



UNIVERSIDAD SIMÓN BOLÍVAR

Ingeniería de la Computación

Teleoperación de un Robot Bombero para Evaluación de Situaciones de
Riesgo

Por

Tomas Lampo y Javier Sierra

Proyecto de Grado

Presentado ante la Ilustre Universidad Simón Bolívar
como Requerimiento Parcial para Optar el Título de
Ingeniero en Computación

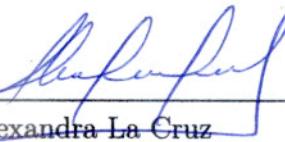
Sartenejas, Octubre de 2008



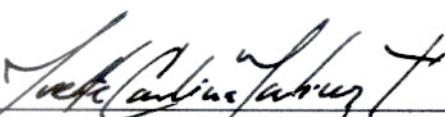
UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN
ACTA FINAL DEL PROYECTO DE GRADO
TELEOPERACIÓN DE UN ROBOT BOMBERO PARA EVALUACIÓN
DE SITUACIONES DE RIESGO

Presentado Por:
TOMAS LAMPO Y JAVIER SIERRA

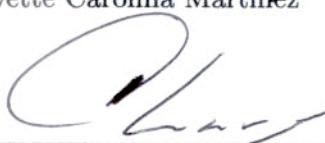
Este proyecto de Grado ha sido aprobado por el siguiente jurado examinador:



Prof. Alexandra La Cruz



Prof. Iyette Carolina Martínez



Prof. Carolina Chang (Tutor Académico)

SARTENEJAS, 26 de septiembre de 2008

Teleoperación de un Robot Bombero para Evaluación de Situaciones de
Riesgo

Por

Tomas Lampo y Javier Sierra

RESUMEN

Para este trabajo de investigación se diseñó e implementó una aplicación que permite teleoperar un robot de Búsqueda y Rescate a través de Internet. Ello hace posible que un equipo de rescate evalúe una zona de riesgo de manera remota, manteniendo a sus integrantes fuera de peligro.

La aplicación realizada permite a cualquier equipo de rescate manejar un robot Inuktun VGTV y todas sus funciones, reproduciendo en tiempo real la señal de video recogida por el robot. También permite que la teleoperación sea monitoreada por expertos de distintas áreas, enviando a éstos la señal de video recogida, de modo que puedan aportar sus conocimientos al proceso de Búsqueda y Rescate sin necesidad de encontrarse en la zona de riesgo. De igual manera, se facilita la medición de la frecuencia respiratoria de las víctimas encontradas, con ayuda de una cámara termográfica.

La aplicación fue diseñada suponiendo la existencia de una cámara termográfica, instalada en el robot de Búsqueda y Rescate. El video térmico no sólo permite encontrar víctimas en la zona de riesgo con mayor facilidad; además es posible estimar la frecuencia respiratoria de las víctimas encontradas, lo cual ayuda a determinar la gravedad de su situación. Esta estimación suele hacerse de manera visual por algunos miembros del equipo de rescate. Sin embargo, en este trabajo se ofrecen 4 métodos para calcularla automáticamente: Método Binario con Umbralización por Mínimos Locales, Método Binario con Umbralización Otsu, Método de Costos de Histogramas no normalizados y Método de Costos de Histogramas Normalizados.

Experimentalmente se demostró que el Método de Costos de Histogramas Normalizados calcula la frecuencia respiratoria de una víctima con el menor porcentaje de error. El porcentaje de error obtenido es competitivo con el error de trabajos de investigación consultados [13, 7, 21]. Sin embargo, estos trabajos garantizan una serie de condiciones ambientales controladas, como la posición y buen estado de salud de la víctima y calibración de la cámara termográfica, para realizar la medición de la frecuencia respiratoria. Los métodos propuestos en este trabajo se adaptan a ambientes no controlados, lo cual los hace más robustos que los métodos propuestos por otros autores.

Índice general

Índice general	III
Índice de cuadros	VI
Índice de figuras	VII
Capítulo 1. Introducción	1
Capítulo 2. Marco Teórico	7
2.1. Robots de Búsqueda y Rescate	7
2.1.1. Antecedentes	7
2.1.2. Descripción Detallada del Robot	9
2.2. Navegación del Robot	11
2.3. Redes de Computadoras	13
2.3.1. Protocolos de Transporte de Internet	14
2.4. Manejo del Video tomado por la Cámara del Robot	18
2.5. Mediciones de Respiración con la Cámara Termográfica	20
2.5.1. Visión por Computadora	23
Capítulo 3. Diseño de la Aplicación	25
3.1. Arquitectura del Sistema	25
Capítulo 4. Implementación de la Aplicación	28
4.1. Manejo y envío de video	28
4.2. Interfaz de las Aplicaciones del Cliente Operador y del Cliente Monitor . . .	30
4.3. Medición de respiración en víctimas	31
4.3.1. Método Binario	33
4.3.2. Cálculo automático de umbral	34
4.3.3. Método de Costos	40

4.3.4. Otros Métodos	45
Capítulo 5. Experimentos y Resultados	46
5.1. Experimento I <i>Evaluación de la teleoperación del robot</i>	47
5.1.1. Hipótesis	47
5.1.2. Metodología utilizada	48
5.1.3. Análisis de resultados obtenidos	49
5.2. Experimento II <i>Precisión de los métodos propuestos para la medición de frecuencia respiratoria</i>	50
5.2.1. Hipótesis	50
5.2.2. Metodología utilizada	52
5.2.3. Análisis de resultados obtenidos	53
5.2.4. Casos Atípicos	56
Capítulo 6. Conclusiones y Recomendaciones	58
Bibliografía	62
Apéndice A. Paquetes de Control y Respuesta del Robot	64
Apéndice B. Manual de Usuario de la Aplicación	67
B.1. Aplicación Servidor	67
B.1.1. Configuración de la Aplicación	67
B.1.2. Interfaz de la Aplicación	68
B.1.3. Recomendaciones en Caso de Fallas	68
B.2. Aplicación Cliente Operador	69
B.2.1. Configuración de la Aplicación	69
B.2.2. Interfaz de la Aplicación	70
B.2.3. Recomendaciones en Caso de Fallas	74
B.3. Interfaz de la Aplicación del Cliente Monitor	75

B.3.1. Configuración de la Aplicación	75
B.3.2. Interfaz de la Aplicación	76
B.3.3. Recomendaciones en Caso de Fallas	77
Apéndice C. Métodos y Filtros Rechazados	79
C.1. Filtros de Laplace	79
C.2. Método de Canny	80
Apéndice D. Sujetos Analizados	81

Índice de cuadros

1.	Resumen de características del robot Micro–VGT ^V	11
2.	Asociación de funciones del robot a caracteres	13
3.	Costo de los histogramas de la Figura 10.	41
4.	Costo de los histogramas normalizados de la Figura 17.	43
5.	Resultados de mediciones de los métodos propuestos.	54
6.	Error porcentual total de aproximación con respecto a la medición real de cada método propuesto.	55
7.	Resultados de mediciones de los métodos propuestos sobre los casos atípicos.	56
8.	Estructura de un Paquete de Control	64
9.	Estructura de un Paquete de Respuesta	65
10.	Bits de Control 1	66
11.	Bits de Control 2	66
12.	Descripción de los Controles de la Viñeta <i>Search</i> del Cliente Operador	72
13.	Descripción de los Iconos de la Viñeta <i>Rescue</i> del Cliente Operador	74
14.	Descripción de los Iconos asociados al Gráfico del Cliente Monitor	77

Índice de figuras

1.	Robot Micro–VGTv de Inuktun. Vista lateral.	9
2.	Robot Micro–VGTv de Inuktun. Vista frontal.	10
3.	Robot Micro–VGTv de Inuktun y sus componentes. Vista superior.	11
4.	Modelo de paquete RTP	17
5.	Imágenes del video de las cámaras	21
6.	Diagrama de Módulos	26
7.	Interfaz de la Aplicación del Cliente Operador	30
8.	Imágenes asociadas a un ejemplo con 3 ciclos de respiración	31
9.	Imágenes recortadas de un ejemplo con 3 ciclos de respiración	32
10.	Histogramas correspondientes a las imágenes recortadas de un ejemplo con 3 ciclos de respiración	36
11.	Imágenes binarias de un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método de Mínimos Locales.	37
12.	Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método de Mínimos Locales.	37
13.	Algoritmo <i>Otsu</i> para el cálculo del Umbral	38
14.	Imágenes binarias de un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método Otsu.	39
15.	Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método Otsu.	40
16.	Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Costo de los histogramas calculado mediante el Método de Costos.	42
17.	Histogramas normalizados correspondientes a las imágenes recortadas de un ejemplo con 3 ciclos de respiración	44
18.	Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Costo de los histogramas normalizados calculado mediante el Método de Costos.	45

19.	Comportamiento del <i>Bit rate</i> de la red	49
20.	Ejemplos de cuadros de inhalación y exhalación del sujeto S_{30}	57
21.	Secciones de la Interfaz de la Aplicación Cliente Operador	70
22.	Controles de la viñeta <i>Search</i>	71
23.	Información de la viñeta <i>Rescue</i>	73
24.	Información de la viñeta <i>Rescue</i> durante una medición de respiración	73
25.	Secciones de la Interfaz de la Aplicación Cliente Monitor	76
26.	Imágenes antes y después de la aplicación del filtro de Laplace (Nariz Tibia)	79
27.	Imágenes antes y después de la aplicación del filtro de Laplace (Nariz Fría) .	80
28.	Imágenes antes y después de la aplicación de Canny	80

Capítulo 1

Introducción

En una situación de Búsqueda y Rescate, después de haber ocurrido algún desastre natural o el colapso de alguna estructura, el tiempo es muy valioso y la situación es bastante tensa. Los equipos de Búsqueda y Rescate Urbano intentan mantener la calma mientras que perros y rescatistas son introducidos en la zona del desastre, donde intentarán evaluar el estado de la estructura y rescatar víctimas aún con vida.

De acuerdo con Casper, Murphy y Micire [11, 6], la colaboración de robots en este tipo de acontecimientos es relativamente reciente y suele incomodar al equipo de rescate, así que se espera de ellos una respuesta lo suficientemente rápida y eficiente, que justifique el tiempo invertido en preparar el robot e introducirlo entre los escombros.

Los robots de Búsqueda y Rescate suelen ser teleoperados, dada la complejidad del entorno que debe ser percibido. En estos casos, el tiempo de funcionamiento del robot se divide entre la navegación y la percepción del entorno.

Generalmente, surgen complicaciones al intentar mover el robot y percibir su entorno a la vez. Esto se debe a la sensibilidad de la cámara a los movimientos del robot, produciendo una imagen ruidosa cuando éste se desplaza. Por lo tanto, la navegación del robot no puede ser autónoma, porque habría que detener el robot cada vez que se deseé utilizar la cámara.

Por otra parte, en este tipo de situaciones es preferible que un teleoperador humano tome las decisiones y manipule al robot en tiempo real, en lugar de utilizarse un robot inteligente o autónomo que probablemente fallaría al encontrarse con una situación para la cual no fue programado.

El equipo humano–robot es muy importante. Si el robot fuese autónomo, y su tarea consistiese en ir al campo, recopilar información y regresar, las probabilidades que no regrese y se pierda la información recopilada serían muy altas, como sugieren Carlson y Murphy [5].

Se podría decir que en este caso el robot es una herramienta cooperativa que proyecta

al humano dentro de la situación, para lo cual se requiere una buena movilidad y buenos sensores.

Se espera que un robot teleoperado suministre al operador realimentación sensorial del entorno mediante imágenes y diversas medidas. Ello permitiría llevar a cabo tareas en entornos no estructurados, así como otras tareas difíciles de programar.

De acuerdo con Casper [6] y Micire [10], aquellas ocasiones en las cuales se han utilizado robots de Búsqueda y Rescate, no se ha tenido éxito en encontrar víctimas aún con vida. Sin embargo, los robots han ayudado a los rescatistas a inspeccionar el estado de la infraestructura y detectar materiales tóxicos, además de ahorrarles personal y tiempo de búsqueda al explorar distintas áreas que podrían ser peligrosas y en las que podría no hallarse ninguna víctima con vida.

Al realizarse una búsqueda con un robot, no sólo se espera encontrar sobrevivientes. También se espera poder encontrar caminos a través de los escombros que sean de fácil excavación y acceso para los rescatistas, garantizando la seguridad y eficiencia de estos últimos. Los rescatistas esperan también encontrar los cuerpos de víctimas fallecidas, para recuperarlos una vez terminada la operación de rescate.

Generalmente, cuando ocurre una situación de riesgo, desplazarse hasta el lugar de los hechos toma mucho tiempo y a veces se hace imposible, si se bloquean las vías de comunicación o no se pueden obtener los permisos legales necesarios para acceder a la zona del desastre.

Por esta razón, se plantea la posibilidad de teleoperar un robot de búsqueda y rescate a través de Internet. Suponiendo que el equipo de rescate cuenta con una conexión a Internet, se espera que el robot pueda ser teleoperado de manera remota, de modo que pueda introducirse entre los escombros y evaluar la escena, aún si el teleoperador no se encuentra en el lugar de los hechos.

Teleoperar el robot a través de Internet también permite al equipo de rescate y al teleoperador mantenerse alejados de la zona del desastre. En este caso, el equipo de rescate podría introducir una computadora en la zona del desastre y conectarla al robot. Inmediatamente,

se podría teleoperar el robot mediante una conexión inalámbrica, protegiendo las vidas del equipo de rescate y del teleoperador en caso de un colapso estructural u otro incidente, y permitiendo su acceso en espacios con dimensiones reducidas, por los cuales no pudiese pasar un rescatista.

Adicionalmente, de acuerdo con Casper, Murphy y Micire [11, 6], se recomienda que la búsqueda sea monitoreada por varias personas, de modo que si se nota algún detalle importante, éste sea examinado de inmediato.

En un escenario de desastre, como explica Casper [6], el tiempo de uso del robot no supera los 45 minutos, dado que el tiempo es muy valioso y se estaría postergando la operación de rescate. Una vez transcurrida ese tiempo, se cuenta sólo con la información recopilada por el robot para tomar decisiones importantes.

Para esta investigación se propone entonces la creación de una aplicación con una interfaz sencilla de utilizar, que permita a los usuarios no sólo teleoperar un robot a través de Internet, sino además permitir que este proceso sea monitoreado por varias personas en distintos lugares y de esta manera poder contar con la opinión de expertos en diversos campos durante la evaluación de la situación. Esta aplicación debería ser capaz de proveer realimentación constante al usuario no sólo del video tomado por las cámaras del robot, sino también de los demás sensores del mismo.

Ahora bien, aún no ha surgido la ocasión en la cual algún robot haya encontrado alguna víctima con vida. Pero si llegara a ocurrir, lo ideal sería encontrar la manera de verificar los signos vitales de la víctima con un personal médico.

De acuerdo con Fei y Pavlidis [7, 18], en el presente existen métodos para monitorear el pulso de una víctima a distancia, así como también para medir su frecuencia respiratoria (cantidad de veces que inhala y exhala la víctima por minuto). La frecuencia respiratoria de una persona es uno de los signos vitales y, por lo tanto, es un indicador del estado de salud general del sujeto. Su medición permitiría calcular la gravedad de la situación en la que se encuentra la víctima, para poder tomar decisiones de importancia.

Lo ideal para esta investigación sería medir el pulso de las víctimas encontradas. Sin

embargo, como mencionan los autores [18], para medir el pulso es necesario encontrar la arteria carótida con una cámara termográfica, lo cual se dificulta cuando las víctimas de un desastre se encuentran en posiciones incómodas. Por ello, se eligió medir la frecuencia respiratoria de las víctimas encontradas, dado que resulta más sencillo posicionar el robot para ver la nariz de una persona, que la arteria carótida.

Cabe destacar que después de 48 a 72 horas de haber ocurrido el accidente, según Murphy [11], la probabilidad de mortalidad de los sobrevivientes comienza a aumentar rápidamente. Con la información recopilada por el robot se podría asignar distintas prioridades de rescate a las víctimas encontradas, reduciendo el número de fallecimientos en la zona del desastre.

Como menciona Murthy [13], existen métodos de contacto para medir la frecuencia respiratoria de un individuo. Estos incluyen medidores de temperatura nasal (introducidos en las fosas nasales), electrodos y sensores colocados en el individuo, bandas que se colocan en el abdomen para detectar movimiento, entre otros.

Sería muy complicado equipar a un robot con alguno de los medidores antes mencionados, algún brazo robótico u otra extremidad, ya que se dificultaría la teleoperación del mismo y además se podría incomodar o mover a la víctima a la hora de medir su frecuencia respiratoria.

Grenecker [9] introdujo un método para la medición de respiración de cuerpos en reposo que consiste en un radar que mide los movimientos del pecho del individuo. Sin embargo, de acuerdo con Fei y Pavlidis [7], este método no es robusto ante otros movimientos, lo cual resulta en mediciones erróneas de la frecuencia respiratoria. Por esta razón, estos autores recomiendan hacer las mediciones utilizando una cámara termográfica.

Al equipar al robot con una cámara termográfica, se espera poder medir la frecuencia respiratoria de las víctimas sin tocarlas, enviando el estimado al teleoperador y demás clientes conectados, para que puedan tomar decisiones de importancia.

Se espera que esta investigación provea de información útil al mundo de la búsqueda y el rescate de sobrevivientes en una situación de desastre, ayudando en un futuro a reducir el

número de víctimas fallecidas y proteger a los rescatistas ante posibles colapsos estructurales.

En resumen, los objetivos de este trabajo son:

- Diseñar una aplicación que permita teleoperar un robot de Búsqueda y Rescate a través del Internet.
- Permitir que la aplicación acepte conexiones de otras personas que deseen monitorear la búsqueda a través de Internet, de modo que expertos de distintas áreas puedan colaborar con sus conocimientos durante la búsqueda.
- Diseñar e implementar métodos para asistir a los grupos de rescate en la medición de la frecuencia respiratoria de víctimas encontradas.

Este trabajo se ha dividido en cinco capítulos. El Capítulo 2 presenta una introducción sobre las capacidades de Búsqueda y Rescate de los robots, sus antecedentes y anatomía. También se presentan en este capítulo los mecanismos utilizados para controlar al robot a través de la computadora ya existentes, algunos conceptos de visión por computadora y de redes de computadoras, y los mecanismos empleados para enviar y procesar video. En el Capítulo 3 se muestra una descripción del diseño de la aplicación. La implementación de dicha aplicación se describe en el Capítulo 4, especificándose las técnicas elegidas para el manejo y envío del video, los algoritmos diseñados para las mediciones de respiración y una descripción de la interfaz de la aplicación. El Capítulo 5 detalla los experimentos realizados, los resultados obtenidos y su análisis. Finalmente, el Capítulo 6 presenta las conclusiones y recomendaciones para trabajos futuros.

Capítulo 2

Marco Teórico

En este capítulo se profundiza en los robots de Búsqueda y Rescate. Se hace énfasis en sus antecedentes y se presenta al robot utilizado en esta investigación, junto a las técnicas empleadas para teleoperarlo a través de una computadora.

Posteriormente, se presentan algunos conceptos de redes de computadoras, seguidos de los métodos empleados para el envío de video por Internet en tiempo real y las técnicas para detectar la frecuencia respiratoria en víctimas, mediante el procesamiento del video de una cámara termográfica instalada en el robot.

2.1. Robots de Búsqueda y Rescate

A continuación se ofrece una breve reseña sobre los antecedentes de los robots de Búsqueda y Rescate. Luego se introduce el modelo de robot utilizado en esta investigación, con una descripción detallada de sus características más relevantes.

2.1.1. Antecedentes

De acuerdo con Casper [6] y Micire [10], los robots de Búsqueda y Rescate han sido mencionados en la literatura científica desde principios de la década de los ochentas del siglo XX. Sin embargo, la primera investigación real comenzaría después del bombardeo en Oklahoma City, en Estados Unidos, en 1995.

Después del bombardeo no se utilizaron robots de rescate, pero la Doctora Robin Murphy participó junto con un estudiante de posgrado en las jornadas de rescate, tomando notas de cómo podría aplicarse la robótica para asistir la búsqueda y rescate de víctimas.

Esta iniciativa tomaría luego dos direcciones: el desarrollo de un sistema experto para determinar cuáles robots existentes son adecuados para situaciones dadas [10] y la investigación de robots marsupiales [12] como una solución a distintos tipos de problema. Estos trabajos darían pie a otras investigaciones, así como a la concepción de varios prototipos de

robots de rescate, para llevar a cabo diversas tareas.

Desde entonces, se llevaron a cabo numerosas pruebas de campo, en las cuales se exploraron la interacción entre el humano y el robot en situaciones de riesgo simuladas. Estos experimentos contribuyeron significativamente al diseño de aplicaciones y modelos de robots que se adapten mejor a distintas situaciones de riesgo y que lleven a cabo distintos objetivos.

Los robots de Búsqueda y Rescate fueron introducidos a una situación real por primera vez en los días siguientes a la tragedia del 11 de septiembre de 2001 en el World Trade Center de New York. Estos robots fueron utilizados principalmente para

1. Búsqueda de víctimas.
2. Búsqueda de caminos entre los escombros, que facilitasen las excavaciones.
3. Inspección de estructuras.
4. Detección de materiales tóxicos.

En cada caso se utilizaron robots porque podían alcanzar profundidades inaccesibles por el equipo tradicional de búsqueda, así como pasar por espacios por los que no caben humanos o perros y penetrar áreas en llamas o al borde de un colapso estructural. Se debe resaltar que se utilizaron robots teleoperados, sin autonomía, dada la complejidad del terreno y las limitaciones de los sensores.

Estos robots lograron ver el área de desastre desde un punto de vista que jamás había sido posible apreciar por los rescatistas. Los robots indicaron a los rescatistas el tipo de señales o marcas que debían buscar entre los escombros, además de proveer una colección importante de imágenes y videos invaluables en caso de haberse encontrado sobrevivientes.

Los robots realizaron las tareas correctamente. No lograron localizar ningún sobreviviente, pero sí encontraron restos de las víctimas del atentado. En consecuencia, fueron bienvenidos a formar parte de la comunidad de rescate.

Entre los robots presentes entre los escombros del atentado, se encontraban los micro-Tracs y micro-VGTV fabricados por Inuktun, acompañados por los modelos Solem y Talon

fabricados por Foster-Miller. Estos robots serían los más adecuados para la tarea, por sus sensores y tamaño, entre otras características.

Desde entonces, una cantidad importante de investigadores [16, 10, 5, 6, 3] ha comenzado a concentrar sus esfuerzos en los robots de búsqueda y rescate, buscando mejorar los sensores, el diseño y la programación de los mismos, para mejorar su respuesta ante situaciones de desastre.

2.1.2. Descripción Detallada del Robot



Figura 1: Robot Micro-VGTV de Inuktun. Vista lateral.

El modelo del robot utilizado en esta investigación es *Micro Variable Geometry Tracked Vehicle* o *Micro-VGTV* (ver Figura 1), fabricado por Inuktun. Este fue prestado por La Universidad de Florida del Sur y CRASAR para esta investigación.

Este modelo cuenta con dos bandas de caucho de tracción tipo oruga que le permiten desplazarse hacia adelante y hacia atrás, así como girar sobre su propio eje, adaptándose a cualquier tipo de terreno. Su chasis es de aluminio molido, lo cual lo hace muy resistente.

Su cuerpo está diseñado de manera que pueda cambiar su geometría. Es decir, el robot puede erguirse o elevarse, y luego *acostarse* hasta retomar su posición horizontal original.

En su *cabeza* se encuentra una cámara a color con *zoom* incluído, que cuenta con dos

bombillos de luz halógena a sus lados, cuya intensidad de iluminación puede ser regulada, y un par de rayos láser que permiten detectar irregularidades en la superficie del terreno. La cabeza del robot se puede apreciar con detalle en la Figura 2. De acuerdo con Aníbal Ollero Baturone [3], tener un buen control sobre el lente de la cámara (*zoom*, apertura y campo de visión) resulta en una mejor percepción del entorno del robot, lo cual facilita la tarea que está llevando a cabo.

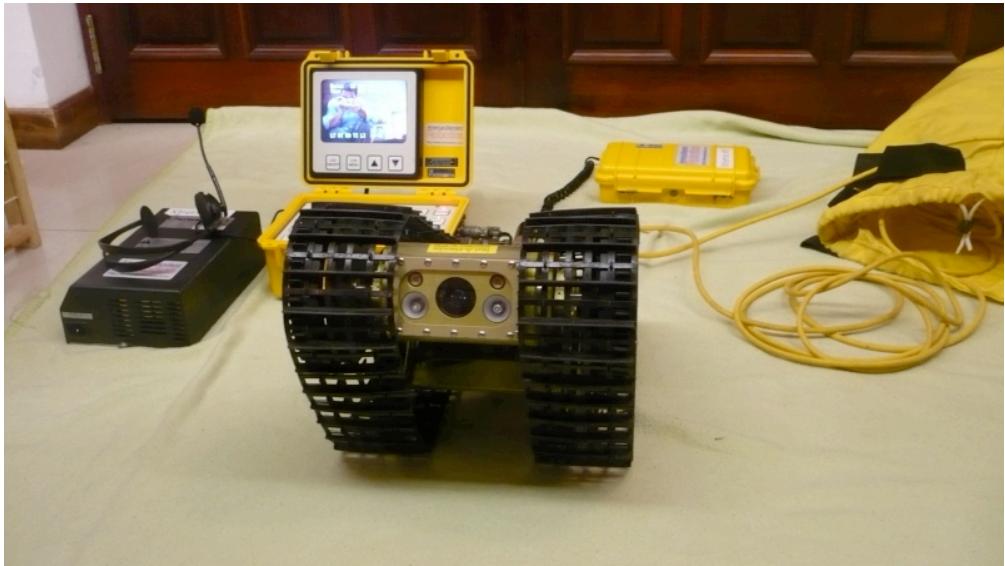


Figura 2: Robot Micro–VGTV de Inuktun. Vista frontal.

Esta cabeza puede girar hacia arriba y hacia abajo, alcanzando ángulos que faciliten al operador inspeccionar el área en la cual podría encontrarse el robot.

Cuando el robot está erguido, su cabeza queda más alta, permitiendo un campo de visión mayor al del robot acostado, sobre todo si se encuentra en un terreno con pendientes. Sin embargo, al estar erguido se hace bastante inestable, dado que eleva su centro de gravedad.

El robot es teleoperado mediante un control, conectado al robot a través de un cable de 30 metros de longitud. A este control se debe conectar la batería que presta corriente eléctrica al robot y al mismo control. El control cuenta con entradas para audífonos, micrófono, cable USB B y video compuesto.

En el Cuadro 1 se muestra un resumen de algunas de sus características.

Comunicación	Alámbrica (por medio de un cable de 30 m.)
Material de construcción	Aluminio
Tiempo de funcionamiento	6+ horas
Tamaño	15 × 30 × 30 (cm)
Velocidad	7.6 cm/s
Separación del suelo	2 cm
Tracción	Orugas
Capacidad de carga de sensores	Poca. Puede añadirse una cámara adicional.
Auto-enderezamiento	No. En caso que el robot se voltee accidentalmente, resulta complicado devolverlo a su posición de funcionamiento sin asistencia del teleoperador.

Cuadro 1: Resumen de características del robot Micro-VGTV

2.2. Navegación del Robot

Como se menciona en la subsección 2.1.2, el control de este robot se puede conectar a la computadora por medio de un cable USB, permitiendo la comunicación entre los dispositivos. En la figura 3 se muestran todos los componentes del equipo del robot.

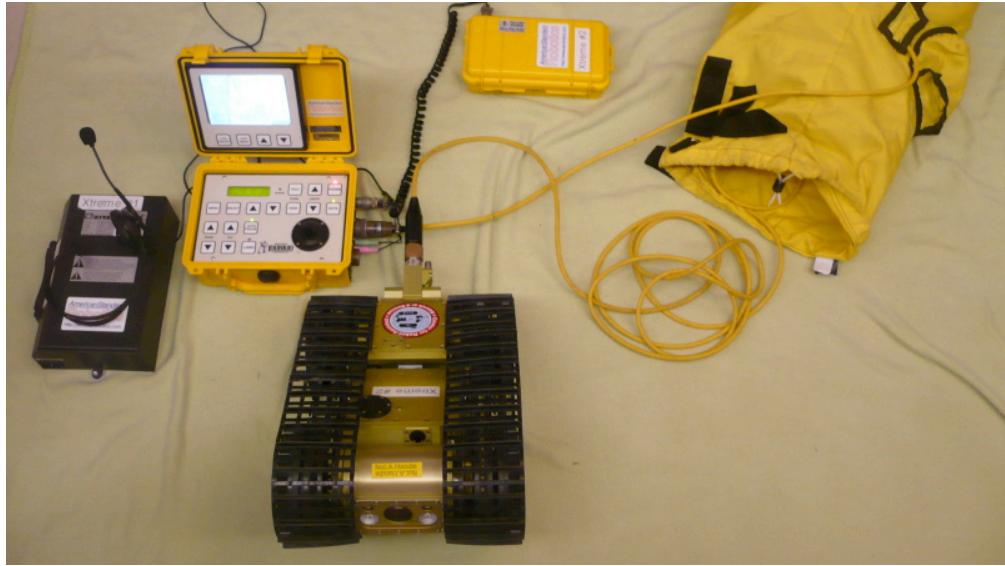


Figura 3: Robot Micro-VGTV de Inuktun y sus componentes. Vista superior.

La Universidad de Florida y CRASAR proveen una librería de funciones, escritas en el lenguaje de programación *Java*, que permite enviar comandos de navegación al robot,

así como también permite erguir y acostar al robot, girar su cámara hacia arriba y hacia abajo, encender y apagar las luces y los rayos láser, hacer zoom, y otras funciones. Esta librería también permite obtener información del estado del robot, como su temperatura, nivel de la batería y ángulo de inclinación o levantamiento.

Dicha librería de funciones, escrita por Jeff Craighead, envía al robot paquetes correspondientes a arreglos de *bytes*, en los cuales se encuentran los comandos que se desea que el robot lleve a cabo. A estos paquetes se les llama paquetes de control (ver Cuadro 8 del Apéndice A). También recibe paquetes correspondientes a arreglos de *bytes* del robot, en los cuales se encuentra la información de su estado actual. A estos paquetes se les llama paquetes de respuesta (ver Cuadro 9 del Apéndice A). Se debe resaltar que cada paquete cuenta con un *byte* de *checksum*, que corresponde a la suma matemática de todos los *bytes* del paquete. El *checksum* permite comprobar si un paquete ha llegado a su destino sin alteraciones.

Para facilitar el envío de comandos al robot, se asocian caracteres a las distintas funciones antes expuestas (ver Cuadro 2). Estos caracteres son luego utilizados para crear los paquetes de control que serán enviados al robot.

Para la elaboración de la aplicación Cliente–Servidor, se toman los distintos comandos provenientes de las acciones que quiera llevar a cabo el operador del lado del cliente, y se envían sus caracteres correspondientes por una conexión segura al servidor, donde luego son enviados al robot.

Carácter	Función
l	Atenuar intensidad de luces
o	Incrementar intensidad de luces
L	Encender/Apagar láser
a	Cruzar hacia la izquierda
s	Cruzar hacia la derecha
w	Avanzar
z	Retroceder
e	Girar cámara hacia arriba
d	Girar cámara hacia abajo
r	Erguir robot
f	Acostar robot
W	Encender luces de advertencia
-	Zoom (alejar)
+	Zoom (acercar)
*	Cambiar la velocidad de Zoom (rápido/lento)
<	Enfocar
>	Desenfocar
/	Encender/Apagar autofocus (enfoque automático)
?	Obtener el status de todas las funciones del robot

Cuadro 2: Asociación de funciones del robot a caracteres

2.3. Redes de Computadoras

Las comunicaciones en Internet se rigen bajo el protocolo de capa de red IP (Protocolo de Internet), como explica Tanenbaum [19].

Según el autor, para que la información pueda ser enviada a través de Internet, es necesario que se creen *sockets* tanto en la máquina de origen como en la máquina de destino. La creación de un *socket* corresponde a la creación de un nuevo punto terminal de comunicación. En otras palabras, las aplicaciones que necesiten enviar información a través de Internet deben escribir dicha información en los *sockets*. Una vez que la información ha llegado a su destino, puede ser leída en los *sockets* de la máquina de destino.

Cada *socket* tiene un número (dirección) asociado, que consiste en la dirección IP de

la máquina en la cual fue abierto, así como un número de 16 bits, que corresponde a un puerto local en esa máquina. Los números de puerto menores que 1024 se llaman puertos bien conocidos, y su uso se reserva para servicios estándar.

Al crear un *socket*, es necesario especificar el protocolo de transporte a ser utilizado para la comunicación. Los protocolos de transporte serán explicados en la subsección 2.3.1. Una vez creados los *sockets*, se procede a establecer la comunicación entre la máquina de origen y la máquina de destino, para luego comenzar a intercambiar información.

La información a ser enviada a través de Internet comienza por ser dividida en conjuntos de datos más pequeños, llamados datagramas. Su tamaño puede ser de hasta 64 *Kbytes*, pero en la práctica suelen ser de 1500 *bytes*. Estos datagramas son transmitidos por Internet, posiblemente fragmentándose en unidades más pequeñas en el camino, dado que pueden pasar por distintas redes con distintos *routers* o enrutadores. Las piezas son posteriormente reensambladas en la máquina de destino, donde se intentará armar de nuevo la información enviada desde la máquina de origen.

Un datagrama IP consiste en una parte de encabezado y una parte de texto, que corresponde a la información que se envía. El encabezado del datagrama suele contener información importante sobre dicho datagrama, como su número de identificación, la dirección IP de la máquina de origen y la de la máquina de destino, el protocolo de transporte de Internet al cual debe entregarse el paquete una vez que ha sido recibido, entre otra información importante.

2.3.1. Protocolos de Transporte de Internet

Los paquetes de información enviados por Internet suelen recorrer varios *routers* antes de llegar a su destino. Esto implica que recorren varias redes, y por lo tanto podrían perderse en el camino, o llegar con mucho retraso. Adicionalmente, los paquetes enviados podrían recorrer caminos distintos, lo cual no garantiza que lleguen en el mismo orden en el cual fueron enviados.

Surge entonces la necesidad de crear mecanismos especiales, llamados protocolos, que

puedan lidiar con este tipo de incomodidades, mejorando la comunicación entre las máquinas. Los protocolos más conocidos y utilizados son el Protocolo de Control de Transmisión (TCP por sus siglas en inglés) y el Protocolo de Datagramas de Usuario (UDP por sus siglas en inglés), descritos a continuación.

Protocolo de Control de Transmisión (TCP)

De acuerdo con Tanenbaum [19], TCP fue específicamente diseñado para proporcionar un flujo de *bytes* confiable de extremo a extremo a través de redes no confiables. Estas redes podrían tener diversas topologías, anchos de banda, retardos, tamaños de paquetes y otros parámetros.

El diseño de TCP se adapta dinámicamente a las propiedades de estas redes, sobreponiéndose a muchos tipos de fallas, lo cual lo convierte en el protocolo más utilizado en Internet.

Las conexiones mediante TCP son bidireccionales y sólo pueden establecerse entre dos máquinas a la vez. Es decir, TCP no permite la multidifusión de información.

El protocolo IP no garantiza que los datagramas serán entregados de manera apropiada, por lo que corresponde a TCP utilizar temporizadores y retransmitir los datagramas en caso que sea necesario. Los datagramas que llegan podrían hacerlo en el orden incorrecto, por lo que corresponde también a TCP reordenarlos en la secuencia apropiada.

Para llevar a cabo este tipo de tareas, el protocolo TCP agrega al datagrama a ser enviado un encabezado TCP de 20 *bytes* que contiene la información sobre el datagrama, como el número de puerto de origen y el puerto de destino de la conexión, el número de secuencia del datagrama, el número de confirmación de recepción, el espacio disponible para enviar información, entre otra información relevante.

Este encabezado sigue al encabezado IP, que a su vez sigue al encabezado Ethernet. A este encabezado le sigue la información a ser enviada, aunque el protocolo permite datagramas con información vacía, que corresponderían a confirmaciones de recepción y mensajes de control.

El protocolo TCP establece una comunicación entre la máquina de origen y la máquina de destino. Al enviarse un paquete de información, la máquina emisora envía un paquete

con información a la máquina receptora, que regresa un paquete indicando que ha recibido el paquete correctamente y el espacio disponible para seguir recibiendo información. Si TCP no recibe una confirmación después de transcurrido un tiempo prudencial, declara el paquete enviado anteriormente como un paquete perdido, y procede a reenviarlo.

En caso que la máquina receptora se quede sin espacio, la máquina emisora se bloquea hasta que recibe un paquete que le indique que el espacio ha sido liberado, y entonces reanuda el envío de información. Si no recibe ningún paquete con autorización para seguir enviando información, espera un tiempo razonable y envía un paquete de 1 *byte*, esperando que el receptor responda que ha recibido dicho paquete y haya liberado el espacio suficiente para continuar el intercambio de información.

Cuando surge una congestión en la red, si la carga ofrecida es mayor a la que puede manejar, el protocolo TCP se encarga de disminuir la tasa de datos que envía, alargando la frecuencia con la cual envía los paquetes. De este modo se busca reducir la carga de la red y aliviar la congestión.

Según Fiorini y Oboe [8], el protocolo *TCP* provee ventajas como manejo automático de errores, retransmisión, reordenamiento de paquetes y garantía de llegada de todos los paquetes. Lamentablemente, este protocolo tiene la desventaja de tener tiempos de envío y llegada poco predecibles. Esta limitación puede ser corregida utilizando protocolos como *UDP*, que será explicado a continuación.

Protocolo de Datagramas de Usuario (*UDP*)

A diferencia de TCP, UDP no es un protocolo orientado a la conexión. UDP proporciona una forma para que las aplicaciones envíen datagramas IP sin tener que establecer una conexión, como explica Tanenbaum [19]. Adicionalmente, UDP permite la multidifusión de información, así que la misma información puede ser enviada a varias máquinas a la vez.

Al igual que TCP, UDP agrega un nuevo encabezado al datagrama. Dicho encabezado, de 8 *bytes* de tamaño, sólo transporta información del puerto de origen y el puerto de destino.

UDP no realiza control de flujo, control de errores o retransmisión cuando se pierde un paquete. Todo lo anterior le corresponde a las aplicaciones que lo utilicen. UDP simplemente

provee la interfaz necesaria para enviar un paquete desde la máquina de origen hacia la máquina de destino, sin asegurar su llegada. De este modo, se aligera notablemente la carga de la red y la comunicación es mucho más rápida, lo cual permite utilizar este protocolo para las aplicaciones multimedia en tiempo real.

La mayoría de los programadores de aplicaciones multimedia en tiempo real concibieron el mismo protocolo de transporte de tiempo real. Eventualmente se hizo evidente que era necesario tener un protocolo de transporte genérico para este tipo de información, lo que dio nacimiento al Protocolo de Transporte en Tiempo Real (RTP por sus siglas en inglés).

Protocolo de Transporte en Tiempo Real (RTP)

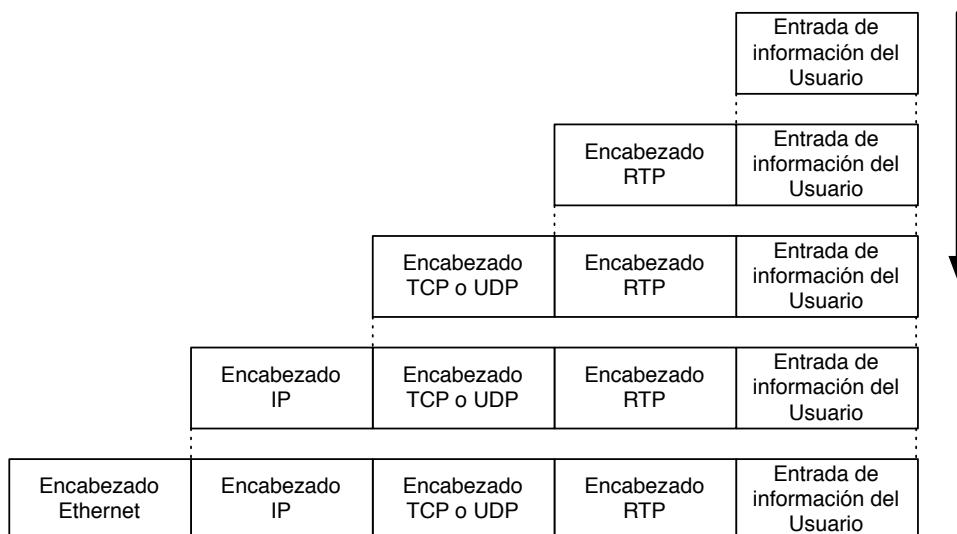


Figura 4: Modelo de paquete RTP

La posición del RTP entre los protocolos es incierta, dado que la implementación de este protocolo se hace en la aplicación que lo utiliza y su ejecución se hace generalmente sobre UDP. Es por ello que los paquetes RTP no son tratados de manera especial por los *routers*, a menos que se habiliten ciertas características de calidad de servicio IP.

A cada paquete RTP enviado se le agrega una numeración mayor a la de su predecesor, lo cual permite a la máquina destino determinar si falta un paquete. Si falta alguno, la máquina destino puede intentar aproximar el valor faltante por medio de interpolación, o simplemente

congelar la reproducción multimedia que se está llevando a cabo.

RTP agrega un nuevo encabezado a los datagramas, como se muestra en la figura 4. En esta figura se pueden apreciar todos los encabezados que se agregan a la información del usuario cuando se crea un paquete RTP. Primero se agrega el encabezado RTP, seguido de los encabezados TCP o UDP, a los cuales se añade el encabezado IP para finalmente colocar el encabezado Ethernet al paquete y enviarlo a través de Internet.

El encabezado RTP incluye la numeración antes mencionada, junto con un *timestamp* que corresponde al momento exacto en el cual debe reproducirse la información contenida en el datagrama, además de otra información útil. Los *timestamps* son relativos al inicio del flujo de la información, lo cual permite que el destinatario haga un pequeño almacenamiento antes de comenzar la reproducción y reproduzca cada muestra el número exacto de milisegundos después del inicio del flujo multimedia. Esto permite que múltiples flujos de información, como audio y video (provenientes de dispositivos físicos distintos), se mantengan sincronizados entre sí, incluso si los flujos son transmitidos irregularmente.

2.4. Manejo del Video tomado por la Cámara del Robot

El control del robot cuenta con una salida de video compuesto analógico, para conectores RCA (por convenio de color amarillo).

Al contar con una computadora equipada con una tarjeta de video, una capturadora con una entrada para conector RCA, o un dispositivo que transforme la señal de video a USB, se puede conectar la cámara del robot a la computadora, y por lo tanto se podría intentar transmitir y guardar una copia de la señal recogida.

Para la elaboración de la aplicación Cliente–Servidor, se podría intentar leer cada cuadro del video como un arreglo de *bits*, y enviar dicho arreglo por una conexión TCP o UDP, pero ello haría que el video no se mantenga en tiempo real, lo cual dificultaría el control del robot.

Para manejar la captura y el envío del audio y del video de la cámara del robot, se decidió entonces utilizar la librería *Java Media Framework* (JMF) para el lenguaje de pro-

gramación *Java*, intentando mantener todo el código del trabajo en este lenguaje, principalmente porque su portabilidad permitiría que la aplicación funcione en cualquier plataforma, reduciendo la necesidad de adaptar el código para cada sistema operativo disponible en el mercado.

La librería JMF permite enviar y recibir *streams* de video casi en tiempo real, manejando todo el procedimiento de compresión y manipulación de los paquetes.

La arquitectura de JMF está compuesta por múltiples *capas* que interactúan entre sí. El primer componente del diseño JMF es la fuente de datos, esta representa los datos multimedia que se desean manipular (video o audio). La fuente de datos maneja el encapsulamiento, ubicación y protocolos utilizados para transmitir el contenido multimedia.

Las fuentes de datos pueden clasificarse, de acuerdo al modo en que se da inicio a la transferencia de datos, en dos categorías:

1. **PushDataSource:** Son aquellas fuentes de datos donde el servidor inicia la transferencia de datos. Algunos ejemplos de este tipo de fuente de datos son transmisiones multimedia y video por demanda, como los utilizados en esta investigación.
2. **PullDataSource:** Son aquellas fuentes de datos donde el cliente inicia la transferencia. Los protocolos *HTTP* y *FILE* son ejemplos de esta categoría.

Los datos multimedia pueden obtenerse de múltiples fuentes, como archivos locales o de red y dispositivos físicos.

El segundo componente del modelo JMF es la representación de los dispositivos de captura de datos (micrófonos, cámaras y otros). Es sencillo convertir un dispositivo de captura en una fuente de datos y utilizar los demás componentes de JMF para procesar la información capturada.

Otro componente importante de esta arquitectura es el Reproductor, el cual recibe los datos de audio o video y los representa a través del monitor o dispositivo de audio disponible. Este componente está representado por una máquina de estados que maneja los siguientes estados:

- **No Realizado:** En este estado el Reproductor desconoce su ambiente y datos, así como el tipo de datos que manejará.
- **En Realización:** Un Reproductor cambia a este estado mientras intenta determinar los recursos que necesita para manejar los datos presentados (codecs y drivers).
- **Realizado:** Un Reproductor cambia a este estado cuando logra determinar los recursos que necesita para manejar los datos presentados. Una vez en este estado el Reproductor es capaz de proveer componentes visuales y de control para los datos manejados.
- **Iniciado:** En este estado el Reproductor es capaz de presentar los datos multimedia.

Existe una extensión para el Reproductor que provee control adicional sobre los datos. Esta extensión es el componente Procesador que, además de manejar todos los estados de un Reproductor, puede crear fuentes de datos nuevas y enviar datos a través de las mismas. Dichas fuentes de datos pueden ser manejadas por otros Reproductores o Procesadores.

En primera instancia, se establece el origen de los datos a través de la designación de un dispositivo de captura u otra fuente de datos como archivos multimedia o *streams* en Internet.

Una vez definida la fuente de los datos, se crea el Reproductor o Procesador, de manera que sea posible manejar la información. La plataforma de JMF provee una implementación del protocolo RTP, lo cual permite transmitir por Internet los datos de audio y video capturados. El protocolo RTP ha sido descrito en la sección 2.3.

En la sección 4.1 se especifica cómo se llevó a cabo el envío de video y audio a través de Internet, para la aplicación diseñada para este trabajo.

2.5. Mediciones de Respiración con la Cámara

Termográfica

Para esta investigación se cuenta con una cámara termográfica no enfriada (que funciona a temperatura ambiente), que permite calcular el perfil de calor de cuerpos midiendo la

intensidad de los rayos infrarrojos proyectados por los mismos.



(a) Cámara del Robot



(b) Cámara Termográfica

Figura 5: Imágenes del video de las cámaras

Con la cámara termográfica se obtiene una imagen en escala de grises, en la cual el color blanco corresponde a la temperatura más alta (cuerpos más calientes), y el color negro a la temperatura más baja (cuerpos más fríos). En la Figura 5 se muestran dos imágenes. La imagen 5(a) fue tomada por la cámara del robot. La imagen 5(b) fue tomada por la cámara termográfica, colocada sobre la cabeza del robot.

Para medir la frecuencia respiratoria de un sobreviviente, se coloca la cámara termográfica de modo que se puedan ver con claridad los orificios nasales. Al respirar, el aire expulsado suele ser de una temperatura similar a la temperatura corporal de la víctima. Por otro lado, el aire inhalado al respirar tiende a estar a temperatura ambiente (frío). Por lo tanto, cuando la víctima inhala sus orificios nasales se muestran de color negro o gris oscuro en la imagen térmica, y cuando exhala sus orificios nasales se muestran de un color similar al que se muestra su piel.

Entre la bibliografía consultada no se pudo encontrar evidencia de la existencia de trabajos relacionados al de esta investigación. Sin embargo, existen trabajos de medición y detección de respiración, que fueron utilizados para desarrollar los algoritmos diseñados en esta investigación. Uno de estos trabajos, conducido por Murthy [13], consiste en medir la respiración de individuos ubicados de perfil con la ayuda de una cámara termográfica. Esta

cámara capta el aire caliente expulsado por el individuo, calificándolo como exhalación.

Este experimento funciona mientras que el individuo esté situado junto a una pared u otra superficie que se mantenga fría, de modo que se logre ver el contraste del vapor expulsado por el individuo.

El algoritmo de este experimento consiste en:

1. Crear una diferencia marcada entre el individuo y la pared del fondo. Esta diferencia se consigue por medio del *Método Otsu para el cálculo automático del umbral* que será detallado en la sección 4.3.2.
2. Resaltar el vapor expulsado por el individuo, agregándole coloración.
3. Cómputo de la frecuencia respiratoria del individuo por medio de métodos estadísticos que determinan si el individuo está inhalando o exhalando, modelando los ciclos respiratorios como distribuciones multi-normales.

Otro trabajo consultado es el conducido por Zhu, Fei y Pavlidis [7, 21], que consiste en medir la respiración de individuos ubicados de frente a una cámara termográfica.

Estos experimentos también requieren de un ambiente muy controlado, en el cual se regula la temperatura de la habitación y se calibra la cámara termográfica para medir temperaturas entre 28°C y 38°C, que corresponden a las temperaturas extremas que puede tomar la cara de un individuo. Dichos experimentos fueron conducidos sobre 20 individuos sanos, situados a exactamente 2 metros de la cámara termográfica.

En este trabajo se intenta medir la frecuencia respiratoria de víctimas estando frente a éstas, de modo que el contraste a medir sea relativo a la temperatura de la piel de la persona, en lugar de depender de estructuras externas. Esto se tratará con más detalle en la sección 4.3.

Se debe recordar que las situaciones en las cuales se prevé el funcionamiento de la aplicación son ambientes no estructurados, lo cual impide garantizar ciertas condiciones de uso, como sucede en los experimentos antes mencionados.

La frecuencia normal de respiraciones por minuto suele estar entre 12 y 20 ciclos¹ para un adulto en reposo. Sin embargo, en una situación de peligro algunas víctimas podrían aumentar o disminuir la cantidad de respiraciones por minuto.

Para poder calcular un promedio de la frecuencia respiratoria de la víctima, se hace necesario recurrir a técnicas de visión por computadora que permitan procesar la imagen. Algunas de estas técnicas se describen a continuación.

2.5.1. Visión por Computadora

De acuerdo con Shapiro [17], la *Visión por Computadora* consiste en tomar decisiones útiles sobre objetos físicos reales y escenas basadas en imágenes obtenidas mediante sensores. La meta de la visión por computadora es construir la descripción de una escena gracias a un conjunto de imágenes.

Dado que la cámara termográfica empleada proporciona imágenes en escala de grises, se decidió emplear la *técnica del umbral para convertir la imagen en escala de grises a una imagen binaria*², como propone Shapiro [17].

Esta técnica permite eliminar la profundidad en la imagen, separando los objetos en dos capas: los objetos de interés en la capa superior y los demás objetos en la capa de fondo. De este modo, se conservan los objetos de interés como figuras negras sobre fondo blanco.

La técnica del umbral para convertir imágenes en escala de grises a imágenes binarias consiste en elegir un *umbral* t , de modo que se asigne blanco (255) a todos los píxeles cuyo valor sea mayor que t y negro (0) a los píxeles restantes.

La selección del umbral puede hacerse de manera interactiva por un usuario. Sin embargo, cuando se tiene un proceso de análisis automático de imágenes, se hace necesario computar el umbral automáticamente.

Para seleccionar el umbral, suele utilizarse el *histograma* de la imagen en tonos de grises.

Un histograma es una representación gráfica de los tonos de colores de una imagen, en

¹Un ciclo respiratorio corresponde a una inhalación y una exhalación

²Una imagen binaria es aquella en la cual el valor de sus píxeles es 0 (negro) o 255 (blanco), correspondiendo a una imagen en blanco y negro puros.

forma de barras. La superficie de cada barra es proporcional a la frecuencia de los valores representados. Existen histogramas que presentan los tonos de los colores rojo, verde y azul de una imagen a color, así como histogramas que representan los tonos de gris, en caso que se trate de una imagen en tonos de grises.

El histograma h de una imagen en tonos de grises I se define mediante la fórmula presentada en la ecuación 1, donde (r, c) corresponde al punto en la fila r y la columna c de la imagen I .

$$h(m) = |\{(r, c) : I(r, c) = m\}| \quad (1)$$

Donde m toma los valores $[0, 255]$ de la escala de grises.

Existen varios métodos probabilísticos para calcular el umbral automáticamente. El método más conocido y utilizado es el *Método Otsu para el cálculo automático del umbral* [17, 15]. Este método se describe con más detalle en la subsección 4.3.2.

Los conceptos y metodología presentados en este capítulo se utilizaron para diseñar la aplicación para teleoperar el robot a través de Internet y detectar la respiración de las víctimas encontradas. Los detalles del diseño de la aplicación se presentan en el Capítulo 3 y detalles de la implementación de la aplicación se presentan en el Capítulo 4.

Capítulo 3

Diseño de la Aplicación

En este capítulo se describe la manera como se ha organizado el sistema servidor–clientes, así como la interfaz de la aplicación. Para el diseño de la aplicación se tomaron en cuenta los conceptos y la metodología discutidos en el Capítulo 2.

3.1. Arquitectura del Sistema

La aplicación sigue una estructura cliente–servidor que permite manejar múltiples clientes simultáneamente. Existen dos modalidades de cliente: *operador* y *monitor*. El cliente monitor está limitado a observar el video que elija el cliente operador y no puede cambiar el estado físico del robot en ningún momento.

Teóricamente, la herramienta servidor puede manejar cualquier cantidad de clientes, sin embargo esta cantidad está limitada por la capacidad de la red disponible y la memoria del computador en el cual se ejecuta el servidor.

El sistema está diseñado en módulos independientes que interactúan entre sí (ver Figura 6). Los módulos que se presentan en la figura son:

- Cliente: Se encarga de establecer y manejar las conexiones TCP así como de crear y manejar los Receptores de datos multimedia del cliente de la aplicación.
- Receptor: Se encarga de establecer y manejar la recepción y reproducción de los datos multimedia enviados a través de la conexión RTP.
- Servidor: Maneja las conexiones de los múltiples clientes (operadores o monitores) y establece la creación de los transmisores de video y, en el caso de los operadores, los controladores del robot. También se encarga de las mediciones de la respiración, utilizando la cámara termográfica.

- Transmisor: Se encarga de la captura de datos de las fuentes multimedia y su retransmisión a los distintos clientes.
- Control del robot: Procesa los paquetes enviados por un cliente al servidor y los envía al Manejador de Paquetes para su posterior conexión con el robot. De igual manera, se encarga de el procesamiento y envío de la información del robot como nivel de batería, inclinación del robot, entre otros.
- Manejador de Paquetes: Controla la comunicación bidireccional con el Robot a través de los módulos Controlador de Puerto Serial y Codificador de Paquetes de Control.
- Controlador de Puerto Serial: Maneja la comunicación a través del puerto serial. Controla el flujo de información y la codificación de los paquetes.
- Codificador de Paquetes de Control: Convierte los comandos enviados por el cliente operador en paquetes interpretables por el robot con la estructura discutida en la sección 2.2.

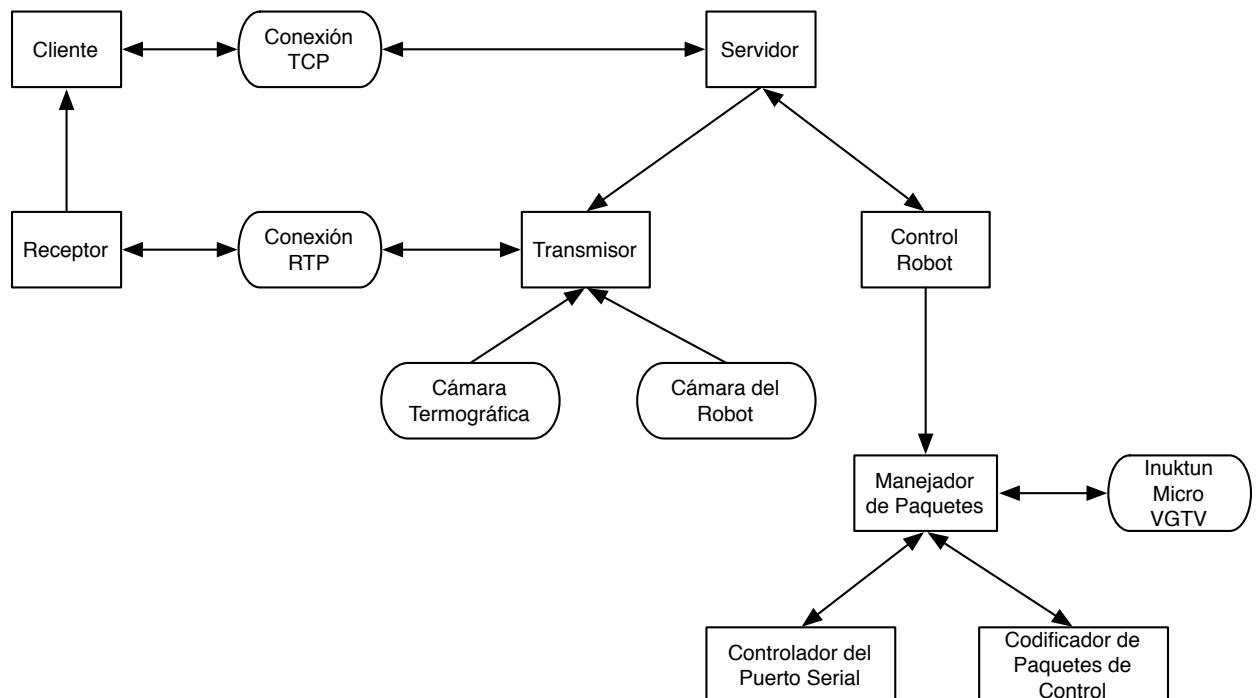


Figura 6: Diagrama de Módulos

Para adaptar el sistema a otro robot basta reemplazar el módulo que maneja la comunicación y genera los paquetes de control.

Como se describe en la sección 2.4 la secuencia de acciones tomadas por el sistema para iniciar una sesión son las siguientes:

1. El cliente operador establece una conexión TCP con el servidor y recibe el puerto al cual estará asociada la transmisión de video.
2. El servidor inicializa las estructuras necesarias para la transmisión de video y audio, así como las de control del robot.
3. El cliente operador inicia los protocolos de conexión para la recepción de video.
4. El servidor inicia la transmisión.
5. Se establece una comunicación TCP bidireccional para el envío de paquetes de control (cliente–servidor) y envío de datos (servidor–cliente).

Una vez iniciada la sesión con el cliente operador, el servidor espera la conexión de clientes monitores, de manera paralela. En caso que se conecte alguno de ellos, se inicia la siguiente secuencia de acciones:

1. El servidor duplica las estructuras existentes para la transmisión de video y audio. Este procedimiento será explicado con más detalle en la sección 4.1.
2. El cliente monitor inicia los protocolos de conexión para la recepción de video.
3. El servidor inicia la transmisión.
4. Se establece una comunicación TCP para el envío de datos (servidor–cliente).

Posteriormente, el servidor continúa esperando la conexión de clientes monitores.

En el Capítulo 4 se detalla la implementación de la aplicación y su interfaz, junto a los algoritmos diseñados propuestos para la medición de la frecuencia respiratoria de las víctimas encontradas.

Capítulo 4

Implementación de la Aplicación

En este capítulo se describen las decisiones de implementación tomadas y las distintas técnicas utilizadas para elaborar la aplicación de teleoperación del robot.

Es importante mencionar que toda la implementación del proyecto fue codificada en el lenguaje de programación *Java* de *Sun Microsystems* [1]. Dicho lenguaje fue elegido principalmente porque provee estructuras cómodas para escribir aplicaciones dirigidas al intercambio de información, de manera estable y segura, a través de redes de Internet, entre otras tareas que se deben llevar a cabo en esta investigación.

Adicionalmente, las aplicaciones escritas en el lenguaje de programación *Java* son conocidas por funcionar independientemente de la plataforma en la cual se están ejecutando. Por lo tanto, se espera que la aplicación diseñada pueda funcionar en cualquier sistema operativo y en cualquier plataforma, sin la necesidad de hacer cambios importantes en su código, lo cual no restringe a los futuros usuarios de la aplicación.

4.1. Manejo y envío de video

El procedimiento para iniciar la captura y transmisión de multimedia en JMF se lleva a cabo mediante la siguiente secuencia de pasos:

Primordialmente, es necesario crear las estructuras de datos abstractas que representan los dispositivos de captura o archivos a transmitir. Esta representación abstracta se conoce como el *DataSource*. La infraestructura de JMF introduce la creación de *DataSources* duplicables, lo cual permite que múltiples clientes puedan recibir la información de un mismo dispositivo al mismo tiempo.

Una vez definidas las estructuras de representación para los datos físicos, se deben crear los manejadores y procesadores de dicha información. JMF provee dos tipos de manejadores: el Reproductor y el Procesador, siendo el Reproductor básico multimedia el más sencillo.

El Reproductor provee una interfaz para el control de los datos multimedia basados en tiempo, así como una máquina de estados que permite manejar el estado global del reproductor.

El segundo manejador de datos provisto por JMF es el Procesador. Este manejador ofrece toda la funcionalidad del Reproductor, agregando algunas funciones adicionales para el manejo del procesamiento de datos y los flujos de salida de información.

Para poder transmitir los datos multimedia a través de la red es necesario que el manejador de datos sea un Procesador. Una vez creado el Procesador del *DataSource* se procede a establecer la conexión RTP.

La única condición para poder transmitir el video a varios clientes es que cada uno debe tener un puerto distinto asignado en el servidor. Para lograr esto, el servidor mantiene una cuenta de los clientes que se han conectado y para cada nueva petición asigna un puerto diferente, siguiendo la fórmula 2.

$$\text{Puerto} = \text{PuertoBase} + \text{Offset} \times \text{numeroDeClientes} \quad (2)$$

Donde el Puerto Base sería el puerto en el cual se comenzó la comunicación, y el *Offset* corresponde al número de puertos que se asigna a cada cliente.

Una vez asignado el puerto, el servidor procesa el duplicado de los DataSources pertinentes e inicia el envío de video. Los clientes tienen la potestad de pausar la reproducción en cualquier momento, a través de los controles tradicionales.

JMF permite la transmisión y recepción simultánea de varios *streams* de multimedia. Utilizando esta herramienta es posible que un cliente reciba información de audio y video al mismo tiempo.

El control para la captura de cuadros se lleva a cabo utilizando las clases que provee JMF para tal función. Una vez que se crea el controlador para la captura de cuadros, el análisis y manipulación de las imágenes se hace utilizando las librerías estándar de Java.

La calidad y tamaño del video también pueden ser parametrizados. Estos parámetros son los mismos para todos los clientes que se conecten en una sesión dada. Sin embargo, es

importante destacar que la calidad del servicio de envío de datos multimedia es inversamente proporcional a la multiplicidad de los clientes.

4.2. Interfaz de las Aplicaciones del Cliente Operador y del Cliente Monitor

La aplicación diseñada debería poder ser utilizada por personas en distintas partes del planeta. Es por ello que en la interfaz se incorporan imágenes, el lugar de texto, para indicar las distintas acciones que se pueden llevar a cabo con la misma.

Es importante mencionar que la interfaz ha sido elaborada con ayuda de *Java Swing* [2], que forma parte de las clases base del lenguaje *Java* de *Sun Microsystems*. Esta librería permite crear interfaces que se adaptan y asemejan a la interfaz de cada sistema operativo del mercado.

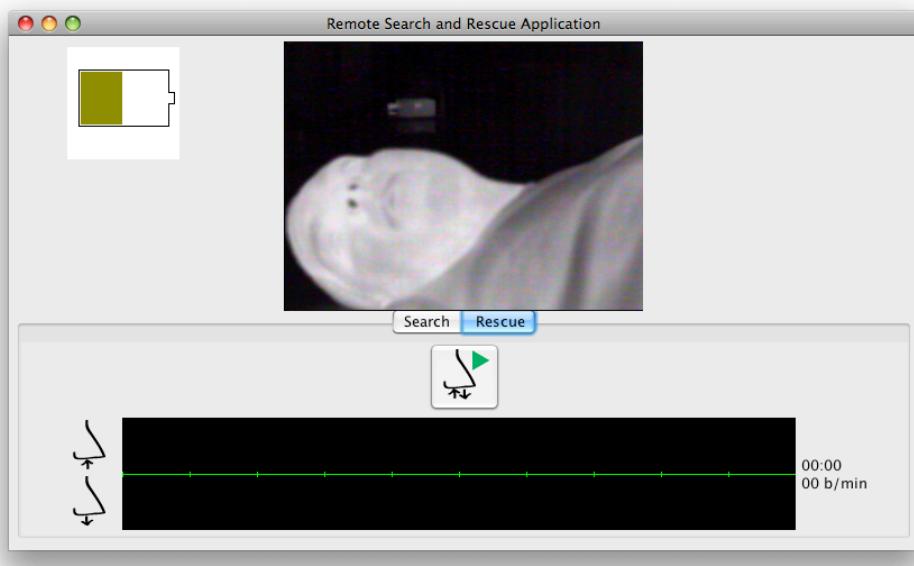


Figura 7: Interfaz de la Aplicación del Cliente Operador

La interfaz de las aplicaciones está dividida en dos secciones: una en la cual se coloca información importante sobre el status del robot y el video enviado por el mismo, y otra en la cual se colocan los controles del robot y medición de la respiración, cada uno en una viñeta

(Ver Figura 7). Estas secciones son explicadas con más detalle en el Manual de Usuario de la Aplicación, disponible en el Apéndice B.

La principal diferencia entre las aplicaciones del cliente Operador y el cliente Monitor radica en que la interfaz de la aplicación del cliente Operador le permite teleoperar al robot mediante los controles de teleoperación e iniciar mediciones de frecuencia respiratoria; mientras que la interfaz de la aplicación del cliente Monitor sólo muestra el video percibido por el robot (o la cámara térmica) y los resultados de las mediciones de frecuencia respiratoria.

4.3. Medición de respiración en víctimas

Con la ayuda de la cámara termográfica, es posible medir la frecuencia respiratoria de una víctima, tal como se describe en la sección 2.5.

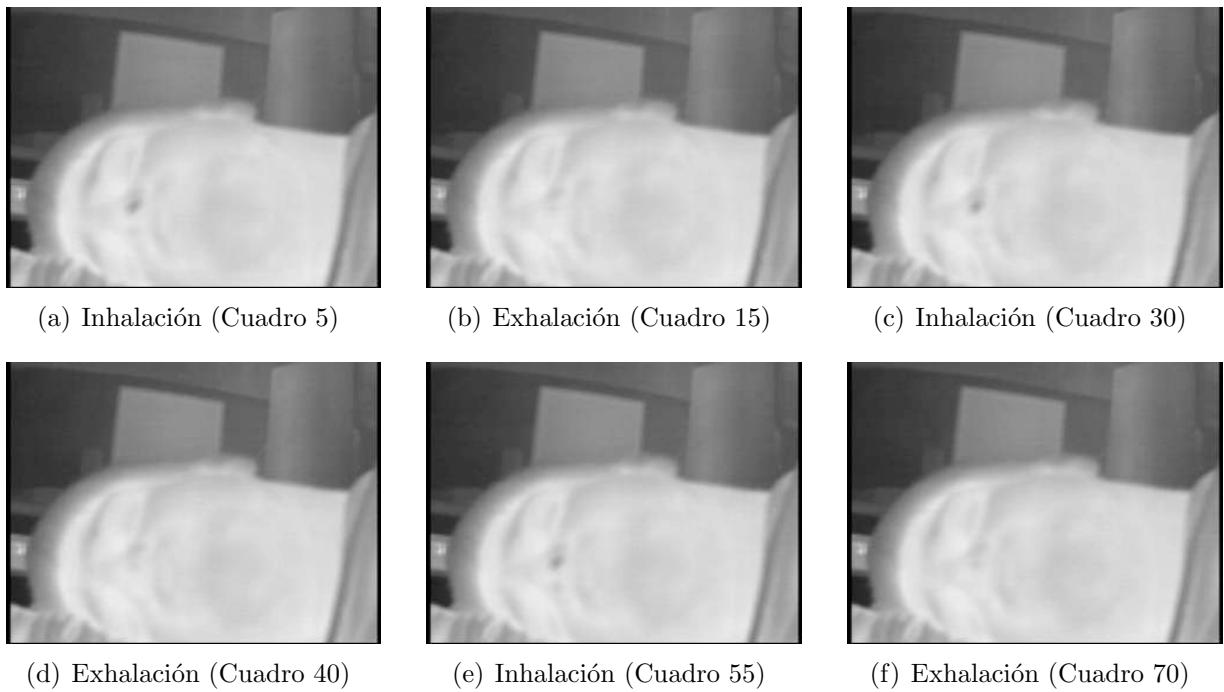


Figura 8: Imágenes asociadas a un ejemplo con 3 ciclos de respiración

En la Figura 8 se muestra un ejemplo de las imágenes obtenidas con la cámara termográfica para 3 ciclos de respiración. Nótese que sólo se muestran los cuadros 5 (subfigura 8(a)), 15 (subfigura 8(b)), 30 (subfigura 8(c)), 40 (subfigura 8(d)), 55 (subfigura 8(e)) y 70 (subfigura 8(f)) de la duración total del ejemplo. Este ejemplo será utilizado para ilustrar el

funcionamiento de todos los métodos propuestos en esta sección.

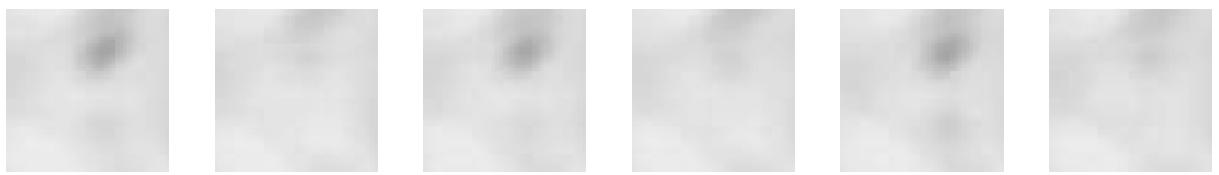
En las subfiguras 8(a), 8(c) y 8(e) los orificios nasales del individuo se observan de color gris oscuro, correspondientes a una baja temperatura, lo cual indica que el individuo está inhalando de acuerdo a lo explicado en la sección 2.5.

En las subfiguras 8(b), 8(d) y 8(f) los orificios nasales del individuo se muestran de un color muy similar al de su piel, lo cual indica que el individuo está exhalando, dado que el aire exhalado suele estar a la temperatura corporal del sujeto.

Las mediciones de respiración son hechas en el servidor. La información recopilada sobre dichas mediciones es enviada a los clientes por las conexiones TCP, para que sean mostradas en la interfaz.

Las mediciones comienzan después que el operador ha definido un rectángulo sobre el video de la interfaz. Este rectángulo corresponde al área de interés sobre la cual se desea hacer la medición. Las coordenadas de dicho rectángulo son enviadas al servidor, donde se comienza un proceso de captura de cuadros del video termográfico.

De cada cuadro capturado, se toma el área especificada por el operador, definiendo una nueva imagen que luego se lleva a una escala de 100×100 píxeles.



(a) Cuadro 5 (b) Cuadro 15 (c) Cuadro 30 (d) Cuadro 40 (e) Cuadro 55 (f) Cuadro 70

Figura 9: Imágenes recortadas de un ejemplo con 3 ciclos de respiración

Continuando con el ejemplo mostrado en la Figura 8, se exponen en la Figura 9 las imágenes recortadas, correspondientes los cuadros antes mencionados.

Para medir la frecuencia respiratoria de las víctimas, se han diseñado e implementado dos algoritmos distintos con múltiples variaciones. El primero de los métodos implementados intenta transformar la imagen de una escala de grises a una imagen binaria utilizando métodos de umbral. El segundo método intenta guiar la medición de la respiración a través de una función de costos aplicada al histograma normalizado de cada cuadro. Los métodos

propuestos serán explicados con mayor detalle en las subsecciones siguientes.

El objetivo de estos métodos es diferenciar las inhalaciones de las exhalaciones en el área de interés designada. Al diferenciarlas, se pueden contar los ciclos de respiración y dividirlos entre la cantidad de tiempo transcurrido, obteniéndose la frecuencia respiratoria de la víctima.

4.3.1. Método Binario

El método de análisis binario de imágenes consiste en la obtención de un umbral que permita definir una partición de los píxeles en la imagen de acuerdo a su valor.

Una vez obtenida esta imagen se procede a contar la cantidad de píxeles negros en la imagen y con este dato se puede determinar si el sujeto está inhalando o exhalando. Dado que la exhalación genera píxeles claros y la inhalación genera píxeles oscuros, es de esperarse que la medición de imágenes sucesivas en un ciclo respiratorio originen una onda. Dicha onda puede ser fácilmente separada en inhalaciones (valles de la onda) y exhalaciones (picos de la onda).

A cada imagen recortada obtenida se le hace un cálculo de histograma. Estos histogramas se utilizan, durante los primeros segundos de medición, para establecer un umbral que permita discretizar los píxeles de la imagen estableciendo su valor en 0 (negro), si su valor original es menor al del umbral, ó 255 (blanco) si su valor original es mayor al del umbral. Los métodos propuestos para el cálculo del umbral serán discutidos en la sección 4.3.2.

A esta fase de la medición, correspondiente al cálculo del umbral, se le llamará *Fase de Calibración*. Una vez transcurrida la Fase de Calibración, comienza la *Fase de Medición*, en la cual se procede a utilizar el umbral calculado para determinar las inhalaciones y exhalaciones de la víctima.

Una vez definido el umbral, se obtiene una imagen binaria. Esta imagen tiende a verse totalmente blanca cuando el sujeto está exhalando. Cuando el sujeto inhala, aparecen en la imagen manchas negras correspondientes a sus orificios nasales. La definición matemática de la conversión binaria de las imágenes viene dada por la ecuación 3, donde I corresponde a la

matriz que representa la imagen original y U el umbral de conversión binaria calculado.

$$NuevaImagen(i, j) = \begin{cases} 0 & \text{si } I(i, j) < U \\ 255 & \text{si } I(i, j) \geq U \end{cases} \quad (3)$$

Durante la Fase de Medición, una vez obtenidas las imágenes binarias, se procede entonces a calcular los valles y picos de la onda correspondiente.

El promedio de todos los valores será utilizado como punto de corte para determinar si la víctima está inhalando (la onda se encuentra bajo el valor encontrado), o exhalando (la onda se encuentra sobre el valor encontrado).

Los valores tomados para la diagramación de la onda vienen dados por la ecuación 4, donde I corresponde a la matriz de tamaño $N \times M$ que representa una imagen procesada a modo binario.

$$Valor = \sum_{i=0}^N \sum_{j=0}^M Negros(I(i, j)) \quad (4)$$

Donde,

$$Negros(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{en otros casos} \end{cases} \quad (5)$$

4.3.2. Cálculo automático de umbral

Para la determinación del umbral de las imágenes se proponen dos métodos distintos. El primero de estos métodos, llamado Método de Mínimos Locales, fue definido en esta investigación, de manera experimental, por medio del estudio y análisis de varias series de histogramas generados por distintas víctimas, tomados con la cámara termográfica. El segundo método que se utiliza es el cálculo de umbrales definido por N. Otsu [15]. Cada uno de estos métodos se detalla a continuación.

Método de Mínimos Locales (Experimental)

Al graficar los histogramas de varias secuencias de imágenes, se pudo apreciar que éstos tienen la tendencia a expandirse hacia los valores de negro cuando la víctima inhala, y de

contraerse hacia los valores de blanco cuando la víctima exhala.

El método definido consiste entonces en encontrar el mínimo valor registrado en cada histograma, que correspondería a la mínima temperatura registrada.

De manera dinámica, se procede a elegir el máximo valor entre los valores mínimos locales registrados, que correspondería al umbral buscado. La definición matemática de este umbral viene dada por la ecuación 6, donde H corresponde al conjunto de los histogramas de los cuadros del muestreo.

$$Umbral(H) = \{ \max(E) : h \in H \wedge E = \min(h) : E \} \quad (6)$$

En otras palabras, durante la Fase de Calibración se procede a encontrar el máximo de las mínimas temperaturas registradas por la cámara térmica, que correspondería al color de la piel del sujeto cuando está exhalando.

Siguiendo con el ejemplo ilustrado en la sección 4.3, en la Figura 10 se muestran los histogramas correspondientes a los cuadros recortados que se muestran en la Figura 9.

De acuerdo con el ejemplo propuesto, en la Figura 11 se muestran las imágenes binarias correspondientes a los cuadros de la Figura 9, calculadas con el umbral definido por el método de Mínimos Locales. Nótese que podría haber ruido (algunas manchas) en los cuadros de exhalación.

En la Figura 12 se muestra la onda calculada durante la Fase de Medición. Esta onda se obtiene mediante la aplicación de la ecuación 4 a las imágenes binarias de la Figura 11. Para la medición de respiración se toman en cuenta los saltos a través de la línea correspondiente a los puntos de corte.

Método Otsu

El algoritmo Otsu [15] es un método que busca automatizar la binarización de imágenes, basándose en la forma de sus histogramas.

El algoritmo asume que la imagen contiene dos clases de píxeles: fondo y contenido importante. Por lo tanto, se esfuerza en determinar el mejor umbral posible para que la

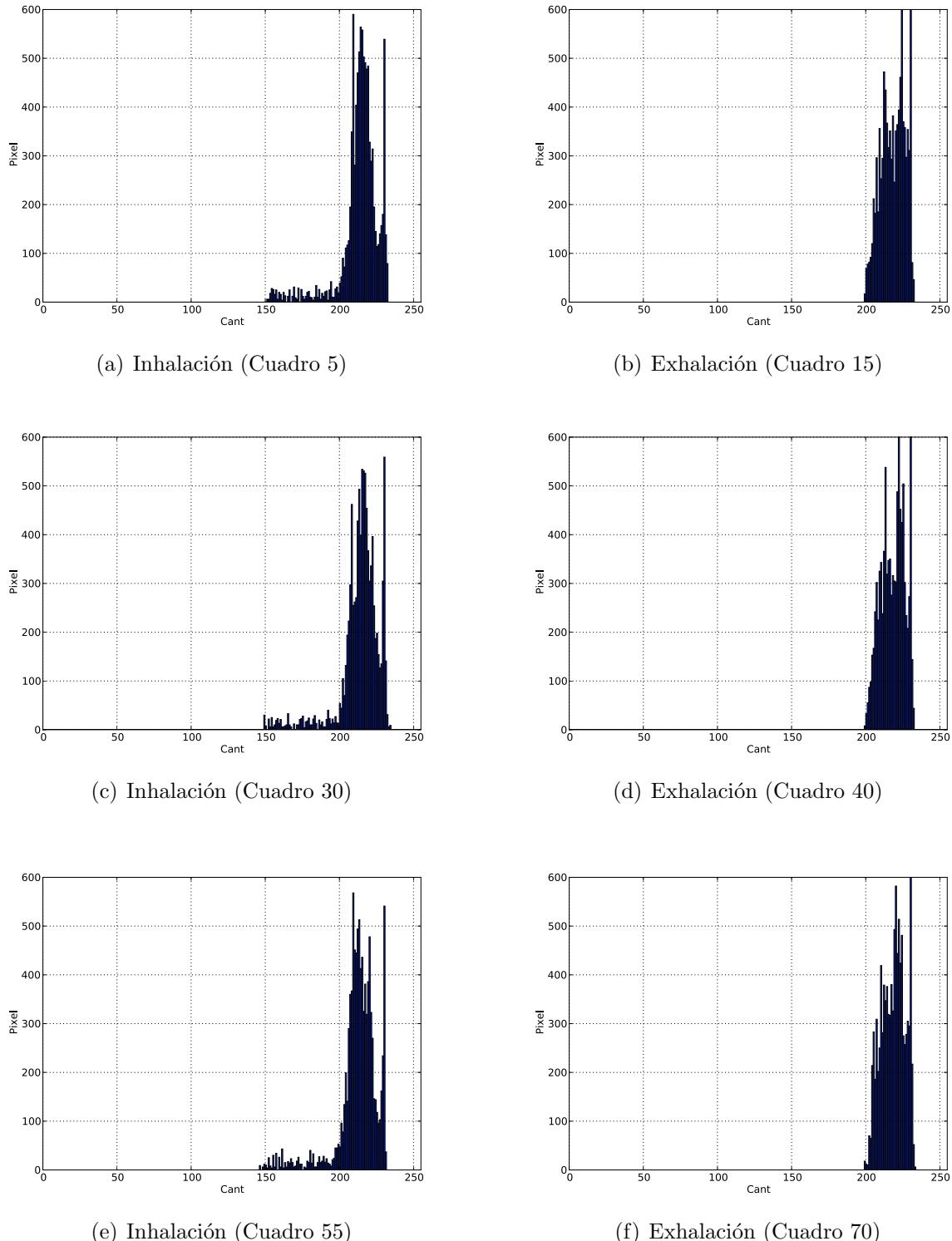


Figura 10: Histogramas correspondientes a las imágenes recortadas de un ejemplo con 3 ciclos de respiración

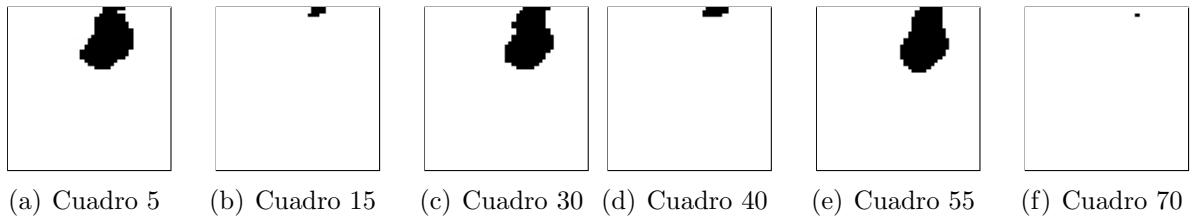


Figura 11: Imágenes binarias de un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método de Mínimos Locales.

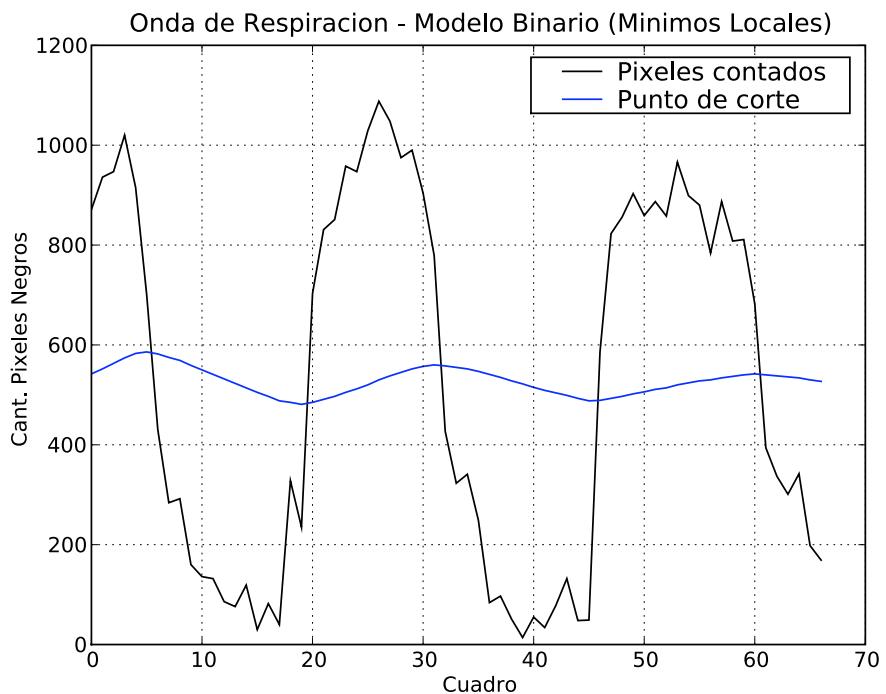


Figura 12: Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método de Mínimos Locales.

varianza interna dentro de cada clase de pixel sea mínima.

El algoritmo se basa en la búsqueda exhaustiva del umbral que minimiza la varianza entre clases, definida como la suma ponderada de la varianza $\sigma_w^2(t)$ de las dos clases, como se expresa en la ecuación 7.

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \quad (7)$$

En la ecuación 7, w_i corresponde a la probabilidad de cada una de las clases generadas

por el umbral t , mientras que σ_i^2 corresponde a la varianza de cada una de estas clases.

Otsu demostró [15] que minimizar la varianza interna de las clases era equivalente a maximizar la varianza existente entre ellas, lo cual conduce a la ecuación 8.

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_1(t) \times w_2(t) \times (\mu_1(t) - \mu_2(t)) \quad (8)$$

A partir de la ecuación 8 se puede generar un algoritmo sencillo que itera a través de los valores de t , recalculando w_i y μ_i , para obtener el máximo $\sigma_b^2(t)$, que corresponde al umbral buscado. El pseudo-código del algoritmo Otsu para el cálculo del umbral se muestra en la Figura 13.

Algoritmo 4.3.1: OTSUThresholding(*Imagen*)

```

Array[256] h
int t = 0
float w1, w2, mu1, mu2, sigma_b^2
sigma_b^2 = -infinity
h = CALCULARHISTOGRAMA(Imagen)
int i = 0
for i ≤ 255
    do {
        CALCULARPARAMETROS(w1, w2, mu1, mu2)
        float sigma_i^2 = w1 × w2 × (mu1 - mu2)
        if sigma_i^2 ≥ sigma_b^2
            then {
                t = i
                sigma_b^2 = sigma_i^2
            }
        i ++
    }
return (t)

```

Figura 13: Algoritmo *Otsu* para el cálculo del Umbral

Durante la Fase de Calibración se procede a elegir el mínimo umbral obtenido por medio del método Otsu. De este modo, se estaría eligiendo el umbral que resalta mejor los orificios de la nariz durante la inhalación.

El umbral elegido mediante el método Otsu, al igual que el umbral elegido mediante el método de Mínimos Locales, será utilizado durante la Fase de Medición para calcular la onda de respiración. Esta onda se obtiene mediante la aplicación de la ecuación 4 a las imágenes binarias de la Figura 14. Los saltos a través de la línea correspondiente a los puntos de corte se leen como inhalaciones (por encima de la línea) o exhalaciones (por debajo de la línea). La onda obtenida al binarizar los cuadros del ejemplo con el método Otsu se muestra en la Figura 15.

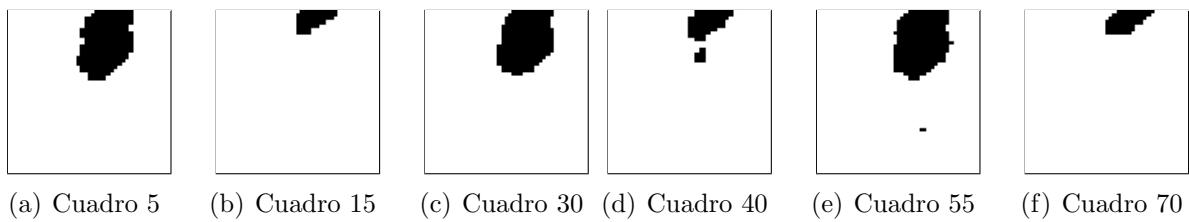


Figura 14: Imágenes binarias de un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método Otsu.

La principal diferencia entre el método Otsu para la determinación del umbral y el método de Mínimos Locales propuesto radica en que el método Otsu hace un cálculo probabilístico para calcular el umbral, mientras que el método de Mínimos Locales intenta determinar la temperatura de la piel de la víctima, para diferenciar las inhalaciones de las exhalaciones.

Ambos umbrales se determinan durante la Fase de Calibración. El método utilizado para medir la frecuencia respiratoria durante la Fase de Medición es el mismo, sin importar cómo fue elegido el umbral utilizado.

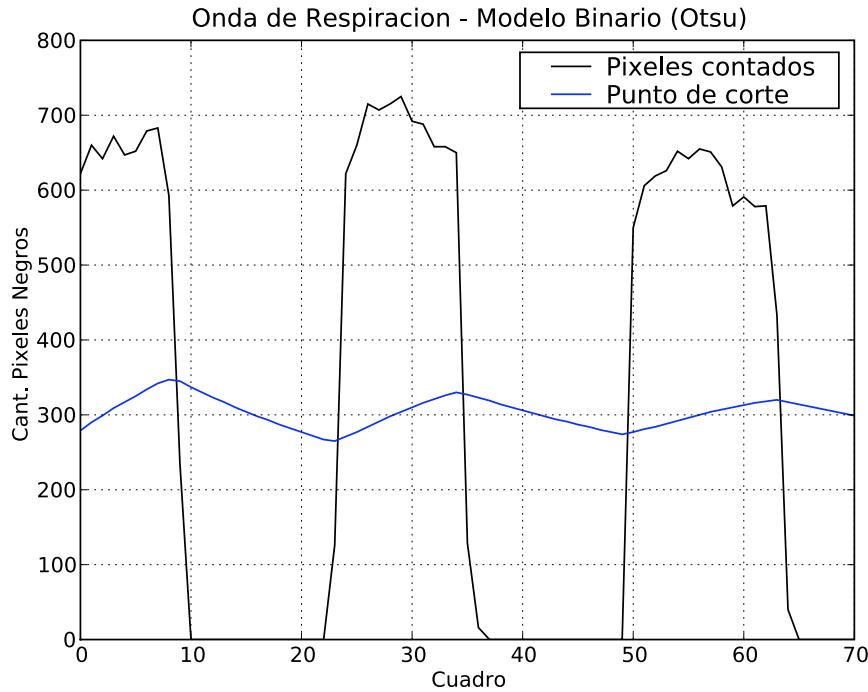


Figura 15: Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Umbral calculado mediante el método Otsu.

4.3.3. Método de Costos

El método de análisis de imágenes por medio de una función de costos consiste en asignar un valor a cada posible tono de gris, de modo que el costo calculado de aquellas imágenes que tengan más píxeles de baja intensidad (oscuros) sea mayor que el costo de aquellas imágenes que tengan más píxeles de alta intensidad (claros).

En otras palabras, los cuadros de video correspondientes a inhalaciones tendrían un costo más elevado que los cuadros correspondientes a exhalaciones, dado que durante las inhalaciones se registran temperaturas más bajas, representadas por píxeles de menor intensidad (más oscuros).

Al igual que el Método Binario, descrito en la subsección 4.3.1, el Método de Costos se aplica sobre el área de interés definida por el teleoperador. El Método de Costos procede entonces a asignar un valor al área evaluada, de acuerdo con la ecuación 9.

$$\text{costo}(I) = \sum_{i \in H} \left[\left(\frac{256}{i+1} \right) \times H(i) \right] \quad (9)$$

Donde I corresponde a la imagen cuyo costo se desea calcular y H corresponde al histograma de la imagen I , siendo $H(i)$ la cantidad de píxeles que toman el valor i .

El costo de cada histograma de los cuadros de la Figura 10 se muestra en el Cuadro 3. Nótese la diferencia de costo entre los cuadros 5, 30 y 55 (inhalación) y los cuadros 15, 40 y 70 (exhalación).

Cuadro	5	15	30	40	55	70
Costo	11999.20	11690.03	11987.54	11696.45	12086.41	11695.78

Cuadro 3: Costo de los histogramas de la Figura 10.

Al igual que el Método Binario, la aplicación del Método de Costos a una serie de imágenes sucesivas resulta en una onda, que puede separarse con facilidad en inhalaciones (picos de la onda) y exhalaciones (valles de la onda). La onda obtenida mediante la aplicación del Método de Costos sobre los histogramas del ejemplo se muestra en la Figura 16.

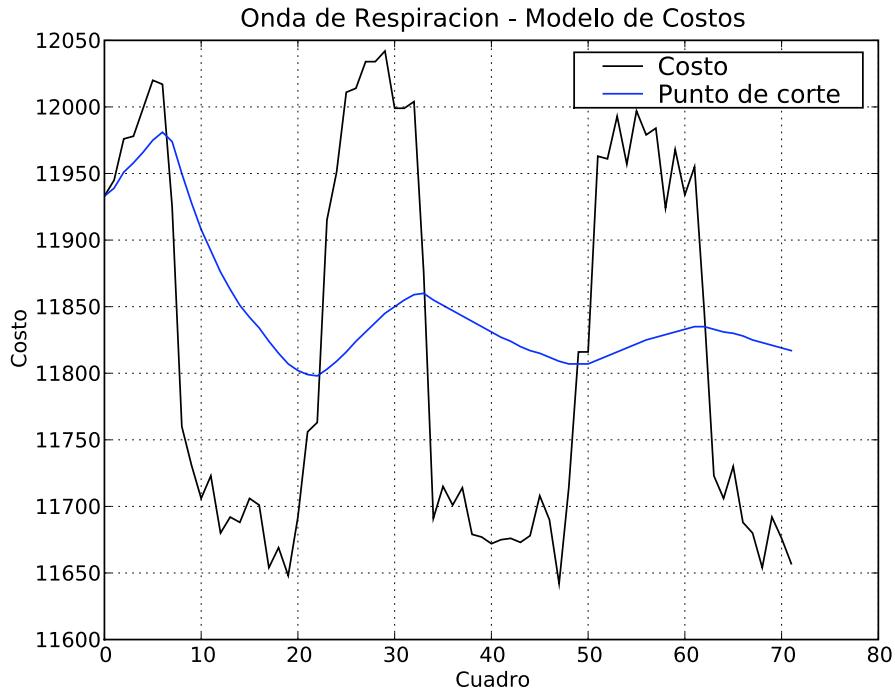


Figura 16: Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Costo de los histogramas calculado mediante el Método de Costos.

El Método de Costos favorece los píxeles de baja intensidad sobre los de mayor intensidad. Por lo tanto, se propone normalizar las imágenes a ser evaluadas, para que el método tenga un mejor desempeño.

De acuerdo con Nixon y Aguado [14], una imagen normalizada es aquella cuyo histograma se distribuye uniformemente, de modo que se estira el rango de intensidades de la imagen. Es decir, un histograma normalizado cubre los 256 niveles de intensidad posibles, obteniéndose una imagen con mejor contraste que la original.

Según los autores, si el histograma de la imagen original I comienza en el valor I_{min} y se extiende hasta el valor de intensidad I_{max} , podemos escalar las intensidades de la imagen, de modo que los píxeles en la nueva imagen N estén entre N_{min} y N_{max} , como se muestra en la ecuación 10.

$$N = \frac{N_{max} - N_{min}}{I_{max} - I_{min}} \times (I - I_{min}) + N_{min} \quad (10)$$

Al aplicar el Método de Costos sobre un histograma normalizado, se garantiza que haya una diferencia importante entre el costo de una imagen de inhalación y otra de exhalación, lo cual disminuye la probabilidad de cometer errores de evaluación.

Continuando con el ejemplo, en la Figura 17 se muestran los histogramas obtenidos al normalizar los histogramas presentados en la Figura 10. Nótese que los histogramas correspondientes a inhalaciones se extienden hacia los valores de menor intensidad (más cercanos a cero), lo cual aumenta la diferencia de costos entre los cuadros correspondientes a inhalaciones y los cuadros correspondientes a exhalaciones. El costo de cada histograma de la Figura 17 se muestra en el Cuadro 4. Nótese que la diferencia de costo entre los cuadros 5, 30 y 55 (inhalación) y los cuadros 15, 40 y 70 (exhalación) es mayor que la diferencia entre estos mismos cuadros en el Cuadro 3.

Cuadro	5	15	30	40	55	70
Costo	21407.58	17811.17	22230.09	17833.99	24999.83	17828.55

Cuadro 4: Costo de los histogramas normalizados de la Figura 17.

Como se menciona anteriormente en la subsección 4.3.1, el proceso de cálculo de la frecuencia respiratoria de una víctima necesita de una Fase de Calibración antes de proceder hacia la Fase de Medición.

En la Fase de Calibración de la aplicación diseñada se procede a calcular el mínimo N_{min} y el máximo N_{max} de los cuadros capturados, para normalizar los histogramas de los cuadros capturados durante la Fase de Medición. Al normalizar todos los histogramas con el mismo rango, se acentúa la diferencia de costo de las inhalaciones sobre las exhalaciones.

De esta manera, durante la Fase de Medición, se aplica el Método de Costos a los histogramas normalizados para obtener la onda de respiración discutida anteriormente. El promedio de todos los valores de la onda calculada se utiliza como punto de corte para determinar las

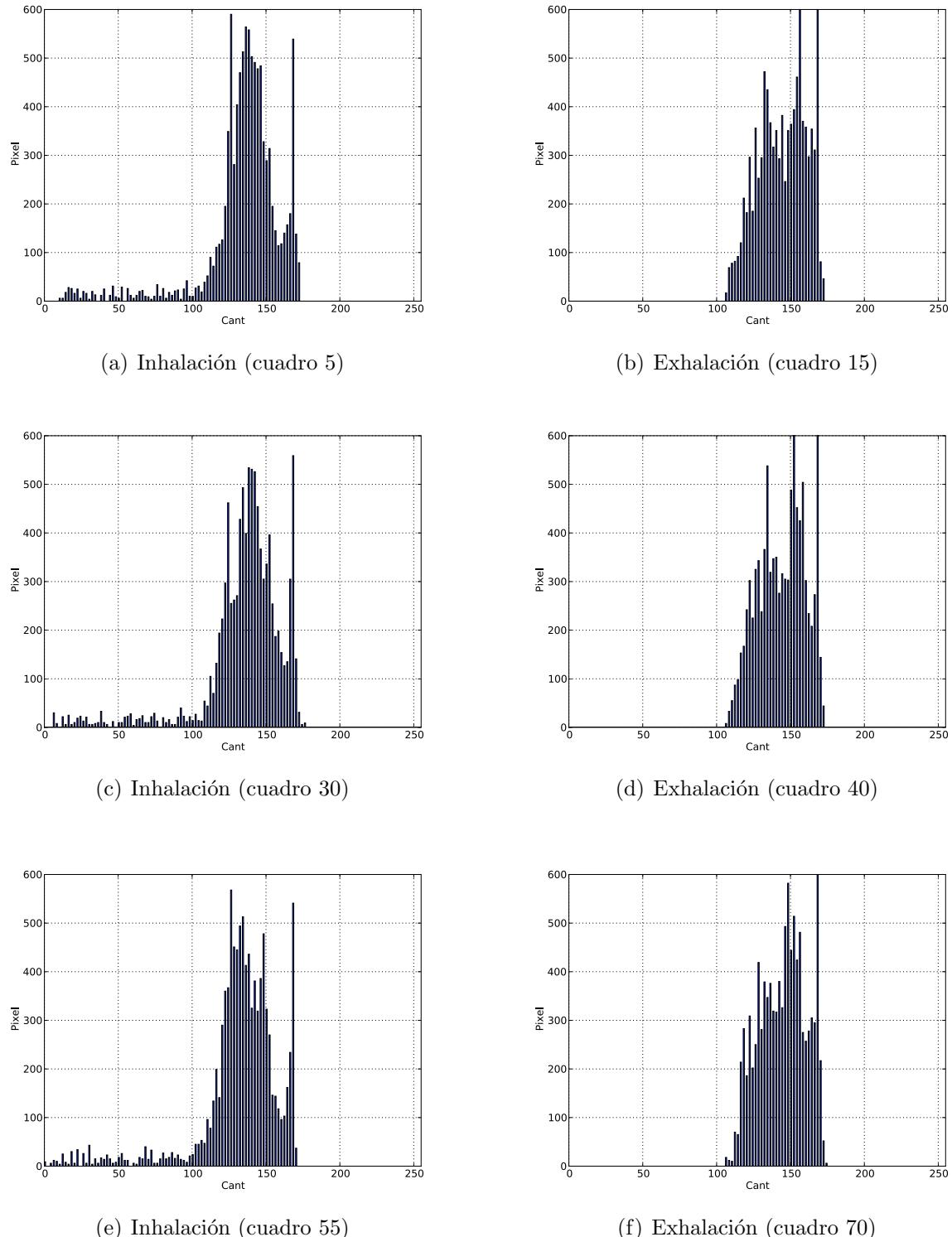


Figura 17: Histogramas normalizados correspondientes a las imágenes recortadas de un ejemplo con 3 ciclos de respiración

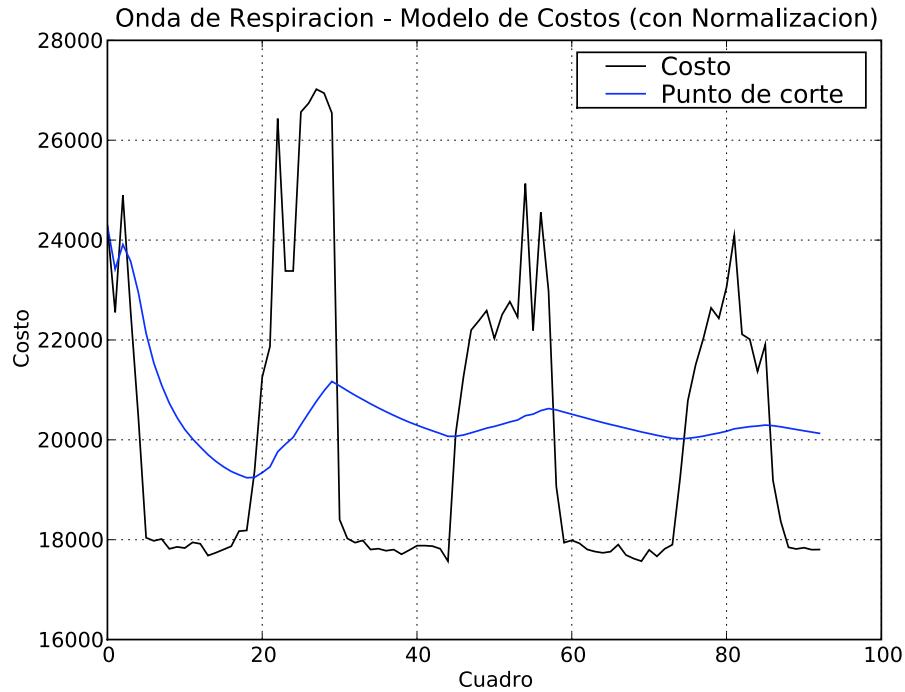


Figura 18: Onda de Respiración correspondiente a un ejemplo con 3 ciclos de respiración. Costo de los histogramas normalizados calculado mediante el Método de Costos.

inhalaciones (la onda se encuentra sobre el promedio) y las exhalaciones (la onda se encuentra bajo el promedio) de las víctimas encontradas. La onda obtenida mediante el Método de Costos aplicado sobre histogramas normalizados del ejemplo se muestra en la Figura 18.

En la sección 5.2 se hace una comparación de los métodos diseñados, haciendo mediciones de frecuencia respiratoria sobre cierta cantidad de sujetos y calculando el error porcentual cometido por cada uno de los métodos.

4.3.4. Otros Métodos

Como complemento de la implementación de los métodos de medición, se sugirió utilizar algunos algoritmos tradicionales de procesamiento de imágenes. Se intentó detectar los bordes de las imágenes utilizando el algoritmo de Canny [4] y filtros de Laplace [20]. En el Apéndice C se explican las razones por las cuales estos métodos fueron desechados y se presentan algunos ejemplos.

Capítulo 5

Experimentos y Resultados

El objetivo de la aplicación consiste en asistir a los rescatistas en la Búsqueda y Rescate de víctimas en una situación de riesgo. En este tipo de situaciones cada detalle podría ser muy valioso para encontrar una víctima con vida. Por esta razón, es importante que el video enviado por el robot sea siempre estable y presente pocas fallas.

De acuerdo con Ollero Baturone [3], una aplicación para teleoperar un robot debe ser estable y robusta, de modo que la realimentación de video y de los sensores, así como los tiempos de respuesta de la aplicación, se mantengan en tiempo real o con un desplazamiento de tiempo relativamente corto. De este modo, también se espera que la calidad del video sea lo suficientemente adecuada como para poder percibir el entorno del robot. Adicionalmente, se espera que la interfaz de la aplicación permita que cualquier usuario pueda teleoperar el robot de manera sencilla sin necesidad de un largo entrenamiento previo.

Por otro lado, las mediciones de la frecuencia respiratoria de una víctima deben ser lo suficientemente aproximadas a la frecuencia respiratoria real de la víctima. De lo contrario, las decisiones a tomarse sobre el estado vital de la víctima podrían ser erradas, lo cual podría costar su vida.

Ante la necesidad de demostrar la confiabilidad, eficiencia, robustez y estabilidad de la aplicación, se llevaron a cabo varios experimentos:

- Experimento I *Evaluación de la teleoperación del robot*: En este experimento se mide la velocidad promedio y la estabilidad de una transmisión de audio y video del sistema, durante 24 horas. Adicionalmente se hacen pruebas de movilidad y funcionalidad del robot para evaluar la interfaz de la aplicación y el tiempo de respuesta del robot. Una descripción detallada de este experimento puede encontrarse en la sección 5.1 de este capítulo.
- Experimento II *Precisión de los métodos propuestos para la medición de frecuencia*

respiratoria: En este experimento se calcula la frecuencia respiratoria de 30 individuos, con los métodos de medición propuestos en la sección 4.3. Este experimento se describe en la sección 5.2 de este capítulo.

Para cada uno de los experimentos se presenta su objetivo, hipótesis, metodología y un análisis de los resultados obtenidos. Las conclusiones derivadas de estos resultados se presentan en el capítulo 6.

5.1. Experimento I *Evaluación de la teleoperación del robot*

Una transmisión multimedia estable garantiza la fluidez del video y del audio recibidos. Sin embargo, dicha transmisión se ve afectada por la red sobre la cual se establece la comunicación.

Como se mencionó anteriormente, una aplicación para teleoperar un robot a través de Internet debe contar con tiempos de respuesta lo suficientemente pequeños como para que sean considerados tiempo real.

El objetivo de este experimento es determinar cuán estable es una transmisión de multimedia desde el servidor hasta el cliente y evaluar la interfaz de la aplicación y los tiempos de respuesta de la misma.

5.1.1. Hipótesis

Dado que:

1. Se está utilizando la librería *JMF*, lo cual debería garantizar estabilidad y constancia en el envío de paquetes RTP desde el cliente hasta el servidor, como se explica en la sección 4.1.
2. De acuerdo con Fiorini y Oboe [8], la transmisión de RTP se ve afectada por los tiempos de viaje de los paquetes en la red.

3. El *bit rate* corresponde al ancho de banda, o la cantidad de bits consumidos por segundo. La calidad del video es directamente proporcional al *bit rate*.
4. El Servidor y el Cliente Operador intercambian caracteres con información a través de una conexión TCP.
5. Las conexiones TCP suelen verse afectadas por el tráfico en la red.

Se espera entonces que exista una pérdida de los paquetes RTP en la red, así como distintos tiempos de llegada para los paquetes entrantes al cliente, lo cual afectaría el *bit rate* del envío multimedia. Sin embargo, se espera que el *bit rate* no sea afectado de manera significativa durante la medición.

Se espera que los paquetes TCP no se pierdan, aunque tengan tiempos de llegada poco predecibles. Dado que los paquetes enviados contienen sólo caracteres es posible que el tiempo de respuesta del robot no se vea afectado significativamente por el tráfico en la red.

5.1.2. Metodología utilizada

Para este experimento se procedió a establecer una conexión entre el cliente (ubicado en la urbanización La Tahona en Caracas, Venezuela) y el servidor (ubicado en la urbanización Macaracuay en Caracas, Venezuela). El cliente y el servidor están separados por 5 kilómetros de distancia lineal y 6 saltos o *gateways* en la red. Es decir, al enviar un paquete de información desde el servidor hasta el cliente, éste debe recorrer 6 *routers* o enrutadores para llegar a su destino.

La librería *JMF* permitió hacer una medición del *bit rate* en la aplicación cliente. Se hicieron mediciones cada segundo y luego se generó una gráfica con la información obtenida.

Las mediciones comenzaron el día 9 de junio de 2008 a las 21:56 y se detuvieron a las 21:56 del día siguiente. Cinco teleoperadores distintos procedieron a operar el robot durante el período de medición, para probar el tiempo de respuesta de la aplicación y su interfaz.

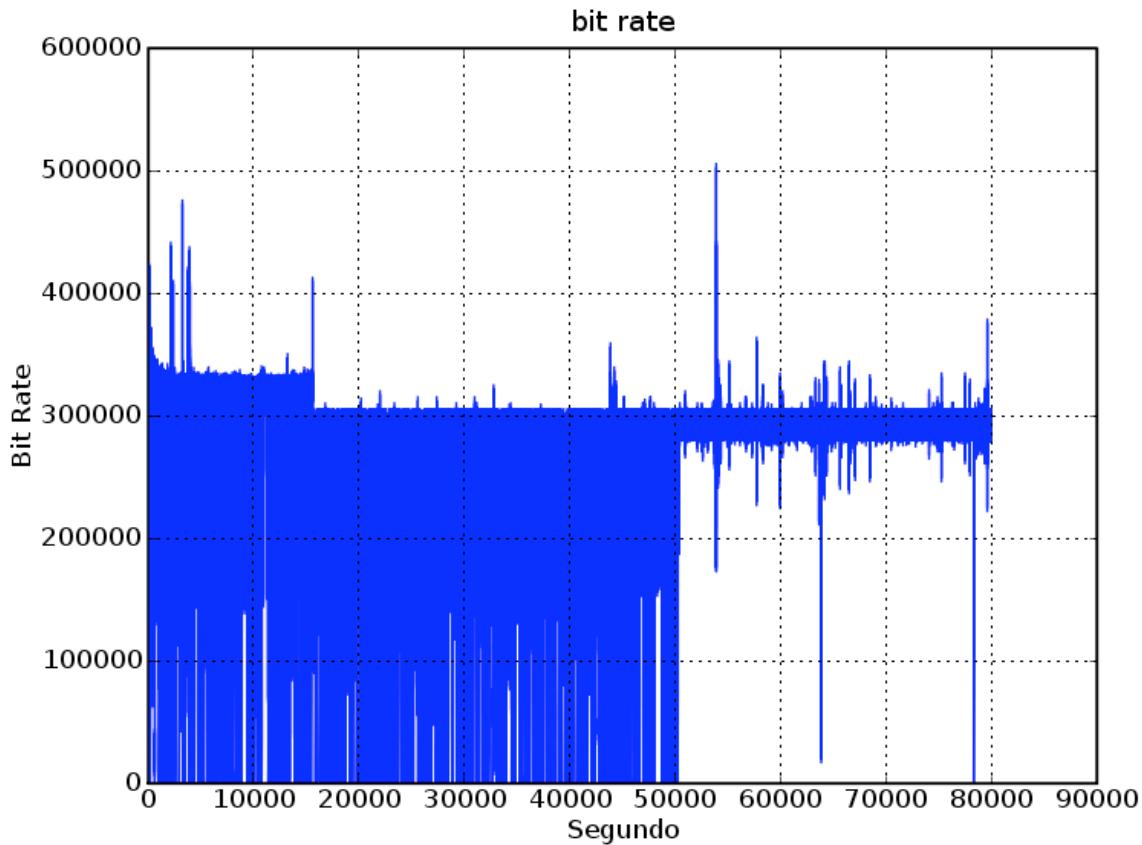


Figura 19: Comportamiento del *Bit rate* de la red

5.1.3. Análisis de resultados obtenidos

El *bit rate* medido tiene un valor promedio de 262562 bits por segundo. En la Figura 19 se muestra la gráfica correspondiente al valor del *bit rate* durante la medición hecha.

La gráfica deja en evidencia el comportamiento errático de la red. Si bien el valor del *bit rate* suele oscilar entre 283000 y 303000 bits por segundo, la red suele tener muchos valles, en los cuales el envío de información cae a 0 bits por segundo. Si los valles de 0 bits llegarán a ser muy amplios el rendimiento de la aplicación se vería severamente afectada por retrasos en la reproducción o calidad reducida de los datos.

Los teleoperadores lograron utilizar la aplicación sin previo entrenamiento y coincidieron en que la interfaz de la aplicación es cómoda y fácil de entender y utilizar. El tiempo promedio de envío de los paquetes en la red fue de 10 milisegundos, lo cual implica tiempos de respuesta

inmediatos por parte del robot y de las mediciones de respiración.

5.2. Experimento II *Precisión de los métodos propuestos para la medición de frecuencia respiratoria*

En la sección 4.3 se introdujeron los distintos métodos propuestos para la medición de la frecuencia respiratoria de una víctima. Estos métodos serían el Método del Costo, el Método del Costo con Histogramas Normalizados, el Método de Binarización con Otsu y el Método de Binarización con Mínimos Locales.

La herramienta para la estimación de la frecuencia respiratoria cumple una labor muy importante a la hora de tomar decisiones sobre la prioridad de rescate de las víctimas encontradas. Es por ello que se necesita que las mediciones hechas por dicha herramienta sean muy precisas o, en su defecto, muy cercanas a la frecuencia respiratoria real.

Entonces es necesario encontrar el método propuesto que mejor se adapte a un ambiente en el cual no existan condiciones controladas, con víctimas que podrían moverse durante la medición y cuya posición no facilite la medición. Adicionalmente, el desempeño de los métodos no debe verse afectado por las distancias a las cuales se encuentran las víctimas.

El mejor método sería aquél que logre aproximar la frecuencia respiratoria real de la víctima, sin importar la temperatura de su nariz y cara, con la posibilidad de existir bigotes y otros cabellos, cuya temperatura podría confundirse con la de sus orificios nasales al inhalar.

Adicionalmente, el método elegido debería ser robusto ante cualquier introducción de una nueva temperatura al ambiente, que modifique el rango de medición de la cámara termográfica.

El objetivo de este experimento es determinar cuál de los métodos propuestos estima con mayor precisión la frecuencia respiratoria de los sujetos a los cuales se les hizo la medición.

5.2.1. Hipótesis

Dado que:

1. Las narices y otras superficies frías se muestran de un tono de gris oscuro, que a simple

vista podrían confundirse con los orificios nasales de la víctima cuando inhala.

2. El cabello, bigotes, cejas y barba no tienden a calentarse. Por lo tanto, se muestran de un tono de gris oscuro.
3. De acuerdo con Murthy y Pavlidis [13], cualquier movimiento por parte de la víctima tiende a afectar el desempeño de los algoritmos de medición de frecuencia respiratoria.
4. De acuerdo con Fei y Pavlidis [7], es necesario calibrar la cámara termográfica para que detecte temperaturas en un rango de 28°C a 38°C y controlar la temperatura de la habitación, o la medición podría verse afectada.
5. Los experimentos de Murthy y Pavlidis [13] y Fei y Pavlidis [7] implican sujetos colocados en posiciones específicas, a una distancia precisa de la cámara termográfica.
6. Los métodos binarios descritos en la sección 4.3 son versiones más simples de los métodos empleados por Murthy y Pavlidis [13] y Fei y Pavlidis [7].
7. En una situación de riesgo no es posible controlar la posición de la víctima, ni acondicionar el ambiente para los experimentos. Tampoco se puede controlar la distancia entre la víctima y la cámara termográfica.
8. La cámara termográfica disponible para esta investigación no puede ser calibrada.

Entonces, se espera que los métodos presenten algunos errores de medición. También se espera que los métodos de binarización tiendan a fallar con las narices frías, dado que el umbral elegido podría tomar porciones de la nariz como parte de la inhalación, lo cual resultaría en mediciones erróneas de respiraciones.

Adicionalmente, se espera que el método de costos con la normalización de histogramas retorne mejores resultados que aquél sin normalización, dado que de este modo se aumentan los contrastes dentro de la imagen. De este modo se demostraría la importancia de la normalización de histogramas para este método.

Se espera que el método de evaluación mediante costos logre medir la frecuencia respiratoria de los sujetos con mayor precisión que los métodos binarios ya que este no depende de la separación de la información en la imagen. Se espera poder demostrar que los métodos basados en modelos de costo consideran la contribución de cada pixel de acuerdo a su intensidad en todo momento y tienen una tolerancia mayor al ruido en las imágenes obtenidas por la cámara termográfica.

5.2.2. Metodología utilizada

Para este experimento se midió la frecuencia respiratoria de 30 voluntarios, con distintas características y anatomías faciales, acomodados en diferentes posiciones y a distintas distancias de la cámara. En el Apéndice D se puede encontrar una tabla con los sujetos analizados.

Las mediciones se hicieron sobre videos de una duración aproximada de un minuto cada uno. Estos videos fueron capturados con la cámara termográfica, conectada a una tarjeta de video 713x BDA Analog Capture o una tarjeta Hauppauge WinTV Capture. Cabe destacar que ambas tarjetas recuperan la entrada video de la misma manera, por lo cual las imágenes obtenidas son idénticas.

Sobre cada video se designó el área de interés para la detección de respiración y se procedió a medir la frecuencia respiratoria con cada uno de los métodos propuestos, conservando el área de interés designada. También se contó (visualmente) la cantidad de respiraciones real de cada sujeto voluntario, para poder compararla con la medición obtenida.

Para cada medición se calculó el error porcentual de aproximación con respecto a la medición real. La fórmula utilizada para el cálculo de dicho error se presenta en la ecuación 11.

$$\delta = \frac{|medicion - real|}{|real|} \times 100 \% \quad (11)$$

Posteriormente, para cada método propuesto, se calculó la magnitud del error porcentual total de aproximación con respecto a la medición real. Dicho error fue calculado mediante las fórmulas de las ecuaciones 12 y 13 (fórmula del módulo euclíadiano).

$$\Delta = \frac{\|medicion - real\|}{\|real\|} \times 100 \% \quad (12)$$

Donde,

$$\|a\| = \sqrt{\sum_{i=0}^n a_i^2} \quad (13)$$

5.2.3. Análisis de resultados obtenidos

En el Cuadro 5 se presentan los resultados de las mediciones realizadas a 26 sujetos con cada uno de los métodos propuestos. Para cada sujeto se presenta el valor real y el valor medido (VM) por el algoritmo de medición utilizado, junto a su error porcentual (EP). En la última columna del cuadro se presentan notas relacionadas al sujeto estudiado.

De los 30 sujetos estudiados sólo se reportan 26 en este cuadro. Los otros sujetos son presentados posteriormente como casos atípicos, que no pueden ser manejados correctamente por ninguno de los algoritmos implementados.

Sujeto	Real	Costo		Costo + Normal.		Bin + ML		Bin + Otsu		Notas
		VM	EP	VM	EP	VM	EP	VM	EP	
S_1	8	9	12.50 %	8	0.00 %	8	0.00 %	8	0.00 %	
S_2	15	20	33.33 %	17	13.33 %	23	53.33 %	25	66.67 %	Ángulo incómodo
S_3	20	23	15.00 %	20	0.00 %	20	0.00 %	20	0.00 %	
S_4	29	29	0.00 %	28	3.45 %	29	0.00 %	28	3.45 %	Nariz fría
S_5	21	21	0.00 %	21	0.00 %	21	0.00 %	21	0.00 %	
S_6	13	15	15.38 %	13	0.00 %	16	23.08 %	24	84.62 %	Ángulo incómodo
S_7	15	16	6.67 %	15	0.00 %	14	6.67 %	16	6.67 %	
S_8	13	13	0.00 %	13	0.00 %	25	92.31 %	13	0.00 %	Nariz fría
S_9	23	23	0.00 %	21	8.70 %	28	21.74 %	23	0.00 %	Nariz y mejillas frías
S_{10}	23	23	0.00 %	23	0.00 %	23	0.00 %	23	0.00 %	
S_{11}	20	19	5.00 %	20	0.00 %	23	15.00 %	20	0.00 %	Nariz fría
S_{12}	16	16	0.00 %	16	0.00 %	14	12.50 %	16	0.00 %	
S_{13}	13	15	15.38 %	15	15.38 %	11	15.38 %	13	0.00 %	Nariz fría
S_{14}	17	14	17.65 %	15	11.76 %	18	5.88 %	14	17.65 %	
S_{15}	20	16	20.00 %	20	0.00 %	22	10.00 %	30	50.00 %	
S_{16}	15	15	0.00 %	16	6.67 %	14	6.67 %	14	6.67 %	
S_{17}	19	25	31.58 %	20	5.26 %	10	47.37 %	19	0.00 %	Difícil a simple vista
S_{18}	17	18	5.88 %	17	0.00 %	17	0.00 %	17	0.00 %	Bigotes. El sujeto se mueve varias veces durante la medición
S_{19}	14	22	57.14 %	14	0.00 %	24	71.43 %	17	21.43 %	
S_{20}	28	18	35.71 %	28	0.00 %	22	21.43 %	30	7.14 %	
S_{21}	20	20	0.00 %	20	0.00 %	20	0.00 %	20	0.00 %	Nariz fría
S_{22}	17	14	17.65 %	17	0.00 %	17	0.00 %	16	5.88 %	El sujeto se mueve varias veces durante la medición
S_{23}	35	39	11.43 %	35	0.00 %	35	0.00 %	38	8.57 %	Difícil a simple vista
S_{24}	19	18	5.26 %	19	0.00 %	18	5.26 %	18	5.26 %	
S_{25}	11	12	9.09 %	11	0.00 %	12	9.09 %	8	27.27 %	
S_{26}	17	24	41.18 %	19	11.76 %	21	23.53 %	18	5.88 %	Bigotes. Mejillas frías

Cuadro 5: Resultados de mediciones de los métodos propuestos.

(ML = Mínimos Locales VM = Valor Medido EP = Error Porcentual)

En el Cuadro 6 se muestra el error porcentual total de aproximación con respecto a la medición real de cada método propuesto.

Costo	Costo + Normal.	Bin + ML	Bin + Otsu
18.97 %	4.88 %	22.79 %	19.50 %

Cuadro 6: Error porcentual total de aproximación con respecto a la medición real de cada método propuesto.
(ML = Mínimos Locales)

El método que comete menos errores es el del Costo con histogramas normalizados. Este método es más preciso que el método del Costo sin histogramas normalizados, dado que un histograma normalizado tiene píxeles con valor 0, creando un contraste significativo entre el costo de las inhalaciones y el de las exhalaciones. En un histograma no normalizado el costo de las inhalaciones podría ser muy similar al de las exhalaciones.

El error cometido por el método de binarización con umbralización Otsu es bastante similar al error cometido por el método de binarización con umbralización por Mínimos Locales. Ello ocurre porque los métodos de binarización funcionan de la misma manera a la hora de medir la frecuencia respiratoria, aunque el umbral elegido sea distinto. Sin embargo, pareciera que el umbral calculado con el método Otsu divide mejor la información relevante que el umbral calculado con el método de Mínimos Locales.

Los datos reportados en el Cuadro 5 dejan en evidencia que el método del Costo con histogramas normalizados comete pocos errores en el momento de medir la frecuencia respiratoria de víctimas con nariz fría, víctimas en movimiento y en posiciones incómodas para hacer la medición.

La medición de la frecuencia respiratoria de los sujetos S_{17} y S_{23} resultaba difícil a simple vista. Sin embargo, los métodos de binarización y de costo con normalización de histogramas midieron la frecuencia respiratoria con poco error porcentual.

En resumen, el método con el mejor comportamiento es el del costo con histogramas normalizados. Los métodos binarios y el método del costo sin normalización de histogramas parecieran complementarse entre sí en algunas ocasiones, obteniendo estos últimos errores

porcentuales totales muy cercanos entre sí.

5.2.4. Casos Atípicos

Existen cuatro sujetos que no fueron reportados entre los resultados del Cuadro 5. En estos casos, correspondientes a los sujetos S_{27} , S_{28} , S_{29} y S_{30} , surgen condiciones atípicas con las cuales los métodos no logran lidiar adecuadamente.

Los resultados de las mediciones se presentan en el Cuadro 7. Para cada sujeto se presenta el valor real observado y el valor medido (VM) por el método de medición propuesto, junto a su error porcentual (EP). En la última columna del cuadro se encuentran notas relacionadas al sujeto estudiado.

Sujeto	Real	Costo		Costo + Normal.		Bin + ML		Bin + Otsu		Notas
		VM	EP	VM	EP	VM	EP	VM	EP	
S_{27}	20	22	10.00 %	18	10.00 %	22	10.00 %	35	75.00 %	Nariz muy fría. Muy difícil a simple vista.
S_{28}	17	11	35.29 %	11	35.29 %	19	11.76 %	18	5.88 %	Nariz fría y cambios de intensidad por introducción de objetos en el fondo
S_{29}	35	27	22.86 %	29	17.14 %	25	28.57 %	33	5.71 %	El sujeto se mueve varias veces durante la medición
S_{30}	14	18	28.57 %	26	85.71 %	31	121.43 %	20	42.86 %	Nariz muy fría. Muy difícil a simple vista.

Cuadro 7: Resultados de mediciones de los métodos propuestos sobre los casos atípicos.

(ML = Mínimos Locales VM = Valor Medido EP = Error Porcentual)

En el video del sujeto S_{28} , reportado en la tabla, se introdujo un objeto muy frío en el fondo, lo cual ocasionó un reajuste automático por parte de la cámara térmica. Los métodos binarios se vieron menos afectados por este cambio, mientras que el método de costos presentó un error porcentual considerablemente elevado.

El sujeto S_{29} se mueve bruscamente varias veces durante la medición, lo cual ocasiona que el sistema no detecte muchas de las respiraciones o detecte respiraciones inexistentes, dañando la onda de medición con cada movimiento.

Por último, en el caso de los sujetos S_{27} y S_{30} , la temperatura de la nariz de los individuos es muy baja, por lo cual la cámara termográfica le asigna la tonalidad de gris más baja (casi negro) en las imágenes.

En la Figura 20 se muestran dos cuadros tomados como ejemplo del video de respiración del sujeto S_{30} . El cuadro de la subfigura 20(a) corresponde a una inhalación, mientras que el cuadro de la subfigura 20(b) corresponde a una exhalación.

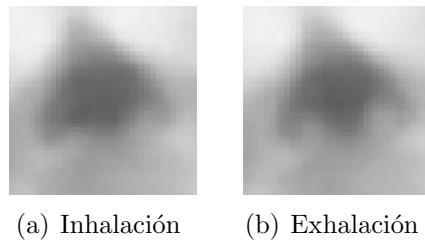


Figura 20: Ejemplos de cuadros de inhalación y exhalación del sujeto S_{30}

La diferencia entre ambos cuadros no es evidente. Adicionalmente, en el primer cuadro se hace complicado distinguir la diferencia de tonalidad entre la nariz y el orificio inhalando, lo cual conduce los métodos propuestos a fallar.

Estos sujetos se consideran casos atípicos, con los cuales ninguno de los métodos propuestos pudo lidiar correctamente. Por lo tanto, no fueron incluidos entre los sujetos reportados en el Cuadro 5.

Capítulo 6

Conclusiones y Recomendaciones

Para este trabajo se diseñó e implementó la aplicación Cliente–Servidor necesaria para teleoperar un robot de Búsqueda y Rescate a través de Internet y permitir que el proceso de teleoperación sea monitoreado por expertos en distintas ubicaciones. Dicha aplicación cuenta con la estabilidad e interfaz necesarias para que pueda ser utilizada por cualquier usuario, permitiendo que personas que desconozcan la aplicación puedan llevar a cabo sus objetivos con la mayor rapidez y comodidad posibles.

Sin embargo, a pesar que la aplicación resulta sencilla de utilizar para usuarios sin entrenamiento previo, se aconseja entrenamiento y trabajo en equipo para garantizar una mejor utilización de la misma. Adicionalmente, la viabilidad de este tipo de aplicaciones en situaciones reales aún es cuestionable, dado que no se hicieron pruebas de campo en escenarios de Búsqueda y Rescate.

Gracias a la librería *Java Media Framework*, fue posible garantizar una buena calidad de audio y video en todo momento, lo cual es crucial a la hora de introducir un robot de Búsqueda y Rescate a un escenario de riesgo. Además, el tiempo de respuesta del robot al ser teleoperado a través de Internet es inmediato. Esto contribuye a que las aplicaciones diseñadas sean herramientas útiles, que permitan proyectar al rescatista dentro de la zona de riesgo, manteniendo su integridad física fuera de peligro.

Dado que se contaba con una cámara termográfica, se decidió añadir a la aplicación una sección para ayudar a los rescatistas a medir la frecuencia respiratoria de las víctimas que podrían encontrarse durante la búsqueda con el robot. Los trabajos sobre mediciones de respiración consultados [13, 7, 21] contaban con equipos sofisticados y aplicaciones que les permitían calcular con precisión la frecuencia respiratoria de los sujetos examinados. Sin embargo, estos trabajos sólo funcionan bajo condiciones muy controladas, y no se hace mención al problema de las narices frías, entre otros.

En este trabajo se comparan distintas propuestas para medir la frecuencia respiratoria de víctimas en ambientes no controlados. La propuesta con el mejor desempeño es aquella que consiste en utilizar una función heurística de costo, que se aplica sobre los histogramas normalizados de los cuadros del video. Esta propuesta tiene un error porcentual de 4.88 %, si no se consideran los casos atípicos como la introducción de objetos de muy baja temperatura en la imagen y sujetos que presentan mucho movimiento.

Una manera efectiva de combatir el caso atípico en el cual se introduce un objeto frío que ocasiona un reajuste por parte de la cámara termográfica, sería utilizar una cámara que pueda ser calibrada, ajustado su rango de medición entre dos temperaturas fijas. Esto probablemente ocasionaría que los métodos propuestos sean aún más precisos al calcular la frecuencia respiratoria de alguna víctima.

Si no se consideran los casos atípicos antes mencionados, se podría decir que el método propuesto se comporta de manera aceptable en situaciones en las cuales no se puede controlar la temperatura de la habitación, la pose del individuo, los movimientos de éste durante la medición, la distancia entre el individuo y la cámara termográfica, entre otras condiciones, como se hace en los trabajos consultados. Es probable que la medición de la frecuencia respiratoria también se vea afectada por ruido en la imagen, pero ello no se pudo constatar con exactitud en los experimentos realizados.

Adicionalmente, se tiene que la frecuencia respiratoria calculada por la aplicación corresponde a una cifra que ha sido truncada al convertirse de flotante¹ a entero. Por lo tanto, es posible que de haberse redondeado el valor de la frecuencia respiratoria, en lugar de truncarse sus decimales, se podría haber obtenido el valor real de las mediciones en muchos de los casos en los cuales la medición obtenida difiere por una unidad del valor real.

Para medir la frecuencia respiratoria de una víctima, es necesario que el teleoperador designe un rectángulo sobre la imagen, que corresponda al área de interés sobre la cual se llevará a cabo la medición de respiración. Dicha selección podría ser llevada a cabo por la aplicación, de manera autónoma. Sin embargo, el gran valor de la vida de las víctimas no

¹La cantidad de respiraciones por minuto se obtiene mediante una división de valores, lo cual resulta en un número flotante (con decimales)

amerita el riesgo de una aplicación autónoma, que podría cometer algún error al hacer la selección. Por lo tanto, se prefiere dejar en manos del operador la selección del área de interés, lo cual asegura que se esté realizando la medición de la mejor manera posible, minimizando las probabilidades de error.

Se debe recordar que la participación de robots en el mundo de la Búsqueda y el Rescate es bastante reciente, por lo que aún se pueden hacer muchas colaboraciones a este ámbito. Durante la investigación se hizo evidente que la tecnología existente no es lo suficientemente poderosa como para poder introducir robots autónomos a hacer búsquedas entre los escombros, razón por la cual éstos deben ser teleoperados a la hora de llevar a cabo sus tareas.

En resumen, los aportes más importantes de este trabajo serían:

- El diseño y la implementación de una aplicación que permite un control estable del robot y sus funciones, utilizando la librería JMF. Esto permite al equipo de rescate evaluar zona de riesgo desde una ubicación segura.
- La medición de la frecuencia respiratoria de víctimas en situaciones no estructuradas, mediante el método de costos sobre histogramas normalizados. El error porcentual total de este método es de 4.88% y es competitivo con el error de trabajos consultados [13, 7, 21], realizados en ambientes controlados. La diferencia entre esta investigación y los trabajos consultados radica en que los algoritmos de medición de frecuencia respiratoria diseñados e implementados para esta investigación logran medir la frecuencia respiratoria de las víctimas sin considerarse la temperatura del ambiente, la posición de las víctimas y su distancia a la cámara y otros factores que son controlados en las investigaciones consultadas, lo cual implica que los algoritmos propuestos en esta investigación son más robustos que los de los trabajos consultados

Finalmente, las futuras investigaciones que elijan desarrollar las ideas propuestas en este trabajo deben considerar la posibilidad de extender los métodos propuestos para la medición de respiración en ambientes no estructurados, que puedan adaptarse a los casos atípicos, y que sean más robustos y eficientes, minimizando su error porcentual. Adicionalmente, se

recomienda a estos trabajos considerar la posibilidad de combinar distintos métodos de medición de respiración, como por ejemplo movimiento torácico, obteniendo un mejor estimado de la frecuencia respiratoria.

Bibliografía

- [1] Java de Sun Microsystems. <http://java.sun.com/>, visitado el 18 de Octubre de 2008.
- [2] Java Swing. <http://java.sun.com/j2se/1.5.0/docs/guide/swing/>, visitado el 18 de Octubre de 2008.
- [3] Aníbal Ollero Baturone. *Robótica. Manipuladores y Robots Móviles*. Marcombo, Ediciones Técnicas, Barcelona, España, 2007.
- [4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, November 1986.
- [5] Jennifer Carlson y Robin Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3):423–437, 2005.
- [6] J. Casper. Human-robot interactions during the robot-assisted urban search mid rescue response at the world trade center. Tesis de Maestría, Universidad de Florida del Sur, Tampa, FL, 2002.
- [7] Jin Fei y Ioannis Pavlidis. Virtual thermistor. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, págs. 250–253, 2007.
- [8] P. Fiorini y R. Oboe. Internet-based telerobotics: Problems and approaches. In *Proceedings 8th International Conference on Advanced Robotics, 1997. ICAR '97.*, 1997.
- [9] E. F. Greneker. Radar sensing of heartbeat and respiration at a distance with security applications. In R. Trebits y J. L. Kurtz, editors, *Proc. SPIE Vol. 3066, p. 22-27, Radar Sensor Technology II, Robert Trebits; James L. Kurtz; Eds.*, volumen 3066 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, págs. 22–27, June 1997.
- [10] M. Micire. Analysis of the robotic-assisted search and rescue response to the world trade center disaster. Tesis de Maestría, Universidad de Florida del Sur, Tampa, FL, 2002.
- [11] Robin Murphy, M. Micire y Jennifer Casper. Issues in intelligent robots for search and rescue. In *SPIE*, volumen 4024, Tampa, FL, 2000.
- [12] Robin R. Murphy. Marsupial and shape-shifting robots for urban search and rescue. *IEEE Intelligent Systems*, 15(2):14–19, 2000.
- [13] R. Murthy, I. Pavlidis y P. Tsiamyrtzis. Touchless monitoring of breathing function. *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, 1:1196–1199, 2004.

- [14] Mark Nixon y Alberto Aguado. *Feature Extraction in Computer Vision and Image Processing*. Newnes, Gran Bretaña, 1ra edición, 2002.
- [15] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, Enero 1979.
- [16] Jean Scholtz, Jeff Young, Jill L. Drury y Holly A. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *ICRA*, págs. 2327–2332. IEEE, 2004.
- [17] L.G. Shapiro y G.C. Stockman. *Computer Vision*. Prentice Hall, 2001.
- [18] Nanfei Sun, Ioannis Pavlidis, Marc Garbey y Jin Fei. Harvesting the thermal cardiac pulse signal. In *MICCAI (2)*, págs. 569–576, 2006.
- [19] Andrew S. Tanenbaum. *Redes de Computadoras*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 4ta edición, 2003.
- [20] L. J. van Vliet, I. T. Young y G. L. Beckers. A nonlinear laplace operator as edge detector in noisy images. *Comput. Vision Graph. Image Process.*, 45(2):167–195, 1989.
- [21] Zhen Zhu, Jin Fei y Ioannis Pavlidis. Tracking human breath in infrared imaging. *Fifth IEEE Symposium on Bioinformatics and Bioengineering, 2005. BIBE 2005.*, págs. 227–231, 2005.

Apéndice A

Paquetes de Control y Respuesta del Robot

Byte	Función
1	Banda de tracción izquierda
2	Banda de tracción derecha
3	Luces
4	Arreglo de bits de control 1 (ver Cuadro 10)
5	Arreglo de bits de control 2 (ver Cuadro 11)

Cuadro 8: Estructura de un Paquete de Control

Byte	Función
1	Batería
2	Temperatura
3	Inclinación (valores altos)
4	Inclinación (valores bajos)
5	Ángulo de erección del Robot
6	Joystick Y (control)
7	Joystick X (control)
8	Arreglo de bits de estado (ver Cuadro 10)
9	Luces
10	Corriente del motor de inclinación de la cámara (mA)
11	Corriente del motor de erección del robot (mA)
12	Corriente consumida por las luces (mA)
13	Corriente consumida por banda de tracción izquierda (mA)
14	Corriente consumida por banda de tracción derecha (mA)
15	Corriente consumida por el enfoque (mA)
16	Arreglo de bits de estado (ver Cuadro 11)

Cuadro 9: Estructura de un Paquete de Respuesta

Bit	Función
0	Autofocus (enfoque automático) encendido
1	Lásers encendidos
2	Desenfocar
3	Enfocar
4	Acostar al Robot
5	Erguir al Robot
6	Girar cámara hacia abajo
7	Girar cámara hacia arriba

Cuadro 10: Bits de Control 1

Bit	Función
0	Luz de advertencia del control
1	Zoom (acercar)
2	Zoom (alejar)
3	Velocidad de Zoom

Cuadro 11: Bits de Control 2

Apéndice B

Manual de Usuario de la Aplicación

Para poder teleoperar un robot a través de Internet es necesario ejecutar dos aplicaciones: La aplicación Cliente y la aplicación Servidor.

La aplicación Servidor se ejecuta en la computadora a la cual se encuentra conectado el robot. El Servidor recibe toda la información del robot y la envía al Cliente, quien a su vez envía información al Servidor.

En otras palabras, la aplicación Cliente sólo puede funcionar si está conectada a la aplicación Servidor, a través de Internet. Estas aplicaciones se comunican entre sí constantemente, permitiendo al Cliente teleoperar o monitorear al robot, sin la necesidad de encontrarse en el mismo lugar que el robot.

El manual se divide en tres secciones: Aplicación Servidor, Aplicación Cliente Operador y Aplicación Cliente Monitor. En cada una de estas secciones se explica detalladamente cómo se debe configurar la aplicación mencionada y su interfaz. También se presentan las recomendaciones a seguir en caso de fallas.

B.1. Aplicación Servidor

La Aplicación Servidor funciona como el centro de operaciones para las demás aplicaciones Cliente. Es por ello que esta aplicación debe ejecutarse en la computadora en la cual se encuentra conectado el robot.

B.1.1. Configuración de la Aplicación

Para asegurar una correcta ejecución de la Aplicación Servidor, se recomienda seguir los siguientes pasos:

1. Conectar el control del robot a la computadora. Para que la aplicación pueda funcionar correctamente es necesario conectar lo siguiente:

- Conectar el control a la computadora mediante un cable USB. La aplicación funciona mejor si se ha conectado el cable USB en el puerto USB principal de la computadora.
 - Conectar el cable de video compuesto (generalmente con conectores de color amarillo) al control del robot y a la tarjeta de video de la computadora.
 - Conectar 2 cables de audio: El primero desde la salida de audio de la tarjeta de sonido de la computadora hacia la entrada de audio del control del robot. El segundo desde la salida de audio del control del robot hacia la entrada de audio de la tarjeta de sonido de la computadora.
2. Ejecutar la Aplicación Servidor en la computadora. La manera de hacerlo depende del sistema operativo (Linux, Mac OS X, Windows, etc.) que se esté utilizando. La versión actual de la Aplicación Servidor no funciona en Mac OS X.
 3. Proveer a la aplicación un número de puerto base para establecer comunicación con los Clientes. En caso que no sepa cuál utilizar, se recomienda proporcionar el puerto predeterminado 6666.

Una vez seguidos estos pasos, el Servidor comienza su ejecución. A partir de este momento se pueden ejecutar los Clientes.

B.1.2. Interfaz de la Aplicación

La Aplicación Servidor no muestra una interfaz gráfica durante su ejecución.

B.1.3. Recomendaciones en Caso de Fallas

A continuación se presenta una lista con las recomendaciones a seguir en caso de surgir una falla en el Servidor:

- Verificar que se haya conectado correctamente el cable de video compuesto a la tarjeta de video de la computadora. Verificar también que se haya conectado el cable USB en el puerto USB principal de la misma.

- Se recomienda deshabilitar los Firewalls de la computadora y de la red, ya que podrían estar bloqueando la comunicación en los puertos utilizados por la aplicación.
- Un problema común es que no hay suficientes puertos libres en la computadora para establecer la comunicación con los Clientes. Se recomienda cerrar todas las aplicaciones que no estén en uso, sobre todo aquellas que se conecten a Internet.
- Cada vez que se conecta un Cliente al Servidor, el consumo de memoria RAM aumenta. La cantidad límite de Clientes que pueden conectarse a cada Servidor depende de cada computadora. En caso que la calidad de video se vea muy afectada en los Clientes, se recomienda reducir la cantidad de Clientes conectados al Servidor.

Se recomienda volver a ejecutar el Servidor en caso que ocurra una falla. El Servidor podría continuar su ejecución si la falla es poco relevante, pero esto podría perjudicar a los Clientes conectados.

B.2. Aplicación Cliente Operador

La Aplicación Cliente Operador permite al teleoperador conectarse al Servidor, para poder enviar comandos que cambien la ubicación y el estado físico del robot, así como iniciar mediciones de frecuencia respiratoria, en caso de encontrarse alguna víctima.

B.2.1. Configuración de la Aplicación

Para asegurar una correcta ejecución de la Aplicación Cliente Operador, se recomienda seguir los siguientes pasos:

1. Ejecutar la Aplicación Cliente Operador en la computadora. La manera de hacerlo depende del sistema operativo (Linux, Mac OS X, Windows, etc.) que se esté utilizando.
2. Proporcionar a la aplicación la dirección IP del Servidor y el **número de puerto utilizado por el Servidor**.

Una vez seguidos estos pasos, la Aplicación Cliente Operador debería haber establecido una conexión con la Aplicación Servidor. A partir de este momento debería mostrarse la interfaz de la Aplicación, cuya utilización será descrita en la próxima sección.

Si la interfaz de la aplicación no se muestra al finalizar los pasos, acuda a la subsección B.2.3 de este manual.

B.2.2. Interfaz de la Aplicación

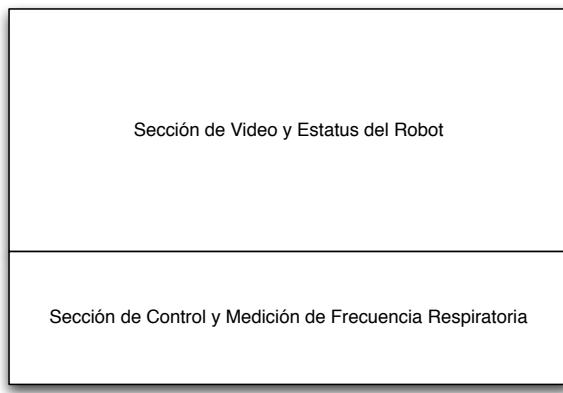


Figura 21: Secciones de la Interfaz de la Aplicación Cliente Operador

La interfaz de la Aplicación Cliente Operador cuenta con dos secciones principales: la sección de video y estatus del robot, y la sección de control y medición de frecuencia respiratoria, como se muestra en la Figura 21.

Sección de Video y Estatus del Robot

Esta sección de la interfaz, ubicada en la parte superior de la misma, presenta el nivel de batería del robot, en el panel a la izquierda.

El nivel de la batería está representado por una imagen de una pila, que se vacía a medida que se agota la batería del robot. La batería se muestra de color verde si está cerca de estar totalmente cargada, y de color rojo si está baja de carga.

En el panel central se presenta el video enviado por el Servidor. Sobre este panel es posible dibujar un rectángulo, designando el área de interés necesaria para hacer mediciones de respiración sobre la cámara térmica.

Sección de Control y Medición de Frecuencia Respiratoria

Esta sección de la interfaz, ubicada en la parte inferior de la misma, presenta dos viñetas: *Search* y *Rescue*.

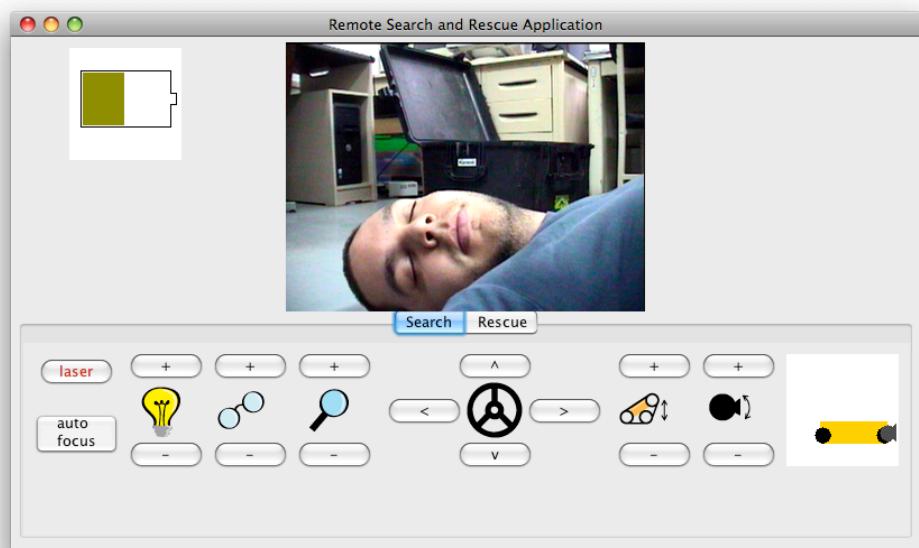


Figura 22: Controles de la viñeta *Search*

Al hacer click sobre la viñeta *Search* (ver Figura 22), se despliegan los controles de navegación y manejo del robot.

Estos controles permiten al usuario mover al robot, así como encender y apagar los láser y el enfoque automático (por medio del botón autofocus). También se incluyen controles para aumentar o disminuir la intensidad de las luces, enfocar y desenfocar la cámara, acercar y alejar la cámara, erguir y acostar al robot y girar la cámara hacia arriba y hacia abajo. Consulte el Cuadro 12 para una descripción detallada de los controles.

	Controles de intensidad de la luz del Robot
	Controles de enfoque de la Cámara
	Controles de zoom (acercamiento) de la Cámara
	Controles de navegación del Robot
	Controles para erguir y acostar al Robot
	Controles de giro de la cabeza (cámara) del Robot

Cuadro 12: Descripción de los Controles de la Viñeta *Search*
del Cliente Operador

Junto a estos controles se presenta un diagrama de vista lateral del robot, que se actualiza para representar cuán erguido está el robot, así como la inclinación de su cámara.

La viñeta *Search* está seleccionada al iniciar la aplicación, mostrando al usuario los controles mencionados.

Al hacer click sobre la viñeta *Rescue* (ver Figura 23) se presenta al usuario la información de medición de respiración.

En este caso se cuenta con un botón que permite iniciar y detener la medición de respiración, siempre y cuando se haya seleccionado un rectángulo con el área de interés sobre el video.

Al iniciarse la medición de respiración, se bloquean los demás controles de la interfaz,

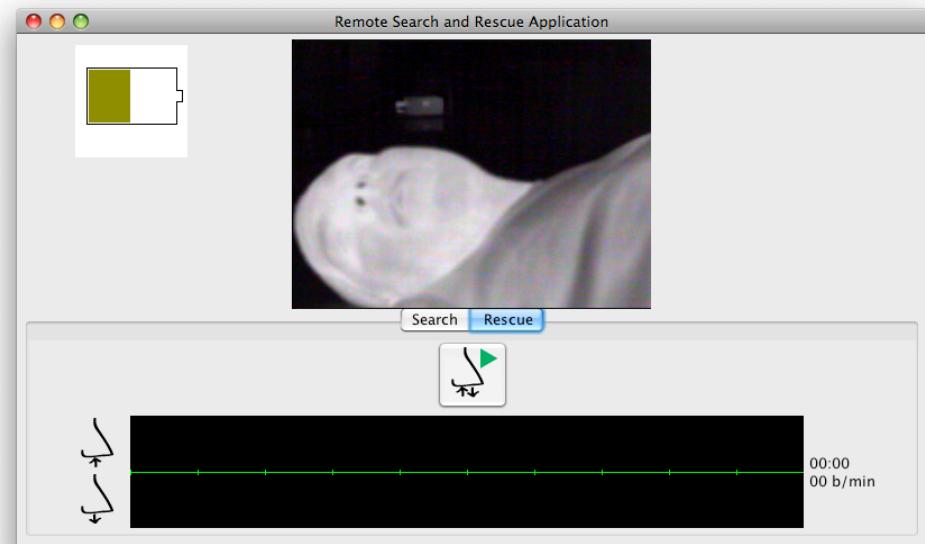


Figura 23: Información de la viñeta *Rescue*

hasta que el Operador decida dar por terminada la medición.

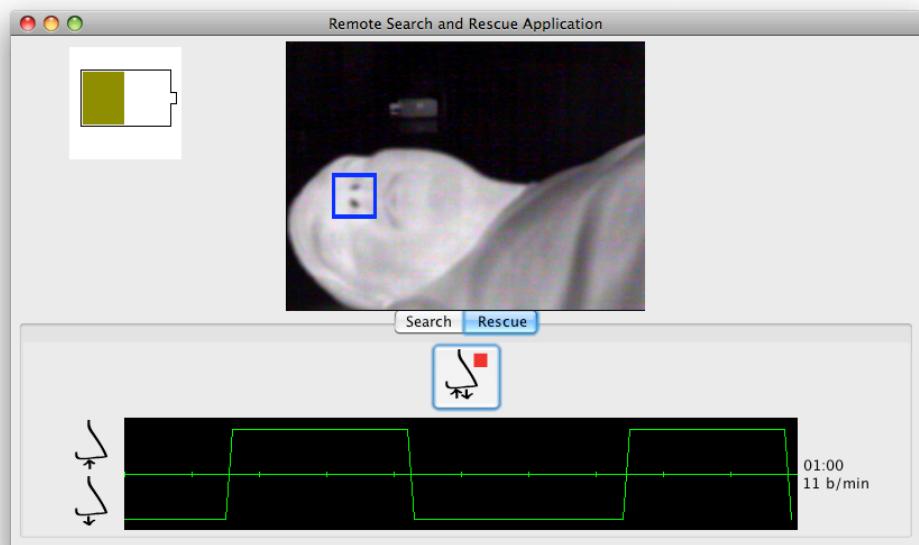


Figura 24: Información de la viñeta *Rescue* durante una medición de respiración

En el Cuadro 13 se describen los iconos de la Viñeta *Rescue* de la Interfaz.

	Inhalación en el gráfico
	Exhalación en el gráfico
	Comenzar medición de frecuencia respiratoria
	Detener medición de frecuencia respiratoria

Cuadro 13: Descripción de los Iconos de la Viñeta *Rescue* del Cliente Operador

La información de la medición de la respiración se presenta al usuario con un gráfico en el cual se van dibujando líneas que representan las inhalaciones y exhalaciones de la víctima durante diez segundos (ver Figura 24). Junto a este gráfico se incluye un cronómetro y, después de haber transcurrido un tiempo prudencial, un estimado de la cantidad de respiraciones por minuto (*bpm* por sus siglas en inglés) de la víctima.

Esta medición se calcula en un lapso de tiempo de a lo sumo un minuto, y se traslada al último minuto transcurrido, a medida que avanza el tiempo de medición. En otras palabras, la frecuencia respiratoria que se presenta corresponde siempre al último minuto transcurrido.

B.2.3. Recomendaciones en Caso de Fallas

A continuación se presenta una lista con las recomendaciones a seguir en caso de surgir una falla en el Cliente Operador:

- Verificar la dirección IP del Servidor y el número de puerto proporcionados. Recuerde que el número de puerto debe coincidir con el proporcionado al Servidor. El valor predeterminado es 6666.
- Se recomienda deshabilitar los Firewalls de la computadora y de la red, ya que podrían

estar bloqueando la comunicación en los puertos utilizados por la aplicación.

- La calidad del video se ve afectada por la memoria RAM disponible en la computadora. Se recomienda cerrar todas las aplicaciones que no estén en uso, para liberar memoria RAM.
- La cantidad de Clientes conectados al Servidor también afecta la calidad del video. Si hay demasiados Clientes conectados al Servidor, podría ser necesario cerrar algunos.

Se recomienda volver a ejecutar el Cliente Operador en caso que ocurra una falla.

B.3. Interfaz de la Aplicación del Cliente Monitor

La Aplicación Cliente Monitor permite al experto conectarse al Servidor, para poder monitorear la búsqueda del robot, llevada a cabo por el teleoperador, y las mediciones de frecuencia respiratoria iniciadas por el mismo.

B.3.1. Configuración de la Aplicación

Para asegurar una correcta ejecución de la Aplicación Cliente Monitor, se recomienda seguir los siguientes pasos:

1. Asegurarse que se haya conectado el Cliente Operador al Servidor antes de conectar los monitores.
2. Ejecutar la Aplicación Cliente Monitor en la computadora. La manera de hacerlo depende del sistema operativo (Linux, Mac OS X, Windows, etc.) que se esté utilizando.
3. Proporcionar a la aplicación la dirección IP del Servidor y el **número de puerto utilizado por el Servidor**.

Una vez seguidos estos pasos, la Aplicación Cliente Monitor debería haber establecido una conexión con la Aplicación Servidor. A partir de este momento debería mostrarse la interfaz de la Aplicación, cuya utilización será descrita en la próxima sección.

Si la interfaz de la aplicación no se muestra al finalizar los pasos, acuda a la subsección B.3.3 de este manual.

B.3.2. Interfaz de la Aplicación

La interfaz de la Aplicación Cliente Monitor es una versión simplificada de la interfaz de la Aplicación Cliente Operador. Esta interfaz cuenta con dos secciones principales: la sección de video y la sección de monitoreo de medición de frecuencia respiratoria, como se muestra en la Figura 25.

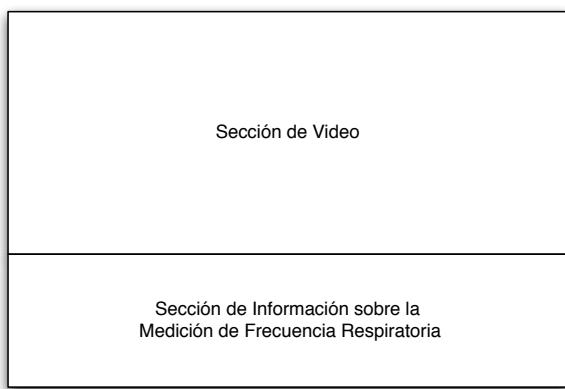


Figura 25: Secciones de la Interfaz de la Aplicación Cliente Monitor

Sección de Video

En el panel central se presenta el video enviado por el Servidor. El panel muestra el video capturado por la cámara del robot o la cámara térmica, dependiendo de la cámara que esté siendo utilizada por el teleoperador.

Sección de Información sobre la Medición de la Frecuencia Respiratoria

Esta sección de la interfaz, ubicada en la parte inferior de la misma, presenta al usuario la información de las mediciones de respiración hechas por el teleoperador.

La información de la medición de la respiración se presenta al usuario con un gráfico en el cual se van dibujando líneas que representan las inhalaciones y exhalaciones de la víctima durante diez segundos. Junto a este gráfico se incluye un cronómetro y, después de haber

transcurrido un tiempo prudencial, un estimado de la cantidad de respiraciones por minuto (*bpm* por sus siglas en inglés) de la víctima. En el Cuadro 14 se describen los iconos asociados al gráfico.

	Inhalación en el gráfico
	Exhalación en el gráfico

Cuadro 14: Descripción de los Iconos asociados al Gráfico del Cliente Monitor

Esta información se actualiza en los Clientes monitor cada vez que el Cliente Operador la solicita, y se actualizará mientras el Operador esté haciendo mediciones de respiración.

Dicha medición se calcula en un lapso de tiempo de a lo sumo un minuto, y se traslada al último minuto transcurrido, a medida que avanza el tiempo de medición. En otras palabras, la frecuencia respiratoria que se presenta corresponde siempre al último minuto transcurrido.

B.3.3. Recomendaciones en Caso de Fallas

A continuación se presenta una lista con las recomendaciones a seguir en caso de surgir una falla en el Cliente monitor:

- Asegurarse que el Cliente Operador se haya conectado al Servidor antes que cualquier otro Cliente monitor.
- Verificar la dirección IP del Servidor y el número de puerto proporcionados. Recuerde que el número de puerto debe coincidir con el proporcionado al Servidor. El valor predeterminado es 6666.
- Se recomienda deshabilitar los Firewalls de la computadora y de la red, ya que podrían estar bloqueando la comunicación en los puertos utilizados por la aplicación.

- La calidad del video se ve afectada por la memoria RAM disponible en la computadora. Se recomienda cerrar todas las aplicaciones que no estén en uso, para liberar memoria RAM.
- La cantidad de Clientes conectados al Servidor también afecta la calidad del video. Si hay demasiados Clientes conectados al Servidor, podría ser necesario cerrar algunos.

Se recomienda volver a ejecutar el Cliente monitor en caso que ocurra una falla.

Apéndice C

Métodos y Filtros Rechazados

Los métodos de detección de bordes que se intentaron utilizar para facilitar la detección de respiración suelen ser muy efectivos. Sin embargo, en una gran cantidad de casos estudiados en este trabajo los resultados eran menos que satisfactorios.

Al aplicarse el algoritmo de Canny, se detectaron como bordes el cabello, contorno y otras características del sujeto, dificultando su utilización para la medición de la frecuencia respiratoria. Se intentó aplicar filtros de Laplace para disminuir el ruido en las imágenes y facilitar la detección de los cambios de gradiente. Sin embargo, existe una gran cantidad de sujetos que presentan cambios marcados de temperatura facial, que hacen que el filtro descarte información que puede ser importante para el algoritmo de medición (ver Figura 27). Estos métodos serán explicados con mayor detalle a continuación.

C.1. Filtros de Laplace

El método de Laplace es utilizado como filtro de baja frecuencia. En este caso la utilidad de dicho método se ve reducida, ya que aunque en casos como el mostrado en la Figura 26 se obtienen resultados que pueden ser útiles, en otros como el de la Figura 27 se pierde toda la información relevante para los algoritmos de medición de frecuencia respiratoria.

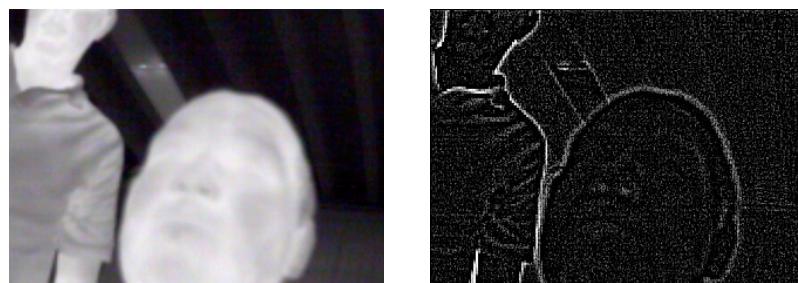
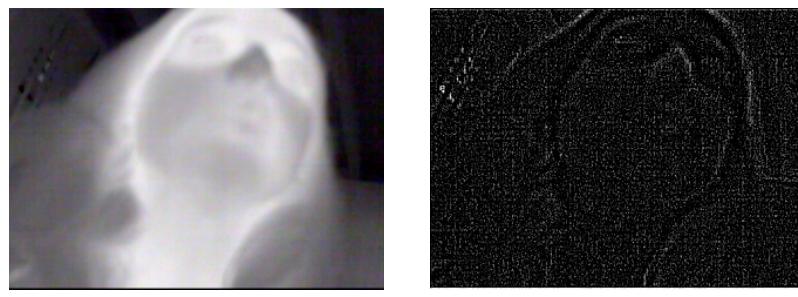


Figura 26: Imágenes antes y después de la aplicación del filtro de Laplace (Nariz Tibia)



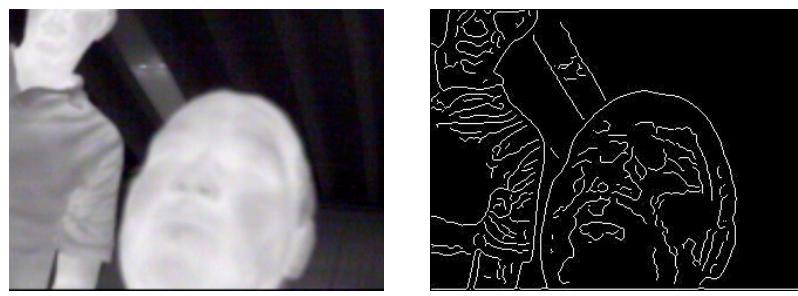
(a) Imagen Original

(b) Imagen con Filtros Laplace

Figura 27: Imágenes antes y después de la aplicación del filtro de Laplace (Nariz Fría)

C.2. Método de Canny

El método de Canny es sumamente efectivo en la detección de bordes en las imágenes. Sin embargo, en los casos de estudio para este trabajo existen demasiados bordes definidos, como se puede observar en la Figura 28. La detección de una cantidad muy alta de bordes hace que la aplicación de este método sea inviable para la detección visual de patrones de respiración utilizando cámaras térmicas.



(a) Imagen Original

(b) Imagen con Canny

Figura 28: Imágenes antes y después de la aplicación de Canny

Apéndice D

Sujetos Analizados

S_1		S_2	
			
S_3		S_4	
S_5		S_6	
S_7		S_8	
S_9			
S_{10}			
S_{11}		S_{12}	
S_{13}		S_{14}	

S_{15}		S_{16}	
S_{17}		S_{18}	
S_{19}		S_{20}	
S_{21}		S_{22}	
S_{23}		S_{24}	

