UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

DESARROLLO DE SOFTWARE PARA EL INTERFERÓMETRO DE MOTAS TOHTLI: CAPTURA Y PROCESAMIENTO DE DATOS, RECONSTRUCCIÓN DE IMÁGENES ASTRONÓMICAS CON ALTA RESOLUCIÓN ESPACIAL

REPORTE DE SERVICIO SOCIAL

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A
ANDRIY SVYRYD

T U T O R:
DR. VALERI ORLOV

2007

FACULTAD DE CIENCIAS

División de Estudios Profesionales

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales
Facultad de Ciencias
P r e s e n t e .

Por este medio hacemos de su conocimiento que hemos revisado el trabajo escrito titulado:

"Desarrollo de software para el interferómetro de motas Tohtli: captura y procesamiento de datos, reconstrucción de imágenes astronómicas con alta resolución espacial"

realizado por **Svyryd Andriy,** con número de cuenta **404036799,** quien opta por titularse en la opción de **Servicio Social** de la Licenciatura en **Ciencias de la Computación.** Dicho trabajo cuenta con nuestro voto aprobatorio.

Tutor(a) Propietario	Dr.	Valeri Orlov
Propietario	Dr.	Valerii Voitsekhovich
Propietario	Dra.	Hanna Oktaba
Suplente	Dr.	José de Jesús Galaviz Casas
Suplente	Dr.	Favio Ezequiel Miranda Perea

A t e n t a m e n t e
"POR MI RAZA HABLARÁ EL ESPÍRITU"
Ciudad Universitaria, D.F., a 13 de noviembre del 2007.
EL COORDINADOR DEL COMITÉ DE TITULACIÓN
DE LA LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

DR. JOSÉ DE JESÚS GALAVIZ CASAS

Señor sinodal: antes de firmar este documento, solicite al estudiante que le muestre la versión digital de su trabajo y verifique que la misma incluya todas las observaciones y correcciones que usted hizo sobre el mismo.

Agradecimientos

Al Dr. Valeri Orlov por su paciencia y dedicación, así como el estimulo y ayuda que me ofreció en la realización de este trabajo.

A los miembros del jurado: Valerii Voitsekhovich, Hanna Oktaba, José de Jesús Galaviz Casas y Favio Ezequiel Miranda Perea por la revisión del trabajo y sus comentarios.

A mis padres por su apoyo y amor incondicional.

A todos mis amigos por no dejarme trabajar en exceso.

A todos mis profesores de la Facultad de Ciencias de la UNAM por enseñarme a ver el mundo de forma racional y estructurada.

Al Instituto de Astronomía de la UNAM por brindarme la oportunidad de realizar este servicio social.

A la Dirección General de Asuntos del Personal Académico (UNAM, México) por el financiamiento bajo el proyecto IN107906-2 (PAPIIT).

Índice

Introducción

La gran mayoría de las investigaciones astronómicas se basan en la radiación electromagnética, emitida por los objetos del Universo, que llega a Tierra. La calidad de las imágenes obtenidas por un telescopio es influenciada por dos fenómenos: la difracción de las ondas electromagnéticas y la distorsión que sufren cuando pasan a través de la atmósfera. Para telescopios grandes lo último es lo que limita la resolución óptica de la imagen. El telescopio espacial Hubble fue puesto en órbita en 1990 para superar las aberraciones debidas a la turbulencia atmosférica, siendo posible alcanzar el límite de difracción como la resolución óptica del instrumento. Sin estas perturbaciones atmosféricas Hubble provee una resolución 10 veces mejor que cualquier telescopio en Tierra con la misma apertura de reflexión, pero no sería sencillo ni práctico poner a todos los telescopios en órbita. Por lo que se están buscando nuevos métodos que permitan superar este obstáculo y así obtener imágenes con resolución limitada solo por difracción.

Hay muchos problemas astronómicos para los cuales se requieren imágenes de alta resolución espacial. El que nos interesa es el estudio de los movimientos propios de sistemas de estrellas binarias y múltiples, que son los sistemas de estrellas más comunes en el universo.

El Instituto de Astronomía de la UNAM cuenta con varios telescopios y el más cercano es el telescopio de 1 m en Tonanzintla, Puebla. Pero el sitio donde se ubica no cuenta con un astroclima bueno, lo que limita su uso.

No obstante, puede ayudar en nuestro estudio si se usa algún método de alta resolución espacial terrestre para aumentar la calidad de las imágenes obtenidas con él. En particular, se puede emplear la interferometría de motas para obtener imágenes de alta resolución las cuales se pueden usar para la medición de la separación angular y el ángulo de posición de estrellas dobles y múltiples.

El problema es que, como cualquier equipo científico universal, un interferómetro de motas es muy costoso, ya que usa componentes y software únicos; sin embargo, es posible construir una versión de bajo costo para una aplicación particular como ésta.

Primero debemos establecer los requerimientos viables para el equipo: necesitamos observar estrellas binarias hasta la magnitud 10 del componente débil con la distancia entre estrellas de 0.1 hasta 8 segundos de arco. Revisando las bases de datos de estrellas dobles y múltiples encontramos que

hay miles de sistemas binarios que cumplen con estos requisitos. Lo que hay que hacer ahora es fabricar un interferómetro de motas utilizando las piezas producidas en serie y software propio para disminuir el costo. Y la parte del problema que se tratará en este trabajo es justamente el desarrollo del software necesario.

En el capítulo 1 de este trabajo se presenta la metodología que se va a usar para el desarrollo del software. En el capítulo 2 se describen los conceptos astronómicos relevantes al problema presente. En el capítulo 3 se plantean los requerimientos para el software. En el capítulo 4 se crea un diseño a partir de los requerimientos. En el capítulo 5 se describen las pruebas a cuales fue sometido el sistema durante la implementación y al final de la misma. Los ejecutables y el código fuente se pueden bajar de http://tohtli.ontalstudios.com

Capítulo 1. Metodología del desarrollo del software

Software es un conjunto de archivos ejecutables generados a partir de los programas en código fuente más los documentos asociados y la configuración de datos que se requieren para que esos programas funcionen correctamente y puedan ser mantenidos (Sommerville, 2002).

El **ciclo de vida** del software es la sucesión de etapas por las que pasa el software desde que inicia un proyecto hasta la finalización del mismo.

El proceso de desarrollo del software lleva asociado una serie de métodos, herramientas y procedimientos que se deben usar a lo largo del proyecto. Para elegir un modelo de ciclo de vida debemos tomar en cuenta la naturaleza del proyecto, así como la aplicación, los métodos a usar, los controles y entregas requeridas.

El fundamento para la implementación de cualquier sistema de software es una metodología adecuada. Hasta la fecha se están desarrollado nuevas técnicas para reducir los gastos del desarrollo y aumentar la calidad del producto final. Entre las principales metodologías están: el modelo en cascada, el modelo en espiral, construcción de prototipos, el proceso unificado y el desarrollo ágil. Se consideraron éstas y algunas otras metodologías para su posible uso en este proyecto, tomando en cuenta que el equipo de desarrollo consiste de una persona, los requerimientos serán bien establecidos y no cambiarán durante el desarrollo, el producto se enfoca más en el procesamiento interno que en la interfaz y no es muy complejo (la cantidad estimada de módulos es relativamente pequeña). El modelo en cascada es bastante sencillo que se puede ser empleado por una persona sin perder tiempo en las prácticas que no ofrecen mucha ventaja en este caso. Entonces llegué a la conclusión que es la metodología más adecuada.

El **modelo en cascada** (Royce, 1970), llamado también el ciclo de vida clásico, exige un enfoque sistemático y secuencial del desarrollo de software que progresa a través de las siguientes fases:

Estudio del dominio: es el primer paso del desarrollo de un software nuevo si los desarrolladores no están lo suficientemente bien informados del dominio de la aplicación del software. Facilita la comunicación del cliente con el equipo de desarrollo ya que ayuda a comprender la terminología y los conceptos específicos a ese dominio.

Captura de requerimientos: en esta fase se formula una descripción clara y no ambigua de lo que será el producto. Se deben proporcionar criterios para validar el producto terminado que se usaran para las pruebas del sistema.

Diseño: en esta fase se produce una descripción abstracta de las partes de las cuales se va a componer el sistema y sus relaciones. También se establece el ambiente de implementación.

Implementación y pruebas unitarias: en esta fase se implementa el sistema usando los lenguajes y las herramientas previamente escogidos. Consta de varias partes: el diseño detallado, la programación y las pruebas unitarias.

Pruebas del sistema: el objetivo de esta fase es la integración y prueba del sistema. Se verifica que el software cumpla con sus requerimientos.

Mantenimiento: es la vida útil del software, es el periodo de tiempo durante el cual el software se usa para su propósito original. Durante esta fase se corrigen los errores encontrados por el usuario.

Capítulo 2. Estudio del dominio

Como se está desarrollando un software para uso científico esta fase es de gran importancia. Para comprender los requerimientos del sistema que se establecerán más adelante, así como para poder implementar las técnicas de procesamiento de imágenes, es necesario conocer algunos conceptos.

2.1 El telescopio

Un telescopio es un instrumento óptico que capta cierta cantidad de luz y la concentra en una región pequeña. La cantidad de luz captada por el instrumento depende fundamentalmente de la apertura del mismo (el diámetro del objetivo o del espejo primario). Para visualizar las imágenes se utilizan los oculares, los cuales se disponen en el punto donde la luz es concentrada por el objetivo (plano focal). Son estos los que proporcionan la ampliación al telescopio. La idea principal en un telescopio astronómico es la captura de la mayor cantidad de luz posible, necesaria para poder observar objetos de baja luminosidad.

Los telescopios se dividen en dos tipos según el tipo de objetivo que utilizan: los reflectores y los refractores. El componente óptico principal de un telescopio, si es refractor se conoce como objetivo o si es reflector como espejo primario, en estos últimos, el espejo más pequeño que hace converger la luz, se conoce como diagonal o simplemente como espejo secundario.

Los reflectores están constituidos de un espejo principal (espejo primario u objetivo), el cual no es plano como los espejos convencionales, sino que fue provisto de cierta curvatura (parabólica) que le permite concentrar la luz en un punto.

El telescopio reflector es el más utilizado por los astrónomos profesionales, dado que es posible construir y dar forma a espejos de grandes dimensiones, no sucede así con los refractores, donde el peso de la lente objetivo se vuelve excesivo y la dificultad de producir una lente de calidad de tales dimensiones es casi imposible y altamente costoso.

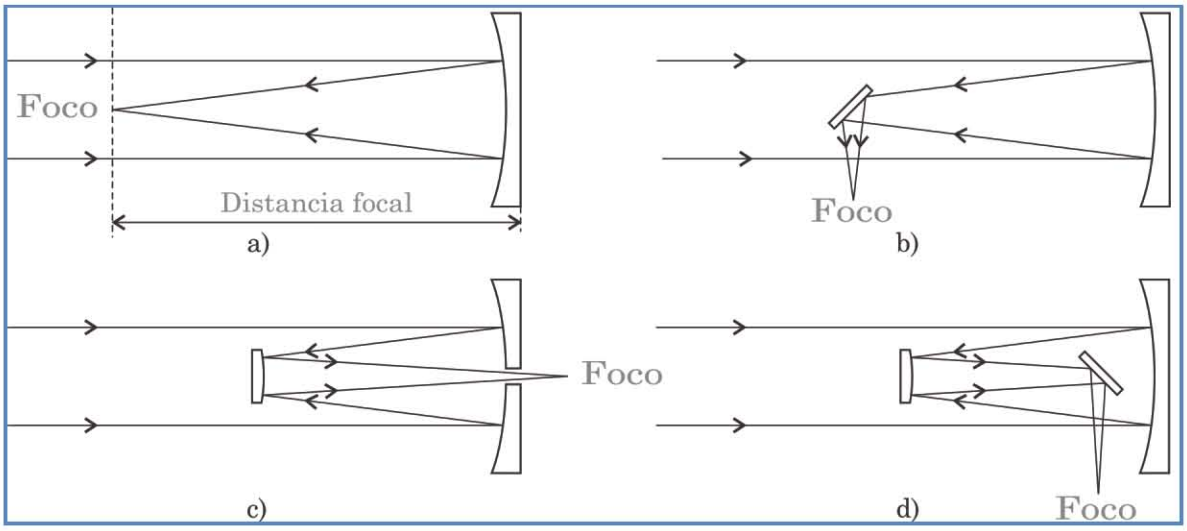

Fig. 2.2 Telescopios Reflectores: Cuatro de los más populares diseños ópticos. a) Foco primario b) Foco Newtoniano c) Foco Cassegrain d) Foco Coude

Dentro de los reflectores existen varios diseños de telescopios. Los más conocidos entre los aficionados son el reflector Newtoniano y el reflector Schmidt-Cassegrain. La principal diferencia radica en la configuración óptica. El reflector Newtoniano dispone de dos espejos, el primario (parabólico) y el secundario (más pequeño y plano), mientras que los Schmidt-Cassegrain poseen un espejo primario también parabólico, pero con una perforación en su centro, para recibir la luz proveniente del espejo secundario, el cual es convexo. Muchos también poseen una placa correctora en la entrada de luz del telescopio.

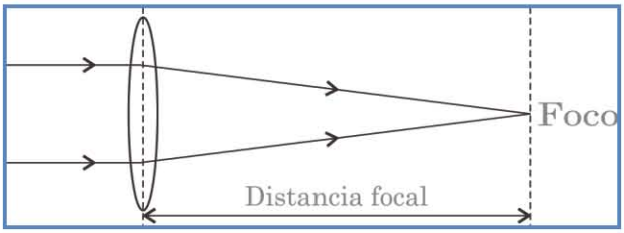

Fig. 2.1 Diagrama del funcionamiento de un telescopio refractor de diseño Kepleriano.

Los telescopios refractores poseen como objetivo una lente (o serie de lentes) que de forma análoga al funcionamiento de una lupa, concentran la luz en el plano focal. Ahí es donde se dispone el correspondiente ocular para ampliar la imagen. Los rayos de los extremos del objetivo son los que sufren la mayor refracción, mientras que en el eje óptico (o eje de simetría), la luz no es desviada.

Una de las ventajas de los telescopios refractores sobre los reflectores es que carecen de obstrucción central (debida al espejo secundario, el cual le "hace sombra" al primario) Esto hace que las imágenes sean más nítidas, y eso se vuelva especialmente adecuado para la observación planetaria y lunar, donde los detalles más finos son los más apreciados. Pero tienen la desventaja de que al querer construir y dar forma a espejos de grandes dimensiones, el peso de la lente objetivo se vuelve excesivo y la dificultad de producir una lente de calidad de tales dimensiones es casi imposible y altamente costoso.

2.1.1 Distancia focal

La distancia focal es distancia comprendida entre el objetivo del telescopio (sea un reflector o refractor) y el plano focal del mismo. Esta medida varía según el diámetro del objetivo y del diseño del mismo (la curvatura del espejo, por ejemplo) Este dato está siempre presente en los telescopios, incluso impreso sobre los mismos dado que es fundamental para determinar muchas características adicionales del equipo. La medida se suele dar en milímetros y sirve para calcular cosas como el aumento, la razón focal, etc.

2.1.2 Razón focal

La razón focal (o **F/D**) es una medida de la luminosidad del telescopio. Está relacionada con la distancia focal y el diámetro del objetivo. Cuanto más corta es la distancia focal y mayor el objetivo, más luminoso será el telescopio.

Para calcular la razón focal de un telescopio solo hay que dividir la distancia focal entre el diámetro del objetivo.

Muchas veces es llamada la "velocidad" del telescopio: se dice que es un telescopio rápido cuando su razón focal es baja (no tiene nada que ver con las características mecánicas del mismo, sino la velocidad de recolección de luz). Esto es especialmente importante en la astrofotografía, donde se pueden reducir sustancialmente los tiempos de exposición si se utilizan sistemas de F/D bajos.

2.1.3 Resolución

La resolución angular de un dispositivo óptico es el ángulo mínimo entre dos objetos distinguibles en la imagen, entre más pequeña sea, más precisión se tiene. En teoría la resolución angular de un telescopio depende de su tamaño y es limitada por la difracción de la luz. Ese límite se calcula como la longitud de onda entre el diámetro del telescopio. Los telescopios grandes tienen teóricamente resolución de hasta un mili-arcosegundo, pero la resolución real es casi siempre menor debido a la distorsión de la luz en la atmósfera. Esto

puede significar fácilmente un factor de 100 entre la resolución potencial y práctica.

La resolución espacial es la resolución angular multiplicada por la distancia focal y determina la distancia mínima entre dos objetos distinguibles. También es el tamaño mínimo de un objeto visible.

2.2 Conceptos básicos del comportamiento de la luz

La radiación electromagnética emitida por los objetos del Universo es con lo que cuentan los astrónomos para realizar sus investigaciones. Esta radiación puede ser de radio, microondas, infrarroja, luz visible, ultravioleta, rayos X o rayos gamma. Lo que diferencia a estas radiaciones entre sí es la longitud de onda (λ) de cada una de ellas. La longitud de onda es la distancia entre dos crestas consecutivas de una onda. A menor longitud, mayor frecuencia.

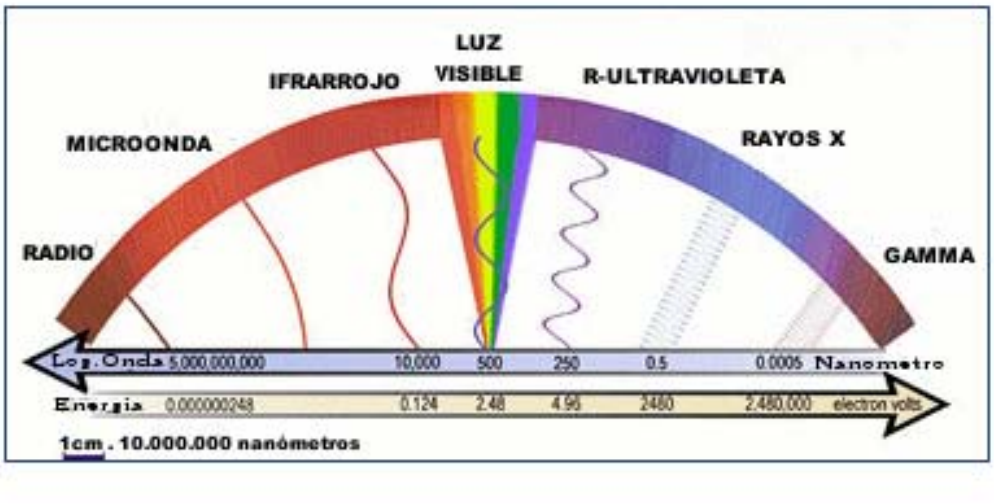

Fig. 2.3 El Espectro Electromagnético

Considerando a la luz como una onda electromagnética y bajo ciertas condiciones se puede suponer que viaja en línea recta; estas líneas son llamadas rayos, y de su estudio se encarga la óptica geométrica.

Un frente de onda es el lugar geométrico en que los puntos del medio son alcanzados en un mismo instante por una determinada onda. Los rayos son ortogonales a los frentes de onda.

Cuando una onda incide sobre la superficie de separación de dos medios distintos, parte de la onda vuelve al medio inicial (reflejada) y parte se transmite al otro medio (refractada).

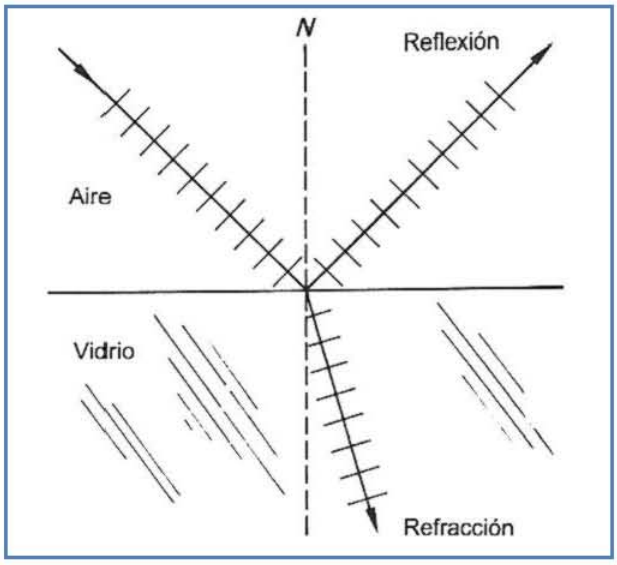

Fig. 2.4 Cuando la luz incide en la frontera entre dos medios, puede reflejarse, refractarse o absorberse.

Como se muestra en Fig. 2.4, los rayos se encuentran en un plano, llamado de incidencia, y en ese plano también se encuentra la normal. La normal es una línea imaginaria que es perpendicular a la superficie y pasa por el punto de incidencia, o el punto donde cae el rayo (Tippens, 1985).

El ángulo de refracción depende del **índice de refracción** del medio, es el cociente de la velocidad de la luz en el vacío y la velocidad de la luz en el medio.

2.2.1 Difracción

La difracción es el fenómeno del movimiento ondulatorio en el que una onda de cualquier tipo se extiende después de pasar junto al borde de un objeto sólido o atravesar una rendija estrecha, en lugar de seguir avanzando en línea recta.

La difracción sólo se observa si el obstáculo que encuentran las ondas es del mismo orden que la longitud de onda del movimiento ya que cuando es mayor, las ondas siguen con una propagación rectilínea.
La expansión de la luz por la difracción produce una borrosidad que limita la capacidad de aumento útil de un microscopio o telescopio. Por ejemplo, los detalles menores de media milésima de milímetro no pueden verse en la mayoría de los microscopios ópticos.

Un modelo para determinar la difracción producida al pasar la luz por una abertura circular fue ideado por George Biddell Airy. Para describir este patrón de difracción Airy empleó la irradiancia. La irradiancia es la potencia promedio por unidad de área que cruza una superficie.

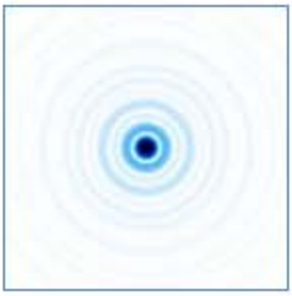

Fig. 2.5 El disco de Airy

El disco formado dentro del primer anillo oscuro recibe el nombre de disco de Airy. Su radio se conoce mediante:

$$r = 1.22\frac{f\lambda}{D}$$

Donde, **r** es el radio del disco, **f** es la distancia focal, **λ** es la longitud de onda, **D** es el diámetro del agujero.

En esta expresión se observa que al aumentar el diámetro disminuye la distancia angular del disco de Airy, con lo que la resolución mejora al aumentar la apertura del telescopio.

2.2.2 Interferencia

El principio de superposición de ondas dice que cuando dos o más ondas se traslapan en espacio, el desplazamiento resultante es igual a la suma algebraica de los desplazamientos individuales. Si las ondas son coherentes (tienen la misma longitud y fase) el efecto producido por la superposición es constante en el tiempo y se conoce como interferencia (Born & Wolf, 1999).

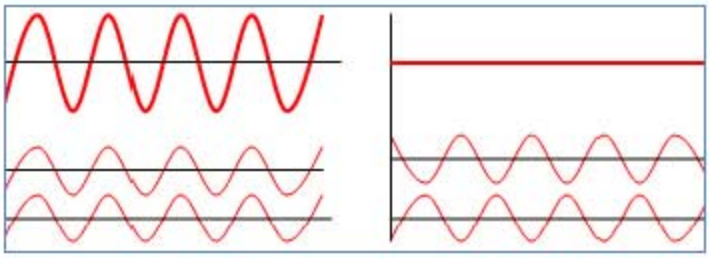

Fig. 2.6 Casos extremos de la interferencia constructiva (izquierda) y destructiva (derecha)

Dependiendo si la amplitud de la onda resultante es mayor o menor que las amplitudes de las ondas originales, la interferencia puede ser constructiva o destructiva.

2.2.3 Motas

El término **"motas"** se refiere a la estructura granular que se observa cuando la superficie irregular de un objeto es iluminada por una fuente de luz coherente. Un ejemplo del fenómeno de motas se puede observar en una alberca cuando muchos nadadores están en ella. Cada nadador emite ondas y la interferencia aleatoria entre ellas causa un campo de ondas salpicadas de manchas en la superficie del agua. Dependiendo de la aleatoriedad de la fuente, espacial o temporalmente las motas tienden a aparecer. El moteado espacial puede observarse cuando todas las fuentes vibran a la misma frecuencia constante pero con diferente amplitud y fase, mientras que el moteado temporal se produce si todas las fuentes son uniformes en amplitud y fase.

2.2.4 La distorsión atmosférica de la luz

Uno de los problemas más grandes de la astronomía basada en Tierra es la distorsión de la luz en la atmósfera. Este fenómeno se conoce como seeing

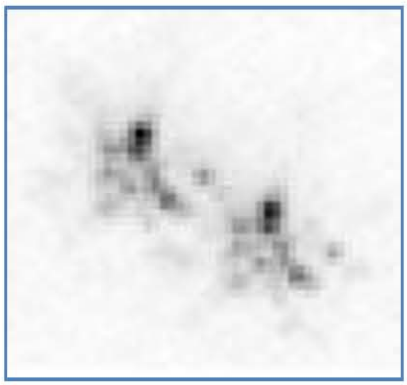

Fig. 2.7 Una imagen invertida de corta exposición de una estrella binaria: Zeta Bootis

astronómico. Se puede apreciar como un centelleo o titileo de las estrellas visible al ojo humano. Es causado por la turbulencia en la atmósfera de la Tierra. Algunas partes tienen diferentes índices de refracción debido a diferencias en temperatura y presión, eso causa que la luz de la misma fuente viaje por caminos diferentes y se distorsiona el frente de onda. La diferencia de fase de los rayos causa la interferencia entre ellos.

El seeing astronómico tiene varios efectos:

- Causa que las imágenes de objetos puntuales (como las estrellas) se rompan en un patrón de motas, que cambia rápidamente con el tiempo
- Las imágenes de larga exposición de estos patrones de motas resultan en una imagen borrosa de la fuente puntual, llamada el disco de seeing.
- El brillo de las estrellas aparenta fluctuar rápidamente.
- La distribución de seeing astronómico no es uniforme, así que las distorsiones de las imágenes de estrellas diferentes tienen patrones diferentes.

2.2.5 Medición del seeing astronómico

Las condiciones del astroclima en un tiempo dado en una locación dada describen cuánto perturba la atmósfera de la tierra las imágenes de estrellas vistas a través de un telescopio. Desde el punto de vista matemático el efecto de la atmosfera sobre la luz se puede describir como una función de dispersión (ya que extiende la luz de una fuente puntual sobre un área) es conocida como PSF (point spread function). La medida más común del seeing astronómico es el diámetro del disco de seeing; técnicamente es el ancho en la mitad del máximo o FWHM (Full Width at Half Maximum) de la función de dispersión. El diámetro del disco de seeing indica la mejor resolución angular que se puede alcanzar por un telescopio óptico con una exposición fotográfica larga. El tamaño del disco de visibilidad varía con el tiempo.

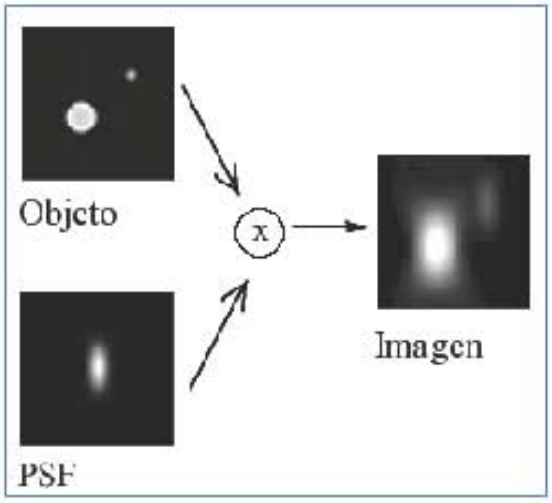

Fig. 2.8 La formación de la imagen en un microscopio mediante la convolución de la imagen original con PSF

Otra medida del seeing astronómico son los valores r_0 y t_0 (Fried, 1966). El primero es el diámetro de telescopio máximo con cual la resolución angular

de imágenes de larga exposición es limitada por el diámetro, todos los telescopios con el diámetro mayor que r_0 tendrán la misma resolución angular que un telescopio en el espacio con diámetro r_0. t_0 es el intervalo de tiempo después de cual la turbulencia del aire cambia significativamente. Los dos valores dependen de la longitud de onda de la luz observada. En un ambiente típico r_0 es alrededor de 12 cm para la luz visible y t_0 es aproximadamente 1 ms.

Fig. 2.9 Imágenes invertidas de corta exposición de una estrella tomada con telescopios de diámetros distintos: a) $20r_0$ b) $7r_0$ c) $2r_0$

2.2.6 Magnitud

En astronomía la magnitud es la luminosidad de una estrella tal como se observa desde la Tierra.

El primer astrónomo que subdividió las estrellas de acuerdo con su magnitud, creando una escala de medidas apropiada, fue el griego Hiparco de Nicea. En la clasificación de Hiparco, se atribuía a las estrellas más luminosas una magnitud 1; a las más débiles visibles a simple vista, magnitud 6.

Con la invención del fotómetro, un instrumento de medida que sirve para determinar la cantidad de luz emitida por una estrella, se ha podido ver que una estrella de magnitud 1 es 100 veces más luminosa que una de magnitud 6. Esto significa que, queriéndole dar una escala precisa a la clasificación de Hiparco (que era empírica, dado que se basaba sobre estimaciones realizadas a simple vista) cada magnitud difiere de la anterior o de la sucesiva en un factor de 2.5.

La escala de magnitudes creada por Hiparco se ha mantenido hasta nuestros días con algunas modificaciones imprescindibles. Se ha extendido a todas las estrellas no visibles a simple vista: aquellas estrellas que tienen magnitudes superiores a 6 y que, en los tiempos de Hiparco, no eran conocidas porque no existían los telescopios (hoy, la estrella más débil visible con los telescopios de tierra más potentes es de magnitud 24).

Por otra parte, en lo relativo a las estrellas más brillantes se ha visto que Hiparco no actuó con mucha sutileza, agrupando bajo la magnitud 1 estrellas que en cambio son mucho más luminosas. Por lo tanto se ha creado una magnitud 0 y después las magnitudes negativas -2, -3, etc. En este caso los números negativos crecientes indican cuerpos celestes siempre más luminosos (el coeficiente de luminosidad entre una magnitud y otra siempre es el mismo, es decir, 2.5).

Este sistema de evaluación de la luminosidad de una estrella se llama también magnitud aparente, porque está condicionado a nuestra posición. Bastaría que nos situáramos en otra estrella para ver cambiar todas las relaciones recíprocas de luminosidad, ya que variarían las distancias entre nuestro punto de observación y las fuentes observadas.

Para conocer la cantidad de energía emitida por una estrella, los astrónomos utilizan la magnitud absoluta, que puede calcularse conociendo las características físicas de la estrella. Conocida la magnitud aparente y la absoluta, los astrónomos pueden también determinar con buena aproximación la distancia de una estrella desde la Tierra.

2.3 Métodos de alta resolución espacial

En la actualidad existen dos clases de métodos que permiten sobrepasar las limitaciones impuestas por la atmósfera terrestre:

En la primera clase están los métodos de tiempo real: la **óptica adaptativa** (Babcock, 1953) que usa equipos ópticos que cambian sus características para compensar la distorsión por la atmosfera y la **síntesis de apertura** (Ryle & Hewish, 1960) que conecta dos o más telescopios para crear un interferómetro grande. Estos métodos son muy costosos, ya que requieren instrumentos muy sofisticados.

En la segunda clase están los métodos de procesamiento posterior a la captura: la **interferometría de motas** (Labeyrie, 1970), la **interferometría de motas diferencial** (Beckers, 1982), la **espectroscopia de motas** (Grieger, 1988), la **holografía de motas** (Liu & Lohmann, 1973), la **polarimetría de motas** (Goodman, 1975) y otros, entre los cuales también hay métodos que no usan imágenes de corta exposición.

Se ha discutido mucho sobre qué métodos dan los mejores resultados, pero ahora está claro que sólo se puede alcanzar la calidad máxima de las imágenes si se combinan los métodos de diferentes clases.

La mayoría de los métodos mencionados solo obtienen un resultado parcial y se necesita aplicar otra técnica para reconstruir la imagen de alta resolución espacial. Esas técnicas se pueden dividir en cuatro clases: **desplazado-y-sumado** (Worden, Lynds, & Harvey, 1976), **método de Knox-Thompson** (Knox & Thompson, 1974), **triple correlación** o **análisis bispectral** (Weigelt, 1977) y **deconvolución** (Gerchberg & Saxton, 1972).

Los métodos que serán empleados para resolver el problema tratado en este trabajo se explicarán más a detalle.

2.3.1 La interferometría de motas

La interferometría de motas es una técnica que consiste en obtener una serie de imágenes de corta exposición de un cuerpo celeste luminoso, donde el efecto del seeing astronómico es congelado y cada imagen obtenida es un patrón de motas. Después se procesan usando la transformación Fourier y se obtiene la autocorrelación. No es la imagen del objeto observado, pero lleva la información sobre su estructura periódica y es simétrica con respecto al origen.

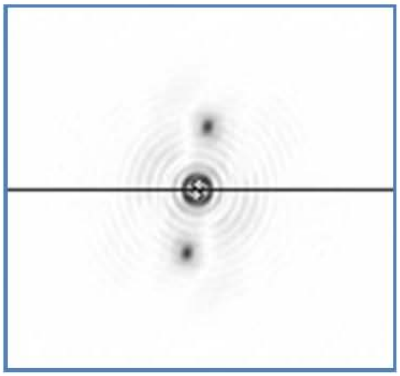

Fig. 2.10 Imagen de una estrella doble tratada con interferometría de motas.

Se puede emplear el análisis bispectral para reconstruir la imagen de alta resolución usando la autocorrelación. Pero no es necesario para la medición de la separación angular de una estrella binaria.

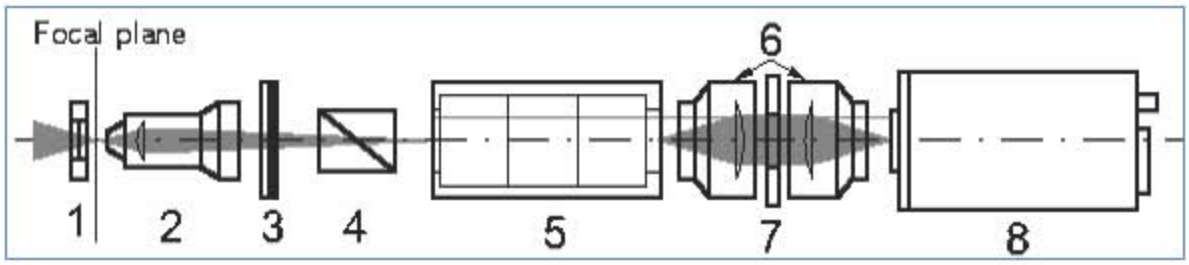

Fig. 2.11 Diseño óptico del interferómetro de motas utilizado en el Observatorio de Rusia: 1) Objetivo magnificador 2) Obturador 3) Filtro 4) Prisma de compensación 5) Intensificador 6,7) Lentes transmisores 8) Cámara CCD

2.3.2 Desplazado-y-sumado

Una técnica simple de reconstrucción de imágenes es el desplazado-y-sumado. Se basa en que una parte importante de las distorsiones de la luz por la atmosfera es simplemente un desplazamiento espacial de la región en observación. En esta técnica las imágenes de corta exposición son alineadas con respecto a la mota más luminosa y se promedian para dar una sola imagen.

Generalmente produce resultados muy buenos a bajo costo pero aun se puede mejorar. Como el seeing astronómico cambia muy rápido algunas imágenes de la serie capturada tienen mejor calidad que las otras. La técnica de desplazado-y-sumado selectivo o afortunado consiste en aplicar el desplazado-y-sumado solo a las mejores exposiciones (generalmente a 1% o menos de la serie original). Con esto sí es posible obtener imágenes con resolución limitada por la difracción.

2.3.3 Eliminación de ruido

La desventaja de los dos métodos descritos es que requieren que el objeto que está siendo observado sea más luminoso que el ruido captado por la cámara. Antes de procesar las imágenes de corta exposición generalmente se aplican varios procedimientos para disminuir el ruido, conjuntamente se denominan como preprocesamiento.

Por lo general para la captura de imágenes de corta exposición se usan cámaras CCD, esto introduce varias imperfecciones en las imágenes, pero muchas se pueden corregir.

Algunas cámaras CCD tienen pixeles defectuosos que se presentan en la secuencia de imágenes como puntos que siempre tienen el mismo color. Son fácilmente identificables por un algoritmo de comparación y se pueden corregir usando interpolación.

El segundo paso es corregir los defectos del equipo óptico como son los rayones y el polvo, también la diferencia de sensibilidad de los pixeles individuales de la cámara CCD. Esto se logra tomando una imagen de un objeto uniformemente iluminado. Después esta imagen se normaliza para que no haya pérdida de intensidad total y todas las imágenes tomadas posteriormente se dividen entre esta imagen referencia.

Por último se elimina el fondo astronómico. Hay varios algoritmos para lograrlo y al más común es la substracción del color predominante.

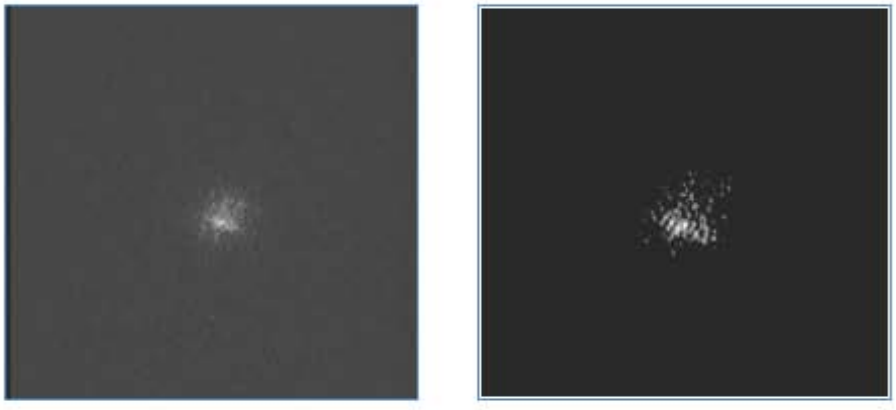

Fig. 2.12 Una exposición antes (izquierda) y después (derecha) del preprocesamiento.

Capítulo 3. Captura de requerimientos

La captura de requerimientos es el proceso que determina qué propiedades o a qué necesidades específicas debe responder o cumplir un producto de software (Endres & Rombach, 2003).

Después de investigar el campo de aplicación del sistema y las técnicas que se usan para propósitos similares, podemos formular los requerimientos.

3.1 El interferómetro de motas Tohtli

Como se mencionó en la introducción la meta de nuestro equipo de investigación es el estudio de movimientos propios de estrellas dobles y múltiples brillantes con el telescopio de 1m en OAN Tonanzintla. Este telescopio permite observar objetos astronómicos con resolución potencial $\lambda B / D = 0.088$", $\lambda V / D = 0.11$", $\lambda R / D = 0.14$" en los espectros de luz azul, visible y roja respectivamente.

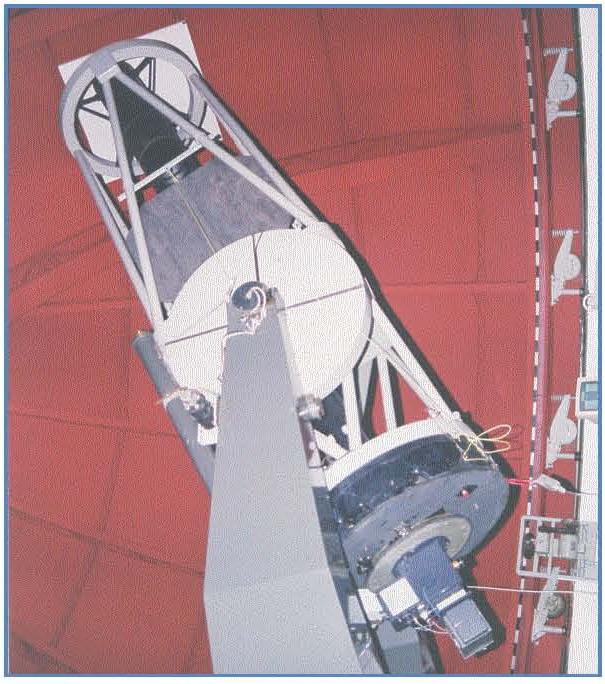

Fig. 3.1 Telescopio de 1 m de Tonanzintla Puebla.

Pero por las condiciones atmosféricas esta resolución no puede ser alcanzada con imágenes tradicionales de largo tiempo de exposición. La resolución común de este tipo de observación casi nunca puede llegar hasta un segundo de arco y en promedio es 1.5 segundos de arco. Es más de 10 veces peor que la resolución potencial. Para realizar el potencial del telescopio fue decidido construir un interferómetro de motas.

En el 2006, en el Instituto de Astronomía de la UNAM se realizaron las primeras observaciones con el método de interferometría de motas. Para ello modificaron el equipo astronómico Dragon (Orlov, Voitsekhovich, Sánchez, & Garfias, 2007).

El equipo Dragon es un instrumento que permite llevar a cabo muchos tipos de investigación experimental relacionada con la óptica adaptativa, la óptica activa, la Astronomía observacional y la instrumentación astronómica (Voitsekhovich, Sánchez, Orlov, Garfias, & Benítez, 2005). El equipo consiste de tres canales ópticos independientes pero sincronizados temporalmente, con lo que se amplía su campo de posibles aplicaciones entre ellas, la interferometría de motas (Voitsekhovich, Orlov, Sanchez, & Garfias, 2007).

La luz de una estrella concentrada por el telescopio es dividida en tres haces con dos divisores de haz. Cada uno de los haces es dirigido a la cámara CCD, la cual graba las imágenes, las digitaliza y las manda al canal correspondiente del acumulador de imágenes. Después de terminar un experimento, la información concentrada por el acumulador es reescrita desde la memoria interna del disco duro de la PC. La PC además controla todas las operaciones que realiza el equipo, y permite que se le asignen todos los parámetros para un experimento y así realizar la reducción de datos.

Continuando con esta línea de investigación, el Instituto de Astronomía de la UNAM, da un paso más y propone el desarrollo de Tohtli (por el vocablo náhuatl que significa "halcón"), un interferómetro de motas de dimensiones adecuadas para su fácil manejo y transporte, que permita la medición de la separación y el ángulo de posición de estrellas dobles y múltiples, utilizando el telescopio de 1m de Tonanzintla, Puebla y que pueda ser manejado por una PC portátil.

Otro miembro del equipo se encargó del diseño del instrumento (Gómez Gallegos, 2007). El resultado fue un interferómetro de motas de configuración sencilla, que puede fabricarse utilizando componentes y manufactura de bajo costo.

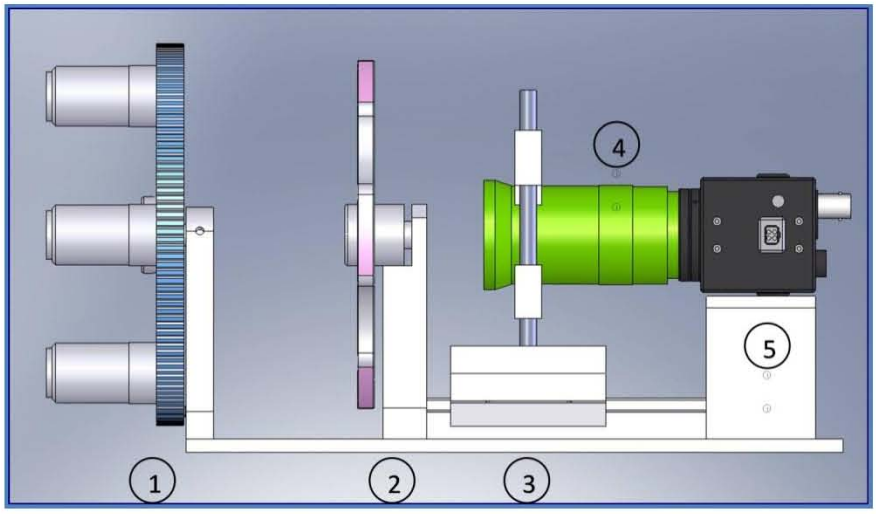

Fig. 3.2 El esquema general del interferómetro de motas Tohtli (1) Rueda de objetivos, (2) Rueda de filtros, (3) Soporte de estructura varifocal (4) Objetivo varifocal, (5) Cámara CCD

La parte más importante del interferómetro de motas Tohtli es la cámara CCD, que transmite el video capturado en forma analógica a una tarjeta digitalizadora. Este video tiene que ser almacenado y posteriormente procesado.

3.2 El enunciado del problema

A continuación se encuentran los requerimientos formulados por el cliente, que en este caso es el Dr. Valeri Orlov.

Se requiere desarrollar un sistema que capture el video de la tarjeta digitalizadora y lo almacene en el disco duro. Posteriormente debe poder abrir el archivo creado, aplicar algunas acciones de análisis, mostrar el resultado y guardarlo en el disco duro.

Las acciones de análisis son: cálculo de la autocorrelación, eliminación de ruido de camara, acumulación de imágenes, desplazado-y-sumado o desplazado-y-sumado selectivo de las imágenes.

Debe poder exportar imágenes individuales del video capturado en formato PNG.

Debe poder usar cualquier fuente de video digital, conectada y configurada en el equipo, y todos los modos de captura que ofrece.

El sistema debe poder mostrar el video a capturar. También debe permitir al usuario escoger la parte rectangular del video que será capturada.

La captura del video debe usar los recursos eficientemente para evitar la pérdida de imágenes.

Antes de la captura se especificarán el número de cuadros a capturar y un comentario. Estos datos serán almacenados junto con el video.

Debe poder almacenar los tres canales de color o sólo uno.

Se deben reportar las imágenes perdidas durante la captura.

La interfaz del usuario de los programas debe ser en inglés.

El equipo en el que se realizarán las observaciones cuenta con la siguiente configuración:

- Sistema operativo: Windows XP SP2 Professional Edition
- Memoria RAM: 1GiB DDR400
- CPU: Intel Celeron de 2Ghz
- Disco duro: 80GB a 7800RPM

3.3 Glosario

Algunas palabras del texto del problema podrían ser ambiguas, entonces en esta sección se les dará un significado único que se usará a lo largo del proyecto.

Cámara – una fuente de video digital compatible con DirectShow.

Modo de captura – el formato en que se digitaliza el video, incluye resolución y el número de bits por pixel.

Video – una secuencia de imágenes con dimensiones y formatos iguales, separados por unos intervalos de tiempo fijos.

3.4 Los casos de uso

Hay dos tipos de necesidades o requerimientos: los funcionales y los no funcionales. Los funcionales incluyen:

- Las entradas y salidas de datos, los cálculos, las funciones o casos de uso.

Entre los requerimientos no funcionales están:

- Las necesidades de la interfaz externa: tipo de usuario, hardware, software, comunicaciones, facilidad de uso por sus usuarios.
- Los atributos del software: eficiencia, disponibilidad, seguridad, conversión, portabilidad, mantenimiento.
- Restricciones del diseño: formatos de archivos, lenguajes, estándares, compatibilidad.
- Otros: base de datos, instalación, etc.

Para especificar los requerimientos funcionales se usarán los casos de uso, que serán el hilo conductor de todo el proceso de desarrollo. Esta técnica se utiliza para identificar los requerimientos funcionales y a partir de ellos se diseña, implementa y se prueba el software. Permite rastrear los requerimientos a través de todo el proceso de desarrollo hasta el producto terminado.

Un caso de uso (CU) es una descripción de un conjunto de secuencias de acciones que realiza el sistema para entregar un resultado o valor observable a un actor (Booch, Rumbaugh, & Jacobson, 1999).

Un actor es algo externo al software que intercambia información con el sistema, puede ser un usuario u otro sistema. El objetivo de un actor es usar una funcionalidad del sistema para obtener un valor o servicio. Se representa con una figura humana con el nombre del actor en singular.

Los casos de uso proporcionan una manera incremental y modular de describir sistemas. El conjunto de los casos de uso describe el comportamiento general del sistema. Es una representación que puede ser fácilmente comprendida por todos los interesados. Se representan gráficamente con diagramas de caso de uso de UML.

Se identificaron dos conjuntos de requerimientos funcionales independientes: los relacionados con la captura de video y los de procesamiento. Para mayor simplicidad de la interfaz gráfica el sistema fue separado en dos programas: Tohtli StarFinder y Tohtli StarCapture.

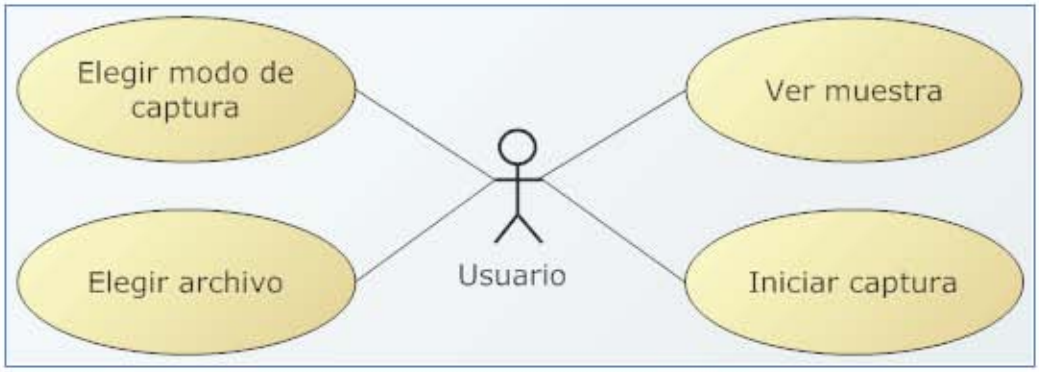

Fig. 3.3 Diagrama de casos de uso de Tohtli StarCapture

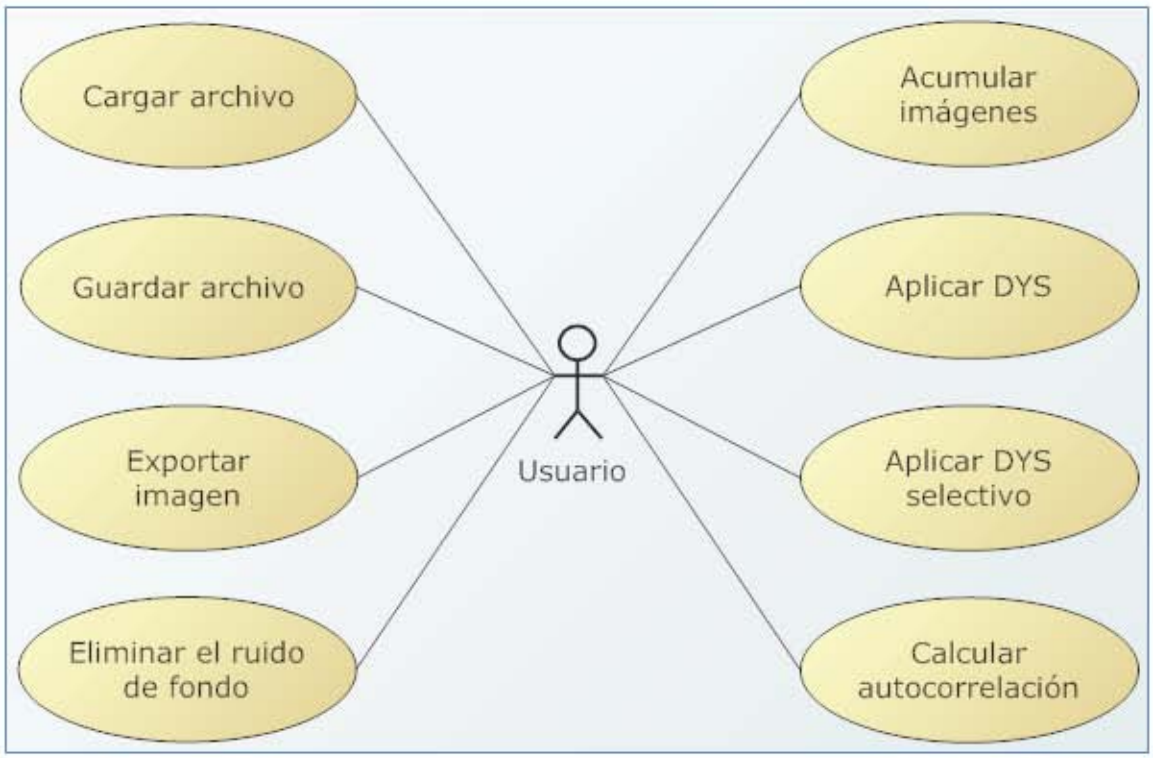

Fig. 3.4 Diagrama de casos de uso de Tohtli StarFinder

A continuación se desglosarán a detalle los casos de uso. Se pueden describir con los siguientes campos:

- Descripción – una breve descripción del propósito del CU.
- Accionadores – los eventos que inician el CU.
- Precondiciones – lo que se asume del sistema antes del inicio del CU.

- Flujo principal – la secuencia de eventos que se presentará en la mayoría de las veces durante el CU.
- Flujos alternativos – condiciones especiales que alteran el flujo del CU.
- Poscondiciones – el estado del sistema después de que se termine el CU.

3.4.1 Elegir modo de captura

Descripción: El usuario elige una cámara y un modo de video.

Accionadores: El usuario activa la lista de las cámaras detectadas.

Flujo principal:

Usuario	Sistema
	1. Presenta una lista de cámaras detectadas.
2. Elige una cámara de la lista.	3. Presenta la lista de los modos de captura detectados para la cámara seleccionada.
4. Elige un modo de captura de la lista de modos.	5. Actualiza la región de captura.

Poscondiciones: Un modo de captura fue elegido.

3.4.2 Iniciar captura

Descripción: El sistema captura el video en un archivo.

Accionadores: El usuario presiona el botón “Iniciar captura”.

Precondiciones: Un modo de captura fue elegido.

Flujo principal:

Usuario	Sistema
	1. Almacena en el archivo elegido la cantidad dada de imágenes.

Flujos alternativos:

Condición	Acción
1a. No se ha elegido un archivo.	CU Elegir archivo.

3.4.3 Elegir archivo

Descripción: El usuario elige el nombre del archivo en el que será almacenado el video capturado.

Accionadores: El usuario presiona el botón "Guardar como…".

Flujo principal:

Usuario	Sistema
	1. Presenta un dialogo de elección del nombre de archivo.
2. Introduce el nombre del archivo.	

Flujos alternativos:

Condición	Acción
2a. Ya existe un archivo con el nombre elegido.	Presenta la opción de sobrescribir el archivo existente.

Poscondiciones: El nombre del archivo fue elegido.

3.4.4 Ver muestra

Descripción: El sistema presenta una muestra del video de la cámara.

Accionadores: El usuario presiona el botón "Ver muestra"

Precondiciones: Un modo de captura fue elegido.

Flujo principal:

Usuario	Sistema
	1. Presenta una ventana con el video recibido de la cámara.
2. Presiona "Ver muestra".	3. Cierra la ventana del video.

Flujos alternativos:

Condición	Acción
2a. El usuario selecciona una parte del video con el puntero.	Guarda la región seleccionada como la región de captura.

3.4.5 Cargar archivo

Descripción: El sistema carga un archivo del disco duro.

Accionadores: El usuario presiona el botón "Cargar".

Flujo principal:

Usuario	Sistema
	1. Presenta un dialogo de elección del archivo.
2. Elige un archivo.	3. Carga el archivo elegido. 4. Muestra información general sobre el video.

Flujos alternativos:

Condición	Acción
2a. El archivo no tiene el formato correcto.	Muestra un error y detiene la carga del archivo.
2b. El sistema está en modo de uso bajo de memoria	Sólo carga la primera imagen del video.

Poscondiciones: Un video fue cargado.

3.4.6 Guardar archivo

Descripción: El sistema almacena en disco duro el video actual.

Accionadores: El usuario presiona el botón "Guardar".

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
	1. Almacena en el disco duro el video actual.

Flujos alternativos:

Condición	Acción
1a. No se ha elegido un archivo.	CU Elegir archivo.

3.4.7 Acumular imágenes

Descripción: El sistema acumula las imágenes del video en una sola imagen.

Accionadores: El usuario elige "Acumular imágenes" desde el menú.

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
	1. Para cada pixel de la imagen calcula el valor promedio de intensidad de todas las imágenes del video.

Flujos alternativos:

Condición	Acción
1a. El sistema está en modo de uso bajo de memoria.	Procesa el video cuadro por cuadro y guarda el resultado al terminar.

3.4.8 Eliminar el ruido de fondo

Descripción: El sistema elimina el ruido de fondo de todas las imágenes del video actual.

Accionadores: El usuario elige "Eliminar el ruido de fondo" desde el menú.

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
	1. Para cada imagen del video el sistema calcula la intensidad predominante y la substrae de todos los pixeles de la imagen.

3.4.9 Exportar imagen

Descripción: El sistema guarda la imagen actual en formato Portable Network Graphics (PNG).

Accionadores: El usuario elige "Exportar imagen" desde el menú.

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
	1. Presenta un dialogo de elección del nombre de archivo.
2. Elige un nombre de archivo.	3. Guarda la imagen actual bajo el nombre elegido.

3.4.10 Aplicar DYS

Descripción: El sistema aplica el desplazado-y-sumado a las imágenes del video actual.

Accionadores: El usuario elige "Desplazado-y-sumado" desde el menú.

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
	1. Desplaza el pixel más brillante de cada imagen al centro y promedia las intensidades de las imágenes resultantes.

Flujos alternativos:

Condición	Acción
1a. El sistema está en modo de uso bajo de memoria.	Procesa el video cuadro por cuadro y guarda el resultado al terminar.

3.4.11 Aplicar DYS selectivo

Descripción: El sistema aplica el desplazado-y-sumado a un subconjunto de las imágenes del video actual.

Accionadores: El usuario elige "Desplazado-y-sumado selectivo" desde el menú.

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
1. Indica el límite de calidad.	2. Se usa el CU "Aplicar DYS" para el subconjunto de las imágenes con calidad superior al límite indicado.

3.4.12 Calcular la autocorrelación

Descripción: El sistema calcula la función de autocorrelación del video.

Accionadores: El usuario elige "Autocorrelación" desde el menú.

Precondiciones: Un video fue cargado.

Flujo principal:

Usuario	Sistema
	1. Calcula el promedio del modulo cuadrado de la transformación de Fourier de cada imagen y después calcula la transformación de Fourier inversa del resultado anterior.

Flujos alternativos:

Condición	Acción
1a. El sistema está en modo de uso bajo de memoria	Procesa el video cuadro por cuadro y guarda el resultado al terminar.

3.5 Los requerimientos no funcionales

- Debe poder usar cualquier fuente de video digital, conectada y configurada en el equipo, y todos los modos de captura que ofrece.
- Debe convertir los colores a RGB si están en algún otro espacio de colores.
- La interfaz del usuario de los programas debe ser en inglés.
- Debe ser intuitivamente fácil la selección del modo de captura y del área del video que será almacenado
- Debe ser compilado para Windows XP de 32 bits
- Debe poder trabajar con archivo de tamaño mayor que la RAM disponible.
- Debe usar el tiempo de procesador y el disco duro eficientemente, para poder almacenar 60 imágenes por segundo con resolución 512x512 en blanco y negro en una maquina con un CPU comparable a Intel Celeron de 2Ghz y un disco duro a 7800RPM

3.5.1 El formato del archivo

El archivo con el video consiste de dos partes: el encabezado y los datos. El encabezado es texto codificado con UTF-16 Big Endian de longitud constante de 1024 caracteres o 2048 bytes que tiene la siguiente estructura:

Caracteres	**Descripción**
0-3	La anchura del video
4-7	La altura del video
8-12	El número de imágenes (puede ser menor que el número real de imágenes en el archivo si se perdieron algunas durante la captura)
13	El número de bytes por pixel (a lo más 8)
14-15, 16-17	El principio y fin del canal rojo
18-19, 20-21	El principio y fin del canal verde
22-23, 24-25	El principio y fin del canal azul
26-33	La hora de captura en formato HH:mm:ss
34-41	La fecha de captura en formato dd/MM/yy
42-1023	El comentario

Como existe una variedad de formatos RGB se especifica la posición se la información de color. Por ejemplo RGB565 tiene la siguiente estructura: RRRRRGGGGGGBBBBB, donde cada letra corresponde a un bit. Como se representa en la computadora como un número las posiciones se cuentan de derecha a izquierda, entonces los caracteres 14-25 del encabezado para este formato serían 11 16 05 11 00 05, puede parecer redundante, pero en algunos

formatos hay bits que no son usados por ningún color y podrían contener información basura. Si el video sólo tiene un color, todos los canales apuntan a la misma sección, para simular un video en blanco y negro. Por ejemplo: 00 16 00 16 00 16.

Capítulo 4. Diseño

"El diseño es un proceso creativo para decidir cómo se construirá el producto" (Humphrey, 2000). El diseño describe las partes o componentes principales de cuales se compondrá el software, cómo interactúan, y cómo se juntan en el producto terminado.

4.1 Arquitectura de software

La arquitectura de software es el diseño del más alto nivel. Consiste en definir cuáles serán los componentes que formarán el software.

Para definir la arquitectura adecuada para un producto de software se revisan los requerimientos no funcionales especificados para definir algunas cualidades que debe cumplir. Algunos de estas cualidades pueden ser:

- Extensión. Que sea posible agregar nuevas características.
- Cambio. Que sea fácil adaptar los cambios en los requerimientos.
- Sencillez. Que sea fácil de comprender y de implementar.
- Eficiencia. Que use pocos recursos.

Es improbable que los requerimientos de este software cambien o que se necesiten nuevas características. Y es importante el rendimiento. Además no tiene partes distribuidas, es autocontenido, ya que permite realizar una acción desde el principio hasta el final usando un solo equipo. Por lo que la arquitectura adecuada es el diseño monolítico: cada programa consistirá de un solo componente sin posibilidad de extensión. Esto beneficia el rendimiento y simplifica el desarrollo.

4.2 Selección del ambiente de implementación

El lenguaje de implementación es muy importante en el desarrollo de un software, ya que influye mucho sobre su portabilidad, eficiencia, funcionalidad, mantenibilidad y reusabilidad.

Uno de los requerimientos del sistema es que debe usar cualquier dispositivo de captura de video digital conectado al equipo. Una solución elegante y simple a esto es usar una capa de abstracción del hardware de multimedia. Existen varios APIs para la plataforma objetivo (Windows XP): Video for Windows (VfW) y DirectShow usan los controladores de dispositivos conforme a Windows Driver Model (WDM), pero VfW ya se considera obsoleto. También están los APIs de más alto nivel, que internamente usan VfW o

DirectShow, pero no ofrecen el nivel necesario de flexibilidad. Por lo tanto la opción más razonable es usar DirectShow.

Microsoft DirectShow es un framework de multimedia basado en filtros con soporte de extensiones. Usa interfaces de Component Object Model (COM) para el acceso a las funciones de los dispositivos multimedia. Aunque estas interfaces pueden ser usadas en muchos lenguajes de Microsoft la mayor eficiencia se logra mediante C++, es el lenguaje que será usado para la implementación de Tohtli StarCapture.

En caso de Tohtli StarFinder el rendimiento no es tan importante, entonces se usará C#, porque acelera la implementación y disminuye el número de errores potenciales.

El entorno de desarrollo que se usará es Microsoft Visual Studio 2005 Express Edition, dado que tiene soporte de C++ y C# y ofrece herramientas que facilitan la programación en estos lenguajes.

4.3 Diseño de clases

El siguiente nivel de diseño requiere identificar los elementos de los componentes de la arquitectura. En el caso del diseño orientado a objetos serán clases. Se construyen dos vistas del diseño:

La vista estática formada por diagramas de clase que representan las clases importantes de la aplicación con sus responsabilidades (atributos y operaciones), y sus relaciones que se modelan en uno o varios diagramas de clases.

La vista dinámica muestra la interacción de esas clases en los casos de uso y se modela con diagramas de secuencia. Los diagramas de secuencia describen la interacción entre los objetos o instancias de las clases que intervienen en cada caso de uso. Muestra los eventos que ocurren en el tiempo que son enviados de un objeto a otro.

DirectShow es basado en filtros que se unen para formar una gráfica en la cual el flujo de datos pasa de un filtro a otro mediante contactos entre ellos. Los filtros pueden dividirse en cinco categorías:

- Fuente, introduce datos en la gráfica.
- Transformación, acepta un flujo de entrada, lo procesa y crea un flujo de salida.

- Destino, acepta un flujo y lo reproduce, lo guarda en el disco o lo transfiere a otra máquina.
- División, acepta un flujo y lo parte en varios flujos de salida, generalmente después de algún procesamiento.
- Unión, acepta varios flujos y los une en un flujo de salida, generalmente se usan para unir el flujo de audio con el de video.

En el sistema se usarán dos configuraciones de filtros: para la reproducción y para la captura. Los dos son similares, consisten de la fuente del video, un filtro de transformación de color si es necesario y un filtro de selección del área de captura. Pero el filtro destino es diferente: la ventana del reproductor o un filtro de almacenamiento en el disco.

Capítulo 5. Implementación y pruebas

Antes de la codificación se crea un prototipo de la interfaz de acuerdo a los casos de uso.

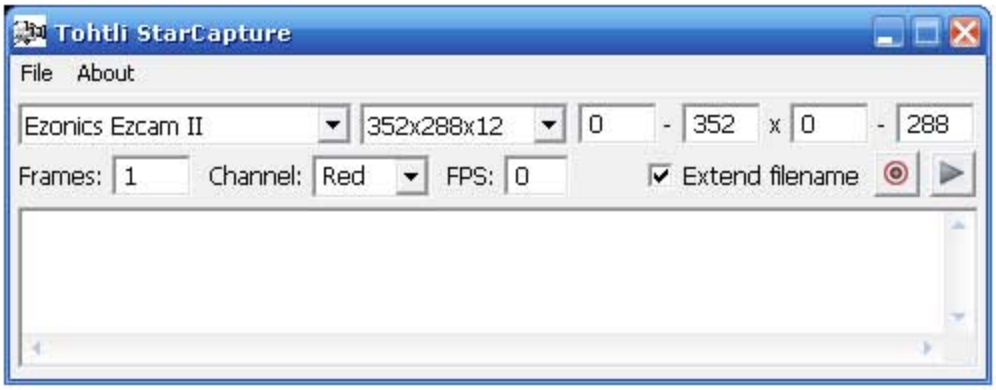

Fig. 5.2 La interfaz gráfica de Tohtli StarCapture

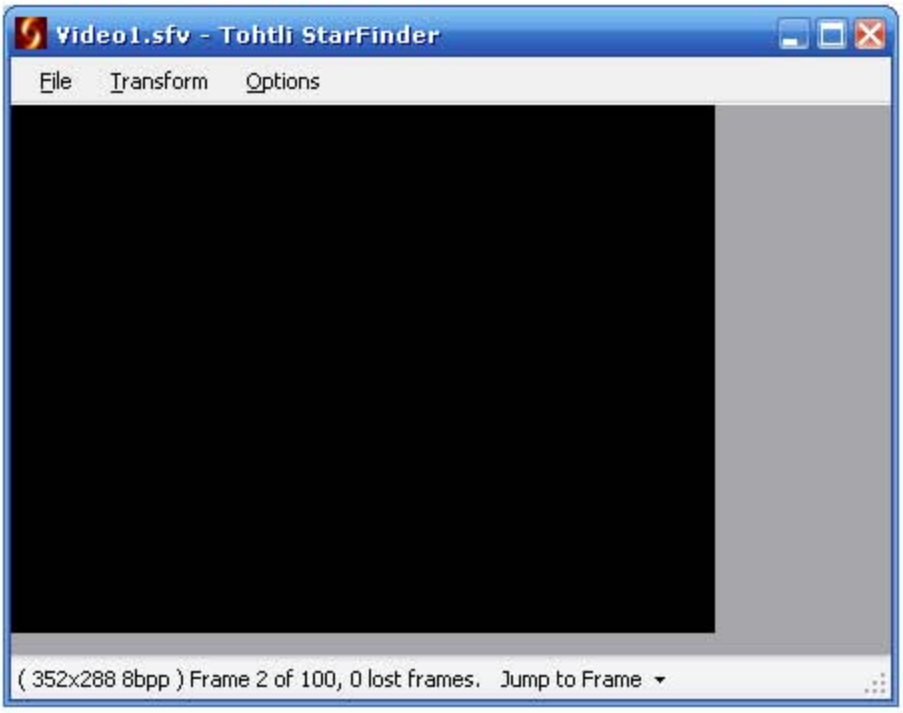

Fig. 5.1 La interfaz gráfica de Tohtli StarFinder

Durante la implementación se usó una técnica llamada **refactorización**. Es el proceso de modificación del código que no altera su comportamiento externo pero mejora su estructura interna. Es una manera disciplinada de limpiar el código que minimiza la posibilidad de errores. En esencia se mejora el diseño del código después de que sea escrito (Fowler, 1999).

El diseño preliminar del sistema nunca es completo, solo durante el desarrollo se puede descubrir la mejor manera de que funcione. Así que el proceso se vuelve iterativo alternando entre codificación y refactorización para obtener mayor calidad del código final.

5.1 Pruebas unitarias

Las pruebas unitarias consisten en que el programador vaya probando cada unidad de código que va terminando. En la orientación a objetos, el elemento unitario es la clase. Las clases se prueban a través de la creación de objetos o instancias de ellas para asegurar su calidad (Ibargüengoitia G. & Oktaba, 2006).

Para probar las operaciones o métodos de una clase se definen **casos de prueba** que son los valores de las variables que se usarán para comprobar que la operación está funcionando como se definió. Para cada valor del caso de prueba se indica el resultado que se espera que regrese la operación. Para cada método tiene que haber al menos una prueba correspondiente.

Si el resultado obtenido coincide con el esperado, la unidad se aprueba y se pasará a aplicarle otro tipo de pruebas. Si una unidad pasa todas las pruebas es aceptada. Si no, se corrigen los defectos encontrados y se aplican todas las pruebas de nuevo, para asegurar que no se introdujeron nuevos errores.

Para tener un grado adecuado de confiabilidad en un componente se deben hacer dos tipos de pruebas: las de caja blanca (transparente) y las de caja negra.

5.1.1 Pruebas de caja blanca

Las pruebas de caja blanca son también llamadas pruebas estructurales pues se trata de revisar que la estructura del código funcione adecuadamente, pasando por todas las instrucciones por los llamados caminos de control. Para eso se definen valores de prueba de tal forma que se ejecuten por lo menos una vez cada una de las instrucciones del programa.

Para saber cuántos casos de prueba son necesarios para tener una buena cobertura de las instrucciones de una unidad, se puede usar la complejidad ciclomática de McCabe (McCabe, 1976). Por lo general la complejidad ciclomática de McCabe es el número de condiciones en la unidad más 1.

Estas pruebas fueron las más numerosas ya que se necesitaba lograr la máxima cobertura posible. Cada prueba elegía un camino de control de modo que no se repitieran. Sirvieron para detectar errores causados por falta de atención al momento de la codificación y los errores causados por modificaciones del código.

5.1.2 Pruebas de caja negra

Estas pruebas también se conocen como pruebas funcionales ya que revisan que la unidad cumpla con la funcionalidad establecida. Los casos de prueba se diseñan a partir de los casos de uso y otros documentos de diseño, no se considera el funcionamiento interno de las unidades, se tratan como "cajas negras".

Muchas veces la unidad que se quiere probar requiere otras partes del código que aún no están disponibles. Para simular su comportamiento se declaran métodos sustitutos. Los sustitutos no hacen nada excepto regresar el valor esperado, pero ayudan a simular el comportamiento esperado.

Es una buena práctica definir todas las pruebas de caja negra antes de escribir el código. Se pueden usar para facilitar la implementación y verificar que las unidades funcionen correctamente en todo momento.

En el caso de este sistema las pruebas de caja negra fueron establecidas y aplicadas después de las pruebas de caja blanca, por lo tanto ayudaron a detectar menos errores que las anteriores. Sin embargo, los errores encontrados fueron en mayoría de tipo lógico, que no se pudieron detectar previamente.

5.2 Pruebas del sistema

En la prueba del sistema, se trata de comprobar que el sistema cumpla con todos los requerimientos establecidos, tanto los funcionales como los no funcionales.

Es consecuencia lógica del paso anterior. Solo que las pruebas son más extensas y completas.

Hay varios tipos de prueba del sistema: pruebas de integración aseguran que las unidades trabajen debidamente en conjunto, las pruebas funcionales son similares a las pruebas unitarias de caja negra pero a mayor escala, las pruebas de usabilidad aseguran que todas las funciones son accesibles y las pruebas de desempeño estiman los requerimientos mínimos de hardware y aseguran el correcto funcionamiento bajo condiciones extremas.

Para la definición de las pruebas del sistema solo se usó el texto original del problema y los casos de uso. Estas pruebas son menos numerosas y tienen carácter más informal que las pruebas unitarias, ya que no todas se pueden aplicar de manera automática, por ejemplo las pruebas de usabilidad y accesibilidad solo pueden ser realizadas por un ser humano.

Como se realizaron adecuadamente todas las pruebas unitarias, las pruebas funcionales y de integración no detectaron muchos errores. Pero sí sirvieron para comprobar la calidad del sistema.

Las pruebas de desempeño fueron aplicadas en el equipo que será usado por el cliente. Las características de ese equipo fueron tomados como requerimiento mínimo, ya que no se contaba con un equipo de configuración más cercana al límite mínimo teórico.

Conclusiones

Este trabajo es parte del proyecto estudiantil Tohtli. Mi tarea era la creación del software para el interferómetro de motas Tohtli. Como resultado se produjo un software que cumple con todos los requerimientos:

Tohtli StarCapture puede mostrar el video de cualquier fuente de video digital, conectada y configurada en el equipo y almacenarlo en el disco duro. También permite escoger la parte rectangular del video que será capturada y el canal de color deseado. La captura del video se hace en forma eficiente para evitar la pérdida de imágenes. Antes de la captura se especifican el número de cuadros a capturar y un comentario. Estos datos son almacenados junto con el video.

Tohtli StarFinder puede abrir el video almacenado, aplicar algunas acciones de análisis, mostrar el resultado y guardarlo en el disco duro. También puede exportar imágenes individuales del video capturado en formato PNG y reportar las imágenes perdidas durante la captura

Las acciones de análisis son: cálculo de la autocorrelación, eliminación de ruido de cámara, acumulación de imágenes, desplazado-y-sumado o desplazado-y-sumado selectivo de las imágenes.

El software fue probado en condiciones de laboratorio con el equipo que cuenta con la siguiente configuración:

- Sistema operativo: Windows XP SP2 Professional Edition
- Memoria RAM: 1GiB DDR400
- CPU: Intel Celeron de 2Ghz
- Disco duro: 80GB a 7800RPM

La fase de mantenimiento empezará cuando se utilice el sistema en el observatorio de Tonanzintla, Puebla. Si se encuentran errores podrán ser corregidos en el lugar de observación, ya que el equipo contará con el código fuente y los instrumentos necesarios para hacer las correcciones.

Referencias

Babcock, H. W. (1953). The Possibility of Compensating Astronomical Seeing. *Publications of the Astronomical Society of the Pacific , 65* (386), 229-236.

Beckers, J. (1982). Differential Speckle Interferometry. *Journal of Modern Optics , 29* (4), 361 - 362.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language User Guide.* (J. C. Shanklin, Ed.) Reading, Massachusetts, USA: Addison-Wesley.

Born, M., & Wolf, E. (1999). *Principles of Optics* (7th ed.). New York: Pergamon Press.

Endres, A., & Rombach, D. A. (2003). *Handbook of Software and Systems Engineering. Empirical Observations, Laws and Theories* (1st ed.). New York: Pearson/Addison-Wesley.

Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code.* Addison-Wesley.

Fried, D. L. (1966). Optical Resolution Through a Randomly Inhomogeneous Medium for Very Long and Very Short Exposures. *Journal of the Optical Society of America , 56* (10), 1372-1379.

Gerchberg, R. W., & Saxton, W. O. (1972). A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik , 35*, 237–246.

Gómez Gallegos, A. A. (2007). *Diseño y manufactura en la instrumentación astronómica: interferómetro de motas.* Universidad Nacional Autónoma de México, Facultad de Ingeniería, México.

Goodman, J. W. (1975). Laser Speckle and Related Phenomena. (J. C. Dainty, Ed.) *Topics in Applied Physics , 9*, 9.

Grieger, F. (1988). Speckle Masking with Coherent Arrays. In F. Merkle (Ed.), *ESO Conference and Workshop Proceedings*, *29*, p. 581. Garching bei München, Germany.

Humphrey, W. S. (2000). *Introduction to Team Software Process.* Addison-Wesley.

Ibargüengoitia G., G., & Oktaba, H. (2006). *Ingeniería de Software Pragmática.* México: Facultad de Ciencias UNAM.

Knox, K., & Thompson, B. (1974). Recovery of images from atmospherically degraded short-exposure photographs. *Astrophysical Journal , 193* (2), 45-48.

Labeyrie, A. (1970). Attainment of Diffraction Limited Resolution in Large Telescopes by Fourier Analysing Speckle Patterns in Star Images. *Astronomy and Astrophysics , 6*, 85-87.

Liu, Y. C., & Lohmann, A. W. (1973). High resolution image formation through the turbulent atmosphere. *Optics Communications , 8*, 372-377.

McCabe, T. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering , SE-2* (4), 308-320.

Orlov, V. G., Voitsekhovich, V. V., Sánchez, L. J., & Garfias, F. (2007). First Speckle Interferometry Measurements of Binary Stars at the OAN-Tonantzintla. *Revista Mexicana de Astronomía y Astrofísica , 43* (1), 137-140.

Royce, W. (1970). Managing the Development of Large Software Systems. *IEEE WESCON* (pp. 1-9). Piscataway, N.J.: IEEE Press.

Ryle, M., & Hewish, A. (1960). *Monthly Notices of the Royal Astronomical Society , 120* (6), 220–230.

Sommerville, I. (2002). *Ingeniería de Software* (6ª ed.). Addison-Wesley.

Tippens, P. (1985). *Física. Conceptos y aplicaciones.* México: McGraw Hill.

Voitsekhovich, V. V., Sánchez, L. J., Orlov, V. G., Garfias, F., & Benítez, R. (2005). First Test of the DRAGON Equipment. *Revista Mexicana de Astronomía y Astrofísica , 41* (2), 399-405.

Voitsekhovich, V., Orlov, V., Sanchez, L., & Garfias, F. (2007). Investigations of 1-m telescope guiding at the OAN - Tonantzintla. *Revista Mexicana de Astronomía y Astrofísica , 43* (2), 309-313.

Weigelt, G. P. (1977). Modified astronomical speckle interferometry 'speckle masking'. *Optics Communications , 21*, 55-59.

Worden, S. P., Lynds, C. R., & Harvey, J. W. (1976). Reconstructed images of Alpha Orionis using stellar speckle interferometry. *Optical Society of America, Journal , 66* (11), 1243-1246.