima. En caso de no ver nada del documento, no se hará nada en absoluto. Si lo que ocurre es que se ve parte del documento, hay que calcular que parte del documento se está viendo, teniendo en cuenta la posición del área visible respecto al área total del documento. Con la posición que ocupa el área parcial del documento tenemos la posición donde se debe colocar, pero también hay que averiguar la escala, que se obtiene calculando las proporciones entre el área visible y el área total respecto a las dimensiones de la imagen inicial creada. Gracias a esto y a la escala, se coloca la imagen con la posición y tamaño correspondiente.

  El método de escaneo puede usarse de dos formas, puesto que no sólo hace lo descrito anteriormente, también se encarga de decidir si es un buen momento para realizar un escaneo o no. Esta decisión utiliza un criterio para evitar escaneos innecesarios en caso de llamar al método en cada fotograma. Si todo el documento está visible sólo se escaneará si la cámara se ha acercado lo suficiente. Este umbral de zoom se puede variar si se desea que se escanee más o menos a menudo. Si sólo vemos parte del documento, el escaneo sólo se realiza si la cámara se ha movido horizontalmente o lateralmente una distancia determinada. Así, se puede seguir escaneando cuando la cámara se encuentre recorriendo el documento desde cerca sin variar la distancia a éste. El método de escaneo también permite la opción de forzar el escaneo por si se prefiere ignorar el criterio de decisión y escanear en un momento cualquiera. Un caso en el que el método no escaneará, independientemente de si se quiere forzar o no, es cuando la cámara se haya acercado al documento tanto que esté excediendo la resolución máxima permitida por la imagen que se creó inicialmente.

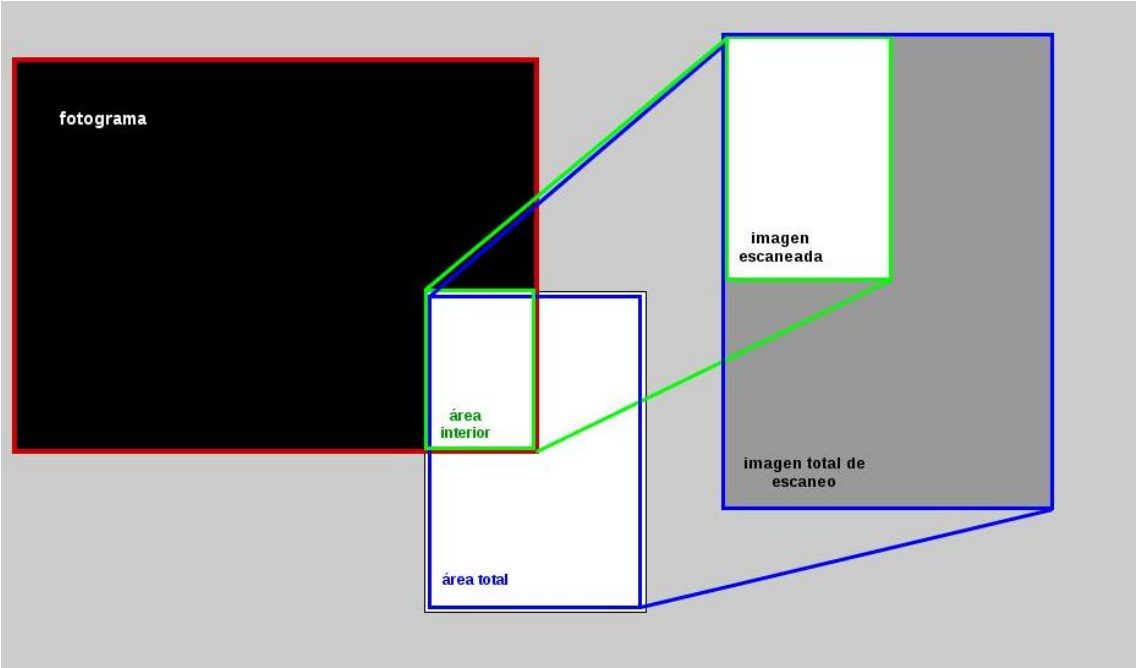

Figura 3.11: Esquema del proceso de escaneo

- **Guardar escaneo:** Independientemente de las veces que se quiera realizar escaneos de la imágenes del vídeo, en cualquier momento se puede llamar al método de guardado especificando una ruta para guardar el fichero de imagen resultante.

Capítulo 4

Resultados Experimentales

4.1. Resultados

4.1.1. Documento 1

Descripción: Pérdida de esquinas inferiores.

Primero se produce una detección de esquinas correcta y tres características interiores, dejando un margen de seguridad obtenemos el área interior al rectángulo verde que determina el área de interés para el escaneo. (Ver figura 4.1)

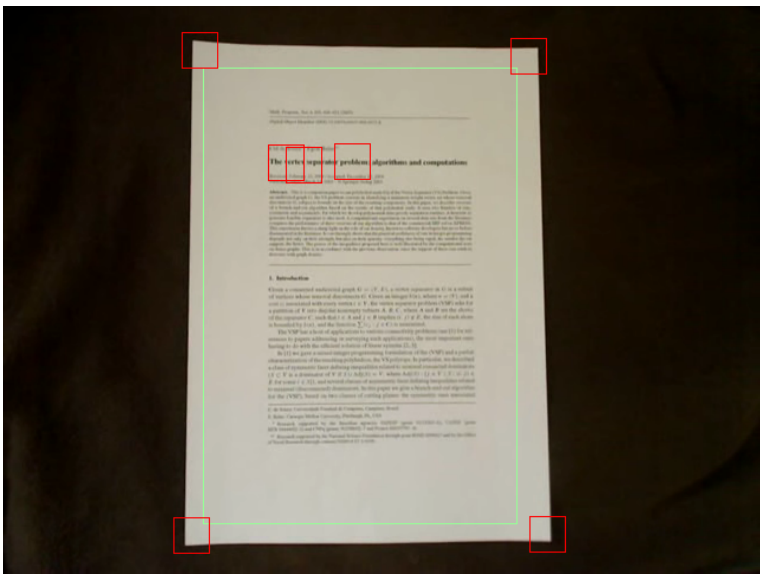

Figura 4.1: Captura 1 del documento 1

En la figura 4.2 se puede observar como se han perdido las dos esquinas inferiores y que llevan perdidas el suficiente tiempo como para adquirir una nueva característica interior para reforzar el seguimiento.

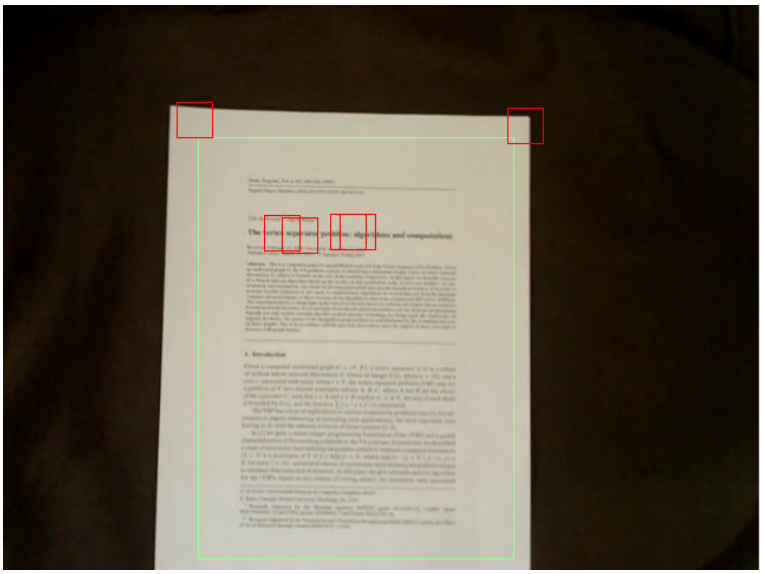

Figura 4.2: Captura 2 del documento 1

Por último al cambiar la posición de la cámara ya se pueden ver de nuevo las cuatro esquinas y como se puede observar no se ha perdido su posición y vuelven a ser relevantes para el cálculo de pesos. (Figura 4.3)

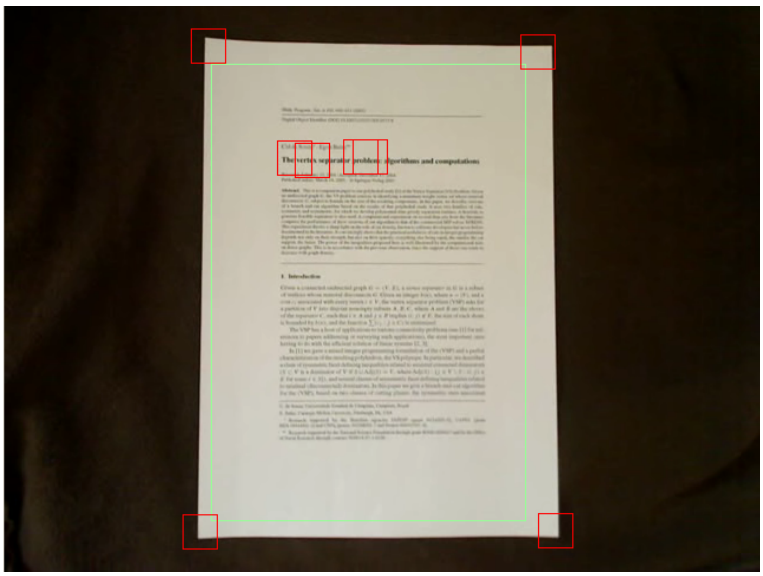

Figura 4.3: Captura 3 del documento 1

En la figura 4.4 se obseva como como se ha perdido el seguimiento debido al uso de un grado de incertidumbre igual a 0. En el caso concreto de esta escena un grado de incertidumbre demasiado bajo hace que la dispersión de las partículas no sea la suficiente

para recuperarse de un movimiento que aleja a alguna de las esquinas de su posición estimada por el filtro.

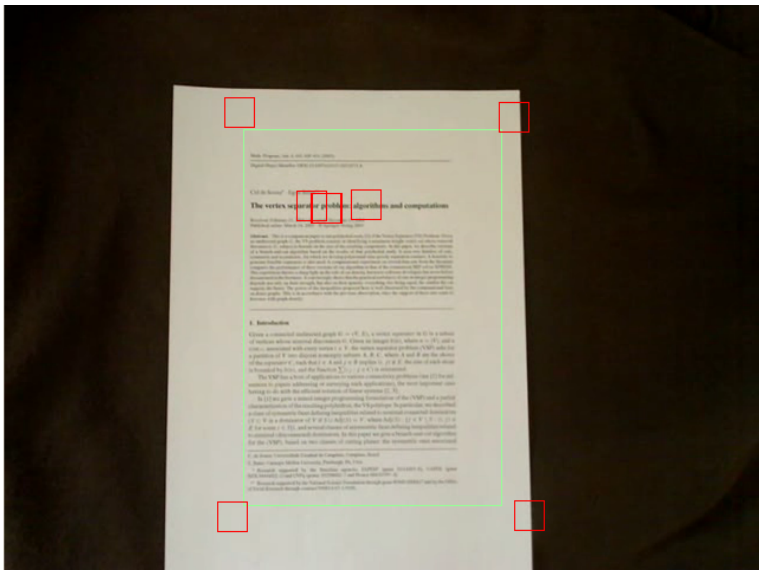

Figura 4.4: Captura 4 del documento 1

4.1.2. Documento 2

Descripción: Documento horizontal, movimiento de translación, acercamiento y pérdida de esquinas.

La detección de esquinas es correcta, aunque éstas tengan letras o figuras en sus alrededores.

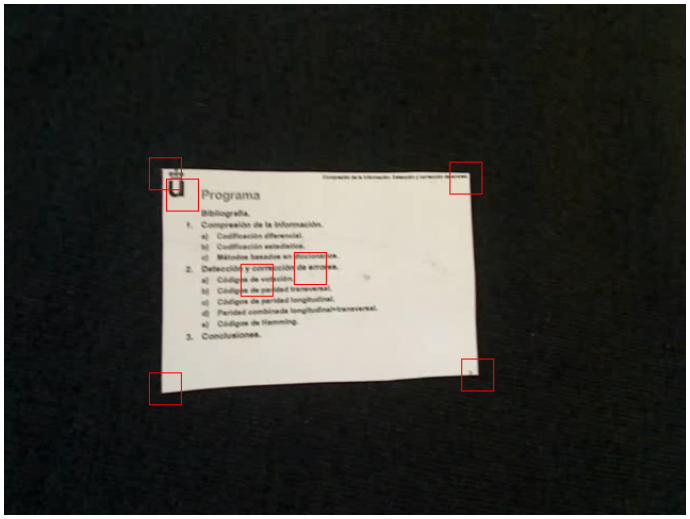

Figura 4.5: Captura 1 del documento 2

En la siguiente figura se aprecia la importancia del margen e seguridad para no tomar el fondo como parte de la zona interior. La cámara empieza a moverse a la derecha.

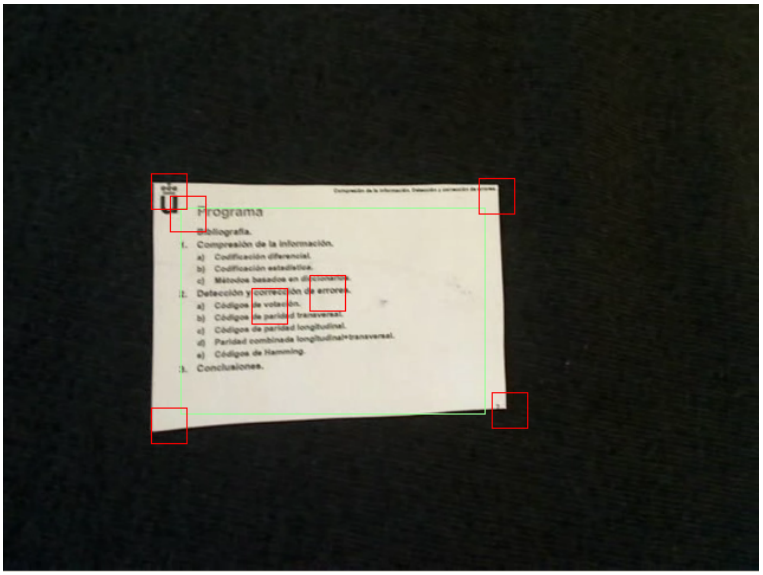

Figura 4.6: Captura 2 del documento 2

La cámara cambia de dirección y se mueve relativamente rápido a la derecha pero el filtro no llega a perderse. (Figura 4.7)

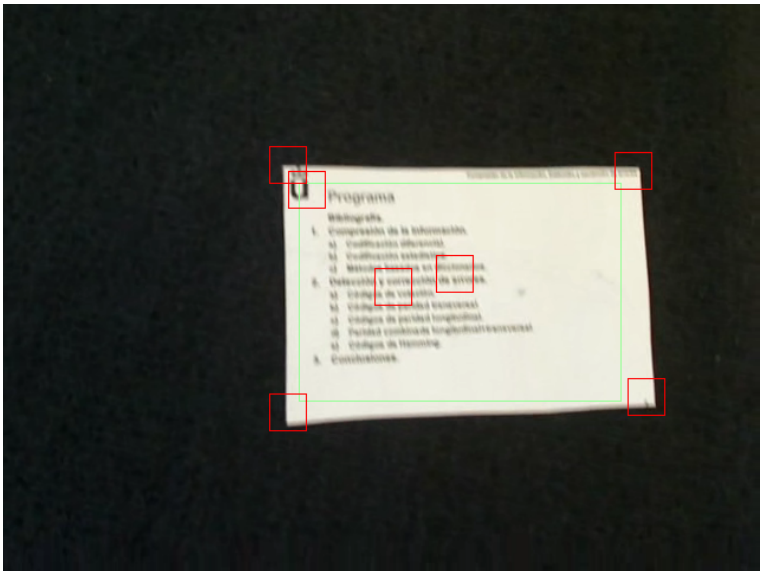

Figura 4.7: Captura 3 del documento 2

Se produce un acercamiento, aumenta la escala hasta aproximadamente 1.4, las muestras se redimensionan y se realiza un escaneo por haber aumentado la resolución. (Ver figura 4.8)

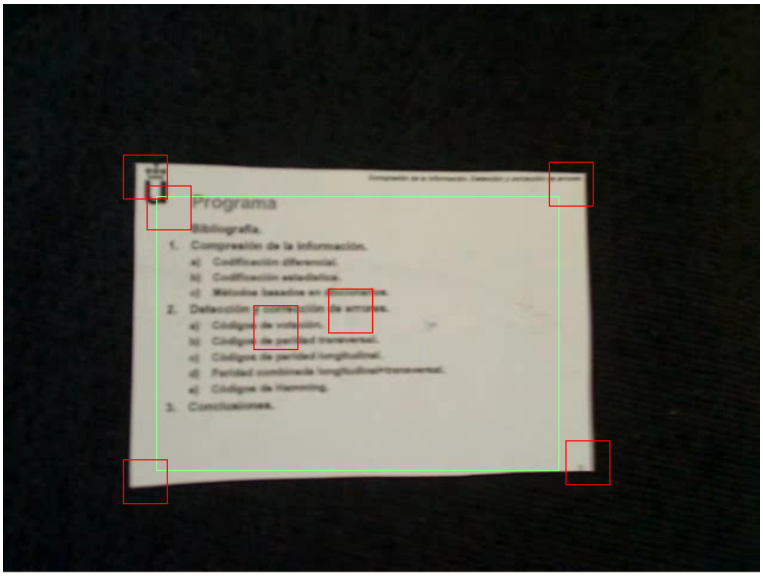

Figura 4.8: Captura 4 del documento 2

Al acercarse más se puede apreciar una pérdida de esquinas y un cambio en la iluminación dejando de ya de ser válido el valor umbral para la binarización de la imagen (ver figura 4.9). Esto hace que el filtro se pierda y estime una escala errónea que acaba por perder el seguimiento por completo.

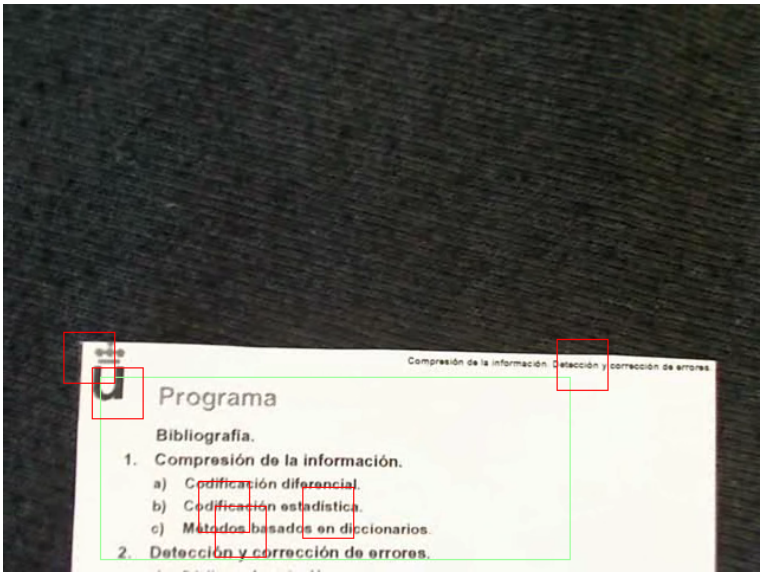

Figura 4.9: Captura 5 del documento 2

4.1.3. Documento 3

Descripción: Acercamiento y pérdida de características.

En esta escena la cámara se acerca mucho al documento y se aprecia una notable mejora en la resolución y el enfoque del documento. (Figura 4.10)

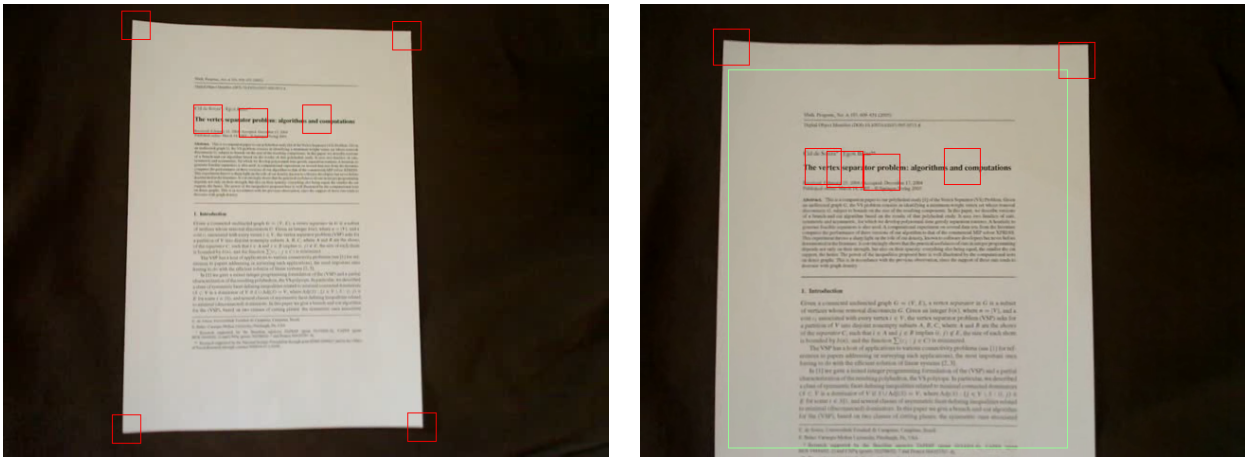

Figura 4.10: Capturas 1 y 2 del documento 3

La figura 4.11 muestra como la resolución ha mejorado mucho pero los cambios en la iluminación, la inclinación del documento y la diferencia entre las características de las muestras redimensionadas y la imagen actual hace que el cálculo de pesos no sea bueno y el filtro no detecta la pérdida de vista de la esquina superior izquierda y posteriormente el resto.

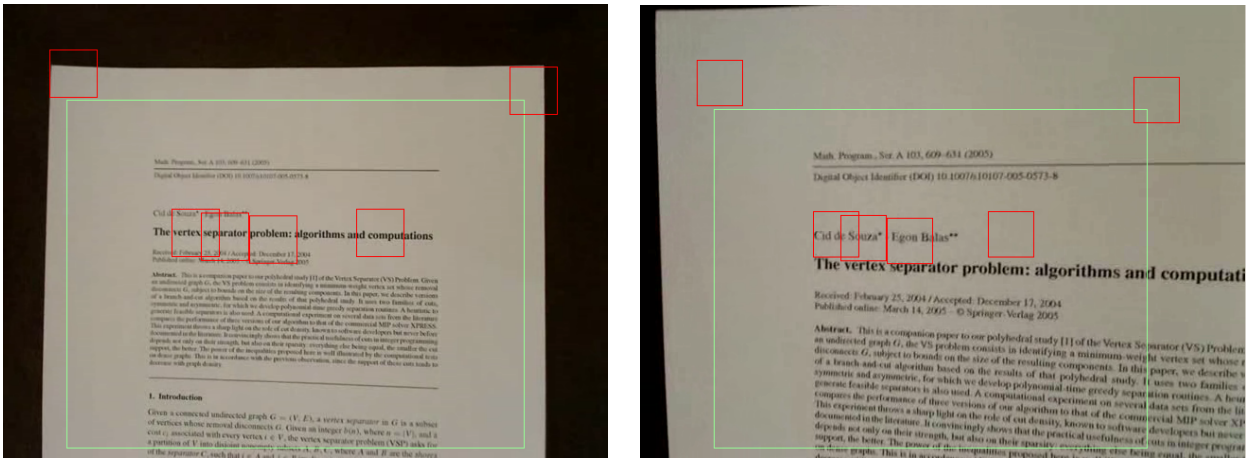

Figura 4.11: Capturas 3 y 4 del documento 3

4.1.4. Documento 4

Descripción: Vídeo sintético con pérdida de las cuatro esquinas.

La detección de esquinas y características es satisfactoria y se produce dispersa por todas las zonas donde están las letras. (Figura 4.12)

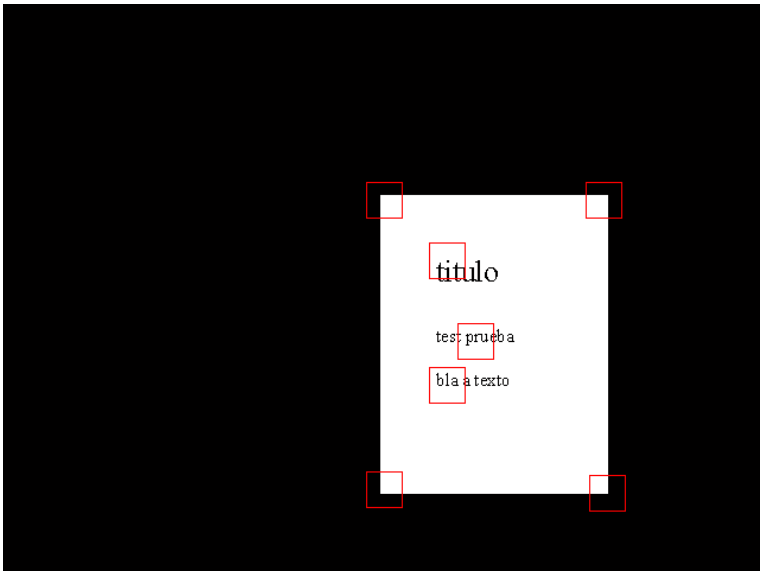

Figura 4.12: Captura 1 del documento 4

Se pierden tres esquinas y se reemplazan las muestras por muchas en el mismo punto de interés. (Ver figura 4.13)

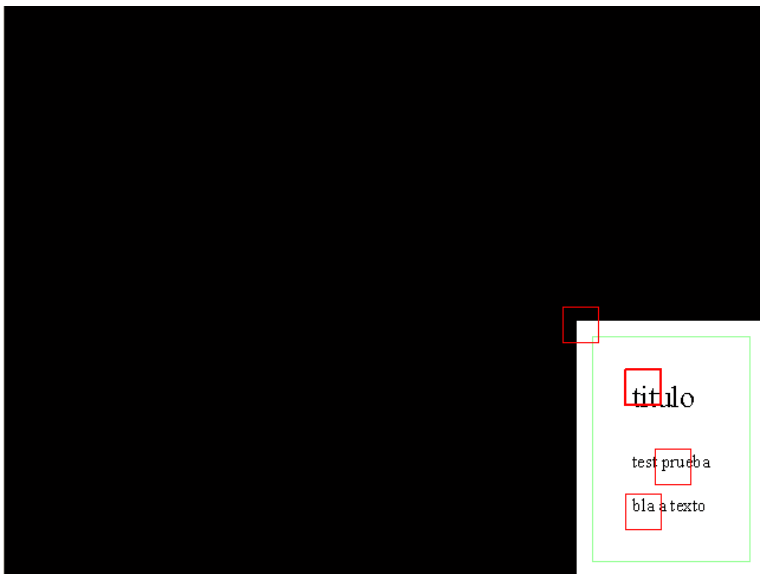

Figura 4.13: Captura 2 del documento 4

Al acercarse al documento la cámara consigue una resolución mucho mejor, recupera una esquina y empieza a perder una de las muestras interiores. (Figura 4.14)

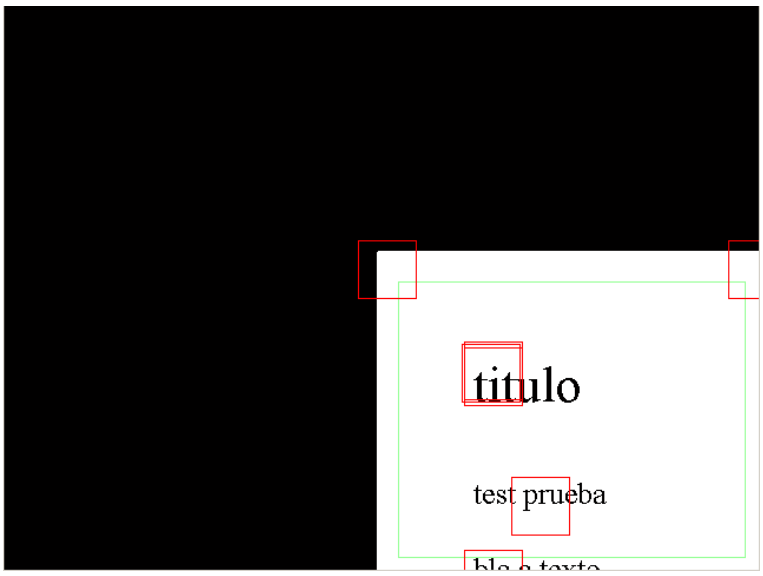

Figura 4.14: Captura 3 del documento 4

La cámara se acerca más y baja de forma que llega un momento de pérdida total de esquinas que se compensa tomando más muestras interiores en la última línea de letras. (Figura 4.15)

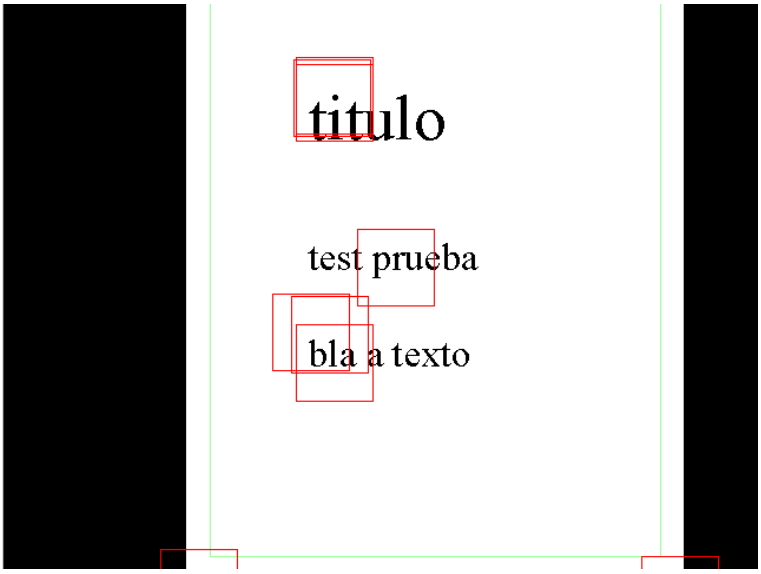

Figura 4.15: Captura 4 del documento 4

En esta figura se puede observar como se han perdido gran parte de las características pero se ha mejorado mucho la calidad de escaneo al haberse acercado tanto la cámara a todas las áreas.

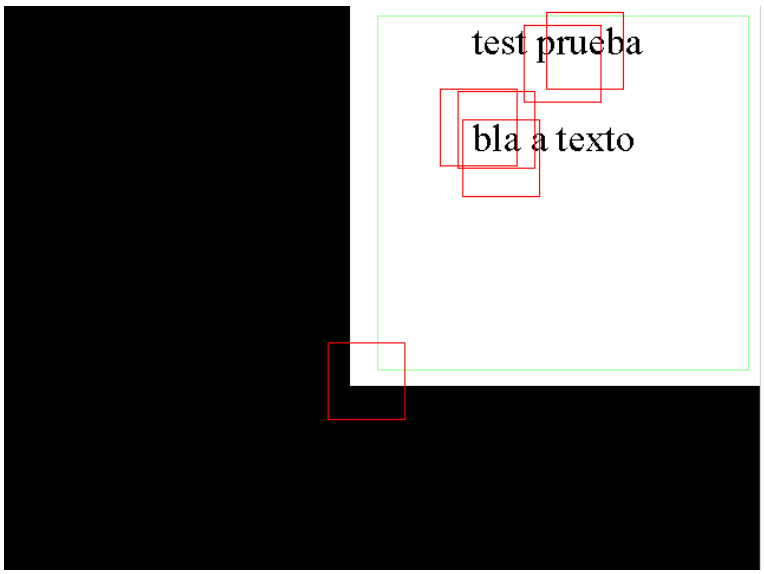

Figura 4.16: Captura 5 del documento 4

Por último se vuelve a un estado en el que puede verse todo el documento y que la estimación del filtro es correcta teniendo todas las muestras en su sitio. (Figura 4.17)

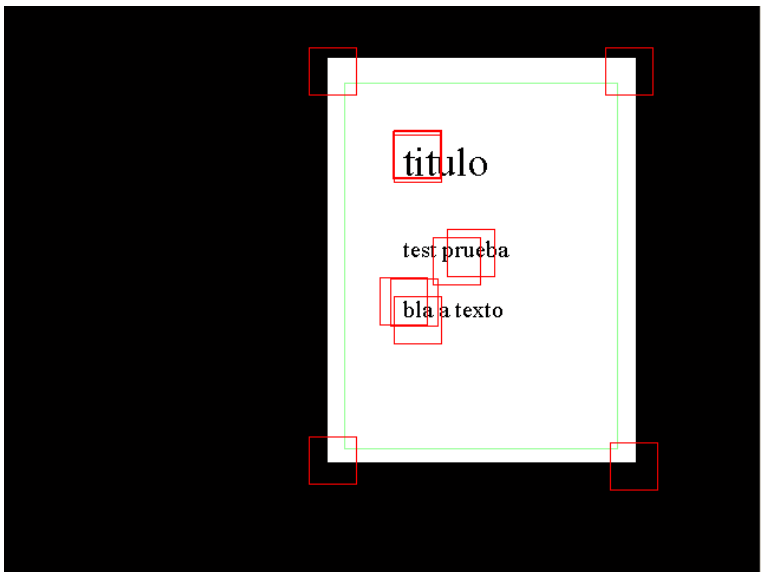

Figura 4.17: Captura 6 del documento 4

Viendo el resultado del escaneo (figura 4.18) se puede observar una mejora notable en la resolución del escaneo inicial habiendo aplicado solamente la detección de esquinas y posteriormente después de haber recorrido el documento completo a una distancia menor.

Figura 4.18: Escaneo 1 al inicio y al fin

Para apreciar mejor la diferencia de escaneos en la figura 4.19 se muestra el detalle en la palabra "titulo".

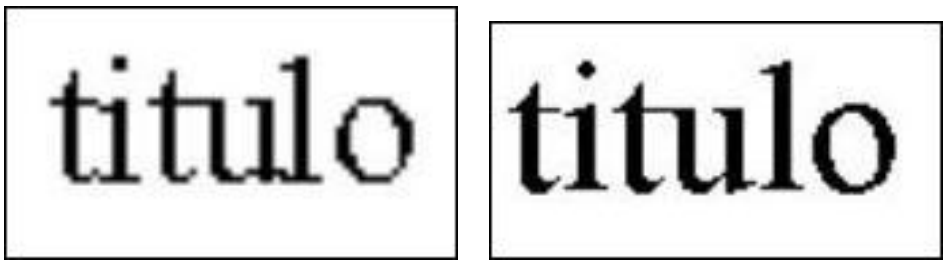

Figura 4.19: Detalle de escaneo 1

4.1.5. Documento 5

Descripción: Vídeo sintético con más texto y pérdida.

La secuencia empieza acercándose al documento y perdiendo las dos esquinas inferiores. (Ver figura 4.20)

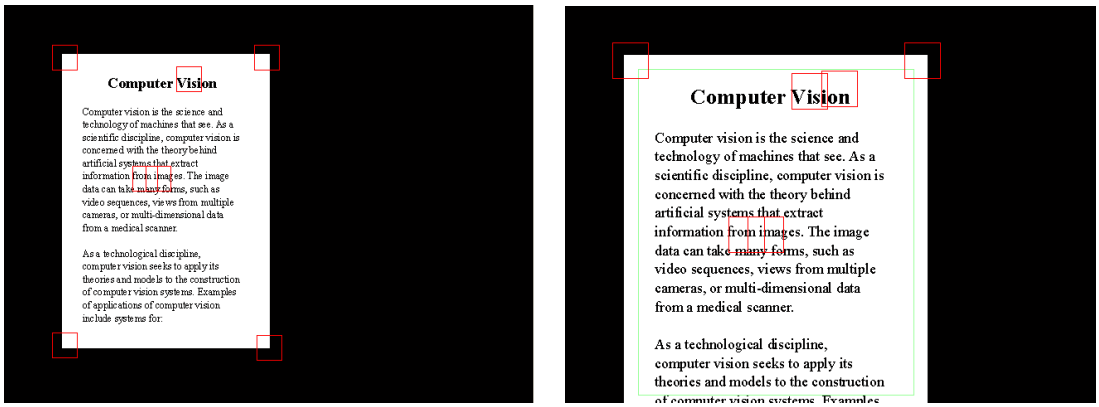

Figura 4.20: Capturas 1 y 2 del documento 5

En la figura 4.21 se puede observar cómo se pierde el seguimiento estimando mal la posición de las esquinas inferiores, la escala y por lo tanto el interior del documento, lo que produce algunos escaneos conteniendo zonas exteriores.

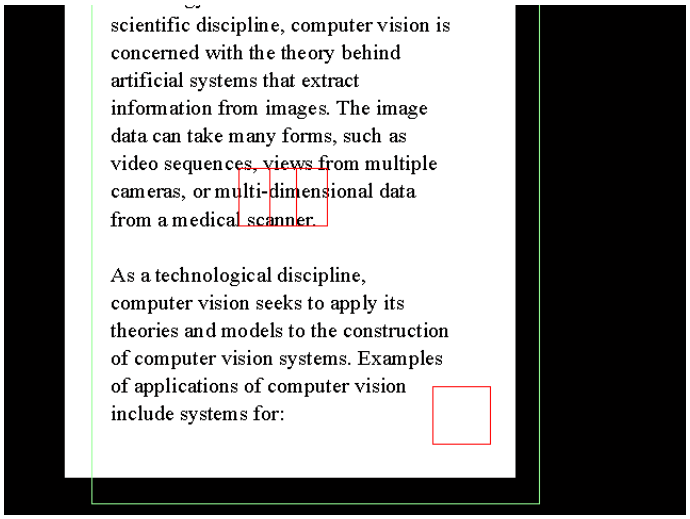

Figura 4.21: Captura 3 del documento 5

En la figura 4.22 se observa como el filtro logra recuperarse de la pérdida aun con sólo tres muestras interiores y ninguna esquina.

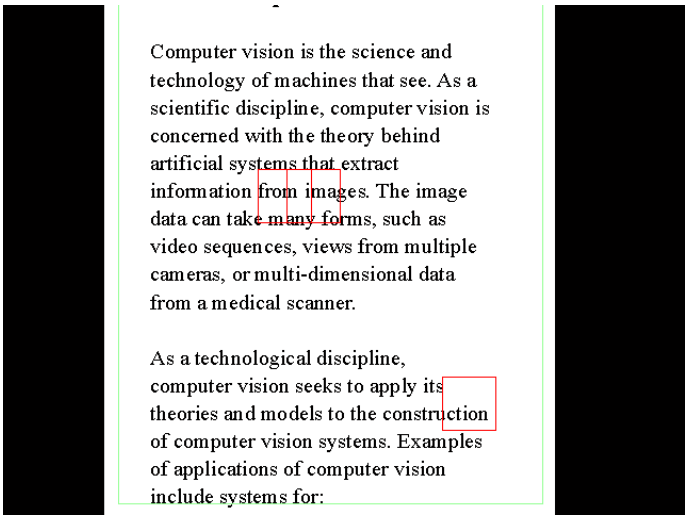

Figura 4.22: Captura 3 del documento 5

Al final la cámara se aleja a mayor distancia que con la que empezó el escaneo y todas las muestras están en su sitio. (Ver figura 4.23)

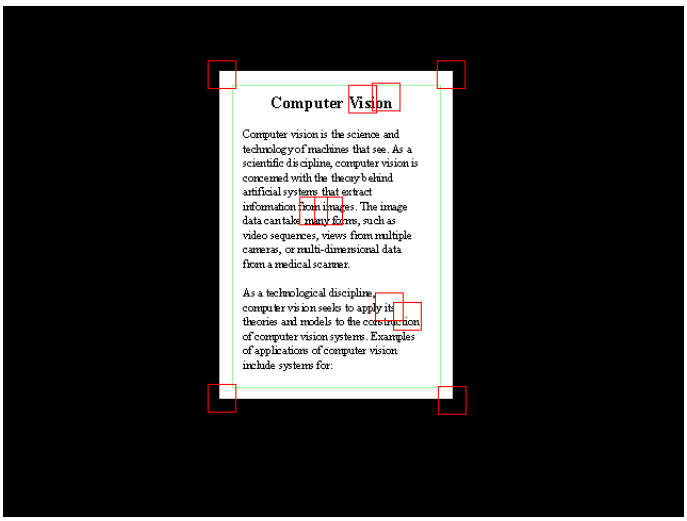

Figura 4.23: Captura 4 del documento 5

Las consecuencias de la pérdida se pueden ver en esta imagen escaneada donde la parte de abajo está descolocada y se ve parte del fondo de la escena. (Figura 4.24)

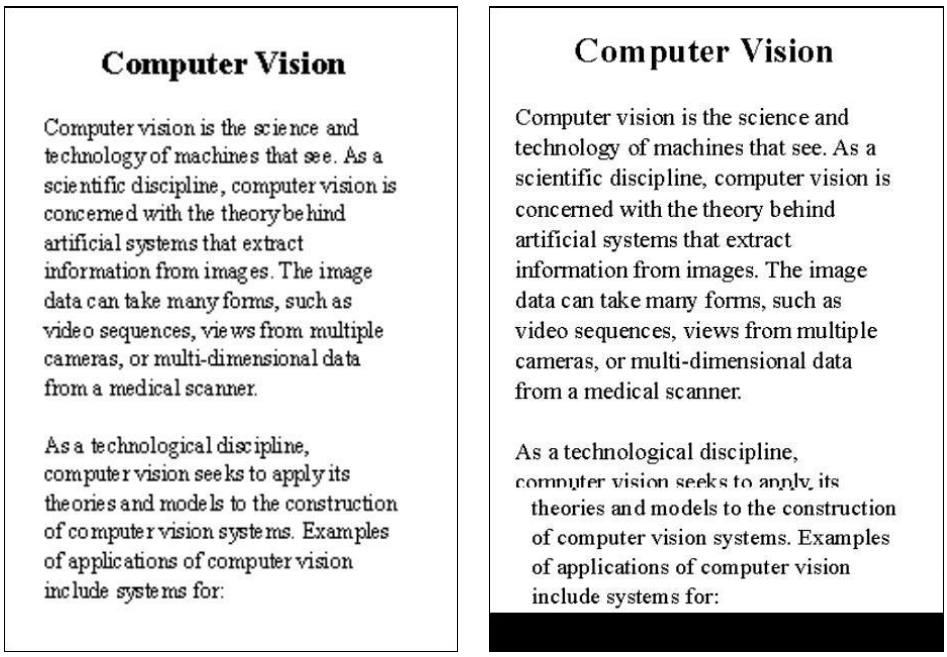

Figura 4.24: Escaneo 2 al inicio y al fin

En la figura 4.25 se aprecia la mejora en la resolución de la parte que ha sido correctamente escaneada.

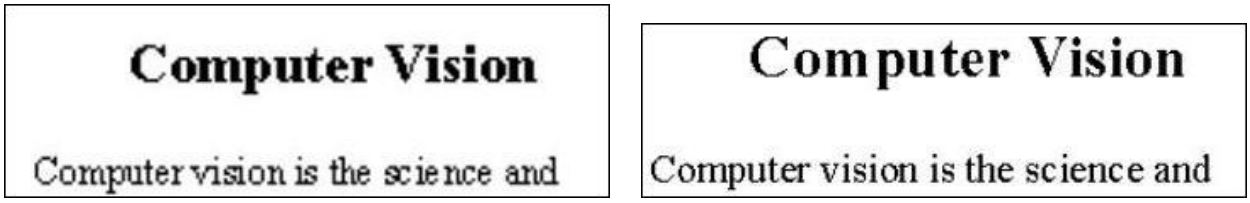

Figura 4.25: Detalle de escaneo 2

Un cambio que puede repercutir mejorando el rendimiento de la aplicación es disminuir el número de partículas, en los ejemplos anteriores se utilizan 200 partículas. En la figura 4.26 se puede observar que el filtro se pierde mucho antes al utilizar sólo 100 partículas.

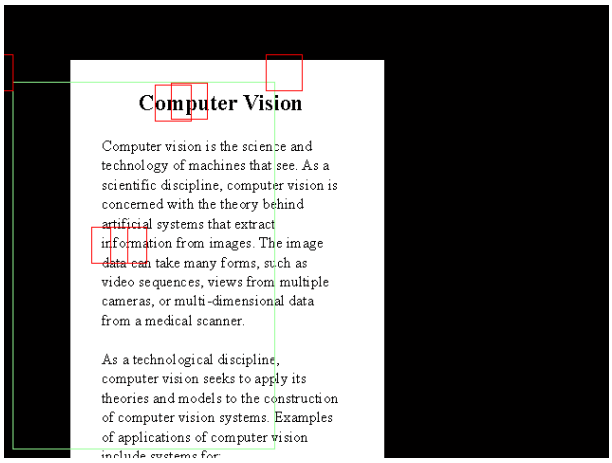

Figura 4.26: Captura de Documento 5 con 100 partículas

4.2. Análisis de los resultados

Después de analizar los resultados en cada una de las escenas probando con diferentes valores de los parámetros se puede decir que el sistema puede escanear los documentos en las secuencias sintéticas pudiendo recuperarse de alguna pérdida mientras que en los vídeos reales los problemas de iluminación y cambios en las características de la imagen hacen que el filtro se pierda al acercar demasiado la cámara al documento.

El número de partículas que utiliza el filtro es un factor determinante ya que afecta al rendimiento en gran medida, un número menor de 200 partículas provoca una mala estimación de la posición del documento en cuanto empiezan a perderse características.

El grado de incertidumbre, o lo que es lo mismo, el número de partículas con posibles pesos bajos que se dejan en el proceso de selección es importante en la recuperación de una mala estimación, especialmente notable en vídeos reales con movimientos imprevisibles.

Usar una malla en lugar de buscar puntos característicos de la imagen usando *Good Features to Track* no ha dado buenos resultados debido a que muchas de las muestras no son tan características como debieran y no proporcionan una buena referencia para el seguimiento.

El ratio de importancia que se da a los pesos muestras interiores y a las esquinas resulta ser determinante ya que en los vídeos sintéticos se obtiene mucho mayor acierto si las esquinas tienen 1/10 de importancia en el peso. Esto ocurre porque las muestras interiores obtienen unos pesos muy significativos al haber unas letras mejor definidas en el documento.

La diferencia de acierto entre vídeos sintéticos y reales se debe principalmente al problema de la medida, destacando los problemas de enfoque y de iluminación que no se producen en vídeos sintéticos.

El problema de la iluminación puede aplacarse en algunos casos cambiando el umbral de binarizado pero sólo hasta cierto punto.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Con este proyecto se presenta la posibilidad de utilizar una cámara de vídeo para realizar escaneos de documentos. Para ello se han combinado y adaptado varios métodos de procesado de imágenes existentes. Las técnicas de filtrado de imágenes han servido para extraer información importante. Con funciones de reconocimiento de patrones se han podido detectar las esquinas de un documento y extraer características interesantes del interior. La implementación de un filtro de partículas ha hecho posible el seguimiento del documento permitiendo así que la cámara pueda moverse. Saber la posición del documento hace posible la auto-localización de la cámara en todo momento. Por último una vez conocida la posición de la cámara sobre el documento se puede extraer la imagen e integrarla como parte de una imagen global del documento.

La aplicación es capaz de detectar y seguir un documento admitiendo variaciones en la posición de la cámara sobre el plano del documento y la distancia a este, así como la pérdida parcial de las características del documento, obteniéndose resultados satisfactorios sobre vídeo sintéticos.

Se ha conseguido escanear documentos en vídeos donde varía la distancia al documento, aumentando la resolución de la imagen escaneada sin perder la proporción y estructura del contenido original. Pudiendo también escanear imágenes en las que sólo se aprecian zonas parciales del documento.

5.2. Trabajos futuros

A continuación se proponen algunas mejoras interesantes que se podrían aplicar al proyecto:

- Añadir una ayuda basada en la geometría del documento para la detección de las esquinas, haciendo esta más fiable.
- Aplicar mejoras al filtro de partículas para minimizar las pérdidas y evitar temblores que puedan afectar al escaneo.
- Hacer que la auto-localización tolere la inclinación y rotación de la cámara teniendo en cuenta los posibles cambios de perspectiva.
- Integrar un sistema de reconocimiento óptico de caracteres que permita reconocer el texto mientras está siendo escaneado.
- Estudiar la utilización de otros filtrados de imagen que toleren mejor los cambios en la iluminación.
- Al realizar el escaneo aplicar una búsqueda local para encajar las imágenes parciales lo mejor posible y conseguir mayor coherencia con la imagen total.

Anexo 1: Funcionamiento de la aplicación

La aplicación toma la ruta de un vídeo como primer parámetro y la ruta del la imagen 'jpg' donde se guarda el escaneo como segundo parámetro. Los vídeos usados en las pruebas se encuentran en la carpeta "videos". Los parámetros de configuración están definidos como constantes dentro del código fuente al inicio de "main.cpp".

Durante la ejecución se pueden realizar determinadas acciones pulsando las siguientes teclas:

- *ESC*: Continuar/terminar.
- *barra espaciadora*: Activar/desactivar auto-escaneo.
- f: Forzar escaneo.
- s: Guardar imagen actual escaneada en la ruta especificada.
- a: Renovar muestras con la imagen actual de cada recuadro.

Bibliografía

[Bru09] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.

[Dav03] A.J. Davison. *Real-time simultaneous localisation and mapping with a single camera*. 2003.

[JV03] Á. Sánchez y J.L Esteban J.F Vélez, A.B Moreno. *Visión por Computador*. Dykinson, 2003.

[Mac02] J. MacCormick. *Stochastic Algorithms for Visual Tracking*. pringer-Verlag, 2002.

[Sán03] J.Hernández Sánchez. Filtros de partículas con realimentación basada en memoria. Proyecto Fin Carrera, Universidad Rey Juan Carlos, 2003.

[Yan85] Wang Ching Yang. *Ph.D. Edge detection using Template Matching (Image processing, Threshold logic, Analysis, Filters)*. Duke University, 1985.