PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DETECCIÓN Y SEGUIMIENTO DE MANOS EN VIDEOS DIGITALES UTILIZANDO COMPUTADORES Y MINI-COMPUTADORES

Tesis pata optar el Título de Licenciado en Ingeniería Electrónica, que presenta el bachiller:

Pedro Arturo Cisneros Velarde

ASESOR: Dr. Paul Rodríguez Valderrama

Lima, Julio del 2013

A mis padres, hermano y abuelos.

Agradecimientos

A Dios por tantas bendiciones en mi vida, a mis padres por su permanente e incondicional apoyo, a mi hermano y abuelos por todo el cariño brindado.

A mi asesor y a los miembros del jurado que por sus acertadas observaciones han permitido la culminación exitosa de esta tesis.

A mis compañeros y profesores que han significado un apoyo a lo largo de mi carrera universitaria.

Índice general

Índice de figuras

Índice de cuadros

Introducción

El problema del seguimiento de manos o *hand tracking* puede definirse como la capacidad de un sistema computacional de poder reconocer las manos de un individuo (usuario) y hacerles un seguimiento en todo momento. El interés por el estudio del movimiento de las manos se debe a dos particularidades. En primer lugar, se debe a que las manos son protagonistas en la realización de varias tareas diarias del ser humano, pues las manos son un distintivo de las diferentes actividades humanas. Las manos permite la manipulación de objetos; de lo cual se basa una gran dimensión de la interactividad del hombre con sus diferentes herramientas de trabajo [1]. No es de sorprender que, con el reconocimiento del movimiento de las manos, se puedan reconocer varias actividades de las personas: comer, saludar, martillar, apuñar, señalar, etc. En segundo lugar, las manos, junto con el rostro, son los dos mayores indicadores gestuales dentro de la comunicación no verbal; lo cual indica que en las manos hay un gran despliegue de diferentes gestos, seas y apariencias, y por tanto, tengan una gran riqueza de significado comunicativo.

En las dos razones de la importancia del estudio del seguimiento de manos expuestas anteriormente, es posible ver que las manos poseen una mayor libertad de movimientos y posturas posibles a tomar con respecto a las otras partes del cuerpo. Esto supone, entre otras cosas, que el seguimiento de manos tiene que lidiar con una serie de particularidades intrínsecas a estos miembros: oclusiones entre ambas manos, cambios violentos y aparentes de su forma, cambios rápidos en su desplazamiento, etc. En consecuencia, esta libertad de las manos les otorga una mayor capacidad expresiva e importancia comunicativa, descriptiva e indicativa que han motivado distintas investigaciones respecto a su seguimiento y reconocimiento para aplicaciones de interacción de las personas con las computadoras [2].

Las implementaciones del seguimiento de manos varían en cuanto a su complejidad y velocidad de seguimiento, con lo cual requieren distintos esfuerzos computacionales e incluso el uso adicional de indumentaria por parte de la persona a quien se desea seguir el movimiento de las manos (usuario) [3]. Gran parte de estas implementaciones no solo concluyen con la obtención de la posición de las manos en un eterminado momento, sino que adicionalmente pueden inclusive reconocer gestos específicos formados [4]. Por otro lado, estas distintas implementaciones suponen el poder contar con un computador capaz de cubrir las demandas necesarias de procesamiento y recurso que demanda la tarea, en lo cual se ha encontrado que la gran mayoría supone el uso de una computadora personal o *desktop*. Si el procesamiento requiere un menor tiempo, más tiempo se tiene luego para realizar algún otro procesamiento posterior. Especial interés también se tiene en la implementación sobre sistemas con recursos y capacidad computacional limitados, pues requieren un crítico equilibrio entre precisión, robustez y velocidad del seguimiento. Entre estos, tenemos implementaciones hechas en dispositivos móviles y computadoras embebidas [5].

La tesis presente pretende implementar un sistema de seguimiento de manos. El aporte de la tesis recaerá tanto en el algoritmo a implementar para el seguimiento, como en las características y comparaciones de su implementación en una computadora personal y una computadora embebida de bajo costo. Este aporte representa un reto porque requiere mantener un equitativo equilibrio entre velocidad de seguimiento, robustez y complejidad del seguimiento.

Capítulo 1

Marco de estudio y problemática del estudio

1.1. Estado del Arte

1.1.1. Presentación del Asunto de Estudio

La importancia del seguimiento de manos radica en el protagonismo de las manos dentro de los quehaceres cotidianos de las personas y su gran riqueza de connotación simbólica comunicativa. Esto ha permitido utilizarla en diversas aplicaciones: enviar señas al computador para establecer una interacción comunicativa, manipular objetos virtuales, y el poder vigilar el movimiento de las manos de algún individuo con el objetivo de encontrar comportamientos de interés.

Los avances en la disciplina del seguimiento de manos tienen una larga trayectoria de décadas de investigación, con una variedad de características distintas en sus métodos, algoritmos e implementaciones. Por ejemplo, existen métodos ”intrusos” para el usuario que requiere el uso de indumentaria adicional especial que facilite el seguimiento: guantes, sensores, etc. (ver [3] para una recopilación de desarrollos existentes hasta hace dos décadas). Por otro lado, están los métodos menos intrusos, donde destaca el seguimiento basado solamente en visión monocular sin el uso de indumentaria adicional.

Por otro lado, también están los factores relacionados con el costo o carga computacional que los algoritmos del seguimiento de manos suponen para la computadora. Por ejemplo, el empleo del filtro de partículas puede suponer una elevada carga computacional que no la hace apto para entornos con menor capacidad computacional [6], lo cual no sucede con otras técnicas de filtrado temporal [7]. Es importante conocer el grado de exactitud que se requiere en la aplicación del seguimiento de la mano porque, dependiendo del algoritmo, un seguimiento menos preciso podra reducir la carga computacional asociada.

Un factor importante relacionado a la implementación del seguimiento de manos es la plataforma *hardware* de implementación, porque ella tiene asociada una serie de restricciones computacionales que influyen críticamente sobre la velocidad y precisión del seguimiento, aspectos importantes si el algoritmo se destina a aplicaciones en tiempo real. Por ejemplo, observar el contraste que existe, en cuanto velocidad y presición, en la implementación del reciente sistema desarrollado por la compañía OMRON para dispositivos móviles [8], con respecto al desarrollado por la empresa GestureTek hace una década [9].

La presente tesis tiene dos implementaciones finales: el uso de una computadora personal (*desktop*) y de la mini-computadora ”MK802” [6]. Además, la implementación de seguimiento no es intrusa (el usuario no requiere ninguna indumentaria o dispositivo adicional) y solo utiliza una cámara de adquisición.

1.1.2. El estado de la investigación

1.1.2.1. Aplicaciones y desarrollos generales

El seguimiento de manos es una aplicación importante dentro de la disciplina de Visión por Computadora [10, 11], con una gran importancia en el campo de *Human-Computer Interaction* (HCI) [12] para la implementación de sistemas computacionales interactivos para el uso humano [13]. Se resalta que en las aplicaciones del seguimiento de manos, ella se perfila como una importante etapa previa a posteriores procesamientos complejos como son el reconocimiento de gestos [14], la reconstrucción tridimensional de las manos [15], entre otros.

Dentro del HCI, el seguimiento de manos se usa en aplicaciones de Realidad Aumentada (AR) y Realidad Virtual (VR). En la Realidad Aumentada, es deseable que el usuario pueda interactuar con los elementos virtuales dentro de ella (seleccionarlos, moverlos y posicionarlos) [10], lo cual comúnmente se realiza por medio de un marcador físico especial, como en el caso de la aplicación ”Augment” para *smartphones* [16]. Sin embargo, otra posibilidad es usar las manos del usuario como los dispositivos de interacción por medio de su seguimiento, como es el caso del sistema ”Handy AR” [5]. Otro ejemplo de interacción en tiempo real es ”SixthSense” [17], en el cual el usuario puede navegar por medio de las manos a través de programas o archivos dentro de algún dispositivo móvil proyectados en el mundo físico.

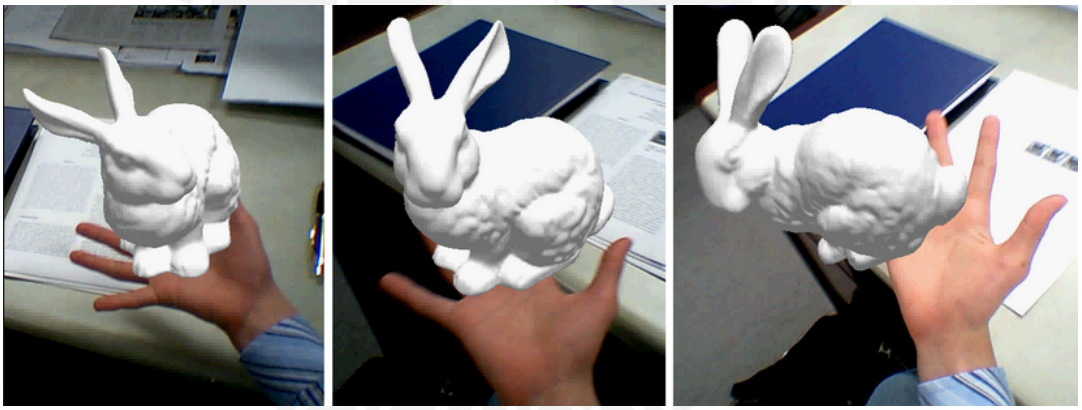


Figura 1.1: Ejemplo de realidad aumentada: secuencia de imágenes donde el sistema Handy AR realiza un seguimiento de manos para ubicar un objeto virtual sobre la mano del usuario [5].

En el caso de la Realidad Virtual, la interacción con el computador puede darse por mecanimos que pueden involucrar a más de un sentido del cuerpo, pero que a su vez puede requerir indumentaria y equipos complejos y costosos [18]. Por ejemplo, para mecanismos de interacción por movimiento que detecten alguna postura del usuario, existen distintos sensores *trackers* que pueden permitir un seguimiento de manos: mecánicos, magnéticos, ultrasónicos y ópticos [18].En particular, los ópticos incluyen a las cámaras de video, las cuales permiten un seguimiento menos intruso para el usuario que puede reducirse al único uso de ellas. Tal es el trabajo [15], donde con una cámara y un guante se localiza y reconstruye la pose de una mano para la realización de alguna acción dentro del mundo virtual en tiempo real.

Una empresa que brinda servicios de *tracking* en general (incluidos el caso de las manos) y que ofrece implementaciones de seguimiento y reconocimiento para sistemas de AR o VR, es *Advanced Realtime Tracking* [19]. Por ejemplo, se ofrecen soluciones de seguimiento de manos orientadas a aplicaciones médicas, como es la captura de movimiento para mediciones de la ergonomía humana.

- **Aplicaciones importantes relacionadas al seguimiento de manos**

Se explican algunas aplicaciones con el objetivo de resaltar la importancia en ellas que podrían desprenderse del tema de estudio de la presente tesis.

Lúdico:

Dentro de las aplicaciones lúdicas existen dos ejemplos comerciales relacionados a las consolas de juego PlayStation y XBOX, donde se hace uso de la visión por computadora gracias a los sensores"Playstation Eye" [20] y "Microsoft Kinect" [21] respectivamente. Por ejemplo, el Microsoft Kinect permite el uso de videojuegos interactivos por medio del seguimiento de diferentes extremidades del cuerpo, incluidas las manos, gracias a que contiene una cámara a color y otros sensores de profundidad [22]. Estas características también han permitido su uso en aplicaciones de seguimiento de manos fuera de lo lúdico e inclusive usando computadoras personales [23], como es el caso de [24].

Otro ejemplo es el sistema "Leap Motion" [25], el cual tiene aplicaciones lúdicas y otras más generales que impliquen una comunicación del usuario con la computadora por movimiento de las manos y gestos denotados por ellas. El sistema requiere la instalación de un periférico de adquisición adicional conectado al computador [26].

Museos:

La aplicación del seguimiento de manos en los "museos interactivos" consiste en que el espectador del museo, por medio del movimiento de las manos y proyecciones de Realidad Aumentada, pueda navegar a través de cualquier información o facilidad multimedia ofrecida por el museo de una manera más intuitiva. De esta manera, se reemplazan las tradicionales formas de navegación como pantallas táctiles y teclados. En [27] se encuentra un avance de hace casi una década que permitía el poder seleccionar ventanas o regiones de algún *display* con simplemente sealar hacia un punto de interés. Actualmente la empresa "lm3labs" comercializa una avanzada plataforma para sistemas interactivos por seguimiento de manos y reconocimiento de gestos destinados exclusivamente a museos [28].

Seguridad:

Otra aplicación es la vigilancia de las actividades en las cajas o puntos de venta de algún centro comercial o supermercado. Esta vigilancia responde a casos existentes en supermercados donde el cliente puede robarse un producto sin registrarlo en la caja, y en varias ocasiones en complicidad con el cajero. Existen empresas que ofrecen soluciones a estos problemas por medio de una distribución estratégica de cámaras y/o la realización de una completa auditoría acerca de la seguridad en la empresa [29]. Sin embargo, la compañía "StopLift" ofrece una solución [30] en el cual, por medio de técnicas de visión por computadora y una cámara de video, se realiza un seguimiento de manos que permite interpretar el comportamiento del cajero y el cliente en búsqueda de algún movimiento sospechoso de robo por parte de ambos de forma automatizada. Por ejemplo, puede detectar si el cajero realizó un robo al no escanear el producto antes de entregarlo al cliente como puede verse en la figura 1.2 [31], o si el cliente escondió algún producto sin conocimiento del cajero. Otros estudios realizan estas tareas por medio de la detección de actividades manuales que estén fuera del patrón rutinario ejercido por el cajero y que por tanto sean sospechosos de un robo [2, 32].

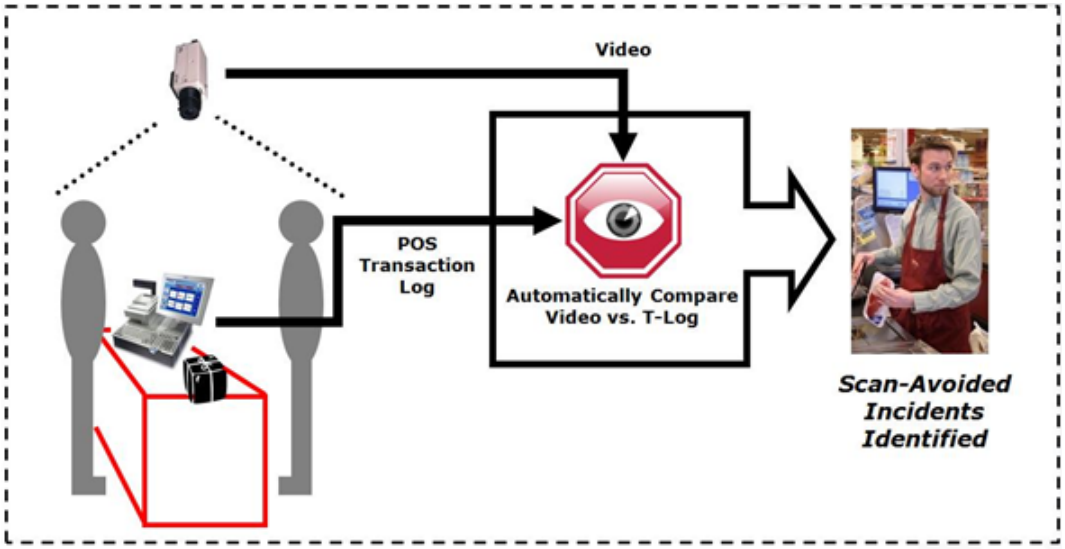


Figura 1.2: Funcionamiento del sistema "StopLift Checkout Vision Systems"[31], una aplicación del seguimiento de manos en vigilancia y seguridad.

Reconocimiento del Lenguaje de señas:

El reconocimiento del lenguaje de señas [33] tiene como objetivo extraer el significado que connota alguna seña hecha por una mano dentro del video, para lo cual necesita saber primero la ubicación de la mano en cada cuadro por medio de su seguimiento [34] (un seguimiento erróneo, genera un reconocimiento erróneo [35]). En el trabajo [14] se encuentra una recopilación de casi dos décadas hecha en el 2005 en el desarrollo de esta aplicación. Un ejemplo de desarrollo que ha sido ampliamente probado es "CopyCAT", un módulo de videojuego que permite el aprendizaje del lenguaje de señas para niños sordos, que inclusive cuenta con una versión usando MS Kinect [36, 37].

1.1.2.2. Enfoques generales del seguimiento de manos

El seguimiento de manos tiene como resultado la obtención de la coordenada 2D de las manos en el cuadro actual, y que visualmente se aprecia por medio de una ventana centrada en dicha coordenada y que debe encerrar la mayor cantidad de píxeles pertenecientes a la mano [7]. De mamera común, la mano se ubica por medio del uso de distintivos o características (*cues*) que permitan identificar alguna propiedad de ella en el cuadro actual, los cuales generalmente caracterizan el color o el movimiento que tienen en el cuadro. A continuación se presenta una clasificación general y común de los distintos enfoques del seguimiento de manos encontrados en los trabajos consultados [2, 4, 12, 26, 38, 39, 40, 41, 42, 43, 44, 15, 45].

- **Enfoques del seguimiento según la realización del reconocimiento de las manos**

Basado en el modelo de la mano: [42, 45]

Es el seguimiento que consiste en la directa comparación de la imagen adquirida de la mano con algún modelo morfológico o anatómico de la mano pre-establecido en memoria [44] de manera *online* u *off-line* [42, 43]. La tarea del seguimiento es identificar qué parámetros que definen a este modelo son identificadas

en las posibles posiciones de las manos en el cuadro actual, de este modo, se da una estimación de la pose de la mano de forma paralela con su detección. Por lo general, la mano ocupa un mayor tamaño entre todos los objetos presentes en el cuadro, ubicándose como el único ”objeto” importante en un primer plano delante de la cámara (*protagonismo en primer plano*) y así disminuir el riesgo de que otros objetos con características similares a la mano puedan dificultar el seguimiento.

Basado en la apariencia de la mano: [2, 45]

Enfoque del seguimiento de manos que no incluye algún proceso de comparación con algún modelo establecido, por lo cual, se opta por usar una serie de descriptores de cómo posiblemente se podría identificar a la mano dentro del cuadro. Sin embargo, los descriptores pueden tener redundancia y generar una mayor cantidad de hipótesis, como también generar falsos positivos. La reducción de hipótesis se resuelve de dos formas: con la inclusión de un método adecuado de discriminación dentro del algoritmo, o con la búsqueda de características muy particulares para identificar a las manos. Esto último se puede lograr mediante el uso adicional de algún tipo de rotulador o guante que tenga un rasgo visual distintivo respecto a acualquier otro objeto de la imagen, de manera que se facilita la identificación de la mano.

Debido a que generalmente las manos no son el único objeto protagónico en el cuadro, pueden estar rodeadas dentro de un ambiente cambiante y con objetos ”muy similares” a él. Por ejemplo, las manos pueden tener un tamaño relativamente pequeño con respecto al cuadro, y confundirse con el movimiento de otras partes del cuerpo (brazos, hombros y cabeza) o tal vez de algún elemento de color similar en el ambiente.

Este es el enfoque en el cual se basa la presente tesis.

- **Enfoques del seguimiento según el procesamiento de las hipótesis**

En cada cuadro existen posibles regiones o hipótesis donde la mano puede estar ubicada y cuyo espacio de búsqueda puede inclusive abarcar a todo el cuadro. La cantidad de hipótesis guarda relación directa con los distintivos usados para identificar la mano y de la existencia adicional de mecanismos auxiliares, como es el uso de indumentaria por parte del usuario.

Basado en rótulos o guantes: [3, 15, 4]

Requiere el uso de indumentaria o algún medio de distinción físico por parte del usuario que rotulan a la mano y ayudan a identificarla mediante los distintivos. Sin embargo, es intruso al usuario por significarle el uso de artefactos adicionales. Estas distinciones físicas pueden restringir la búsqueda de las manos en espacios más reducidos del cuadro, debido a que ellas generalmente involucran características que son exclusivas a las manos y que no se presenta en ningún otro elemento del entorno. Esto reduce la multiplicidad de regiones con características similares a la de las manos, el número de hipótesis, y el número de distintivos para caracterizar a la mano. Ejemplos de indumentaria son el uso de capuchas en los dedos, el uso de guantes, el montaje de espejos o sensores sobre algún guante, etc.

***Barehand tracking* o seguimiento de mano desnuda:** [38, 46]

No requiere el uso de ningún tipo de indumentaria adicional para la mano, simplemente se la captura

”desnuda y natural”, y es muy usada en la detección de ambas manos del usuario sin protagonismo en primer plano. Esta forma de abordar el seguimiento es lo menos intruso al usuario, pero que a la vez supone al algoritmo el tener que lidiar con una mayor cantidad de hipótesis, pues existe mayor posibilidad de tener regiones del cuadro con características similares a las de las manos según los distintivos usados. Una opción para reducir el número de hipótesis es el uso de una mayor cantidad de diferentes tipos de distintivos, de manera que en conjunto permitan caracterizar mejor a la mano. El lidiar con más hipótesis o usar más distintivos involucra mayor procesamiento y por tanto, una mayor carga computacional por parte del seguimiento.

Este es el enfoque en el cual se basa la presente tesis.

1.1.2.3. Problemas y soluciones generales

- **Problemas generales:**

Se exponen tres problemas generales que pueden ocurrir en el caso del *barehand tracking* y que afectan en mayor o menor medida la detección de las manos de acuerdo al contexto en el cual esta se realiza. Esta clasificación se basa en los trabajos consultados [12, 47, 26, 48, 7, 38, 39, 40, 41, 49, 50].

1. **Problemas relacionados al movimiento y aspecto de la mano:** Las manos pueden considerarse como elementos amorfos, no rígidos y cuya apariencia es capaz de cambiar repentinamente entre pocos cuadros sucesivos. Como consecuencia, existen problemas relacionados con la baja resolución o pequeño tamaño que la mano puede presentar respecto a las dimensiones del cuadro, como otros ligados a sus cambios continuos de perspectiva e iluminación. Estos problemas inhiben la identificación de las manos por medio de su forma, textura particular o cambios de iluminación promedio predecibles [51]. Entre otros problemas diversos se presentan:
   - Movimientos fuera de plano: Las manos se ubican fuera del campo visual de la cámara.
   - Cambios de escala y perspectiva: Las manos se acercan o alejan de la cámara, de modo que cambian continuamente de tamaño y dimensión aparente.
   - Poca diferenciación en color y forma con elementos del entorno.
2. **Problemas relacionados a la interacción entre ambas manos:** La oclusión puede darse al anteponere una mano respecto a la otra frente a la visión que la cámara tiene de ellas. Como resultado, existe el riesgo de identificar a ambas manos como si fuera una sola (por compartir ambas la misma posición); y posteriormente, cuando acabe la oclusión, solo se siga a una de ellas.
3. **Problemas en el entorno de las manos:** El entorno que rodea a las manos puede tener elementos con formas de movimiento (entorno dinámico) y aspectos (entorno complejo) muy similares a las manos, los cuales pueden ser falsamente identificados como si fuesen las manos.

- **Solución mediante el uso de distintivos claves de color y movimiento**

Los distintivos son fuentes de caracterización de las manos que permitan resolver los tres problemas generales del seguimiento mencionados, y se basan en dos primitivos básicos: el uso de información de color (*color cues*) y de movimiento (*motion cues*) [44, 11]. La manera cómo se usan ambos *cues* varía en cada algoritmo y son los responsables de generar las hipótesis o posibles posiciones de las manos. Los distintivos de color se analizan sobre el cuadro actual, pero los de movimiento implican también un historial de cuadros.

El uso del color permite delimitar las regiones donde exista un color similar a la de la piel y por tanto una alta probabilidad de encontrar las manos. Una colección de enfoques distintos sobre el uso de este distintivo se encuentra en la recopilación [52]. Por ejemplo, se tiene el uso de modelos y métricas estadísticas del color [53, 54], el uso de diferentes representaciones cromáticas según lo requiera la aplicación [55], la integración comparativa de distintas segmentaciones basadas en niveles de color [56], etc.

Por otro lado, el movimiento se refiere a los cambios de valores que pueden presentar los píxeles pertenecientes a algún objeto o región de interés de una imagen a lo largo de los cuadros del video. El análisis del movimiento es útil si se consideran a las manos como los objetos más movibles dentro del cuadro. Por ejemplo, se tiene a la diferenciación de imágenes por alguna función o métrica, el uso de *optical flow* [57], el análisis de cambios en las degradaciones de color que sugieran un movimiento [10], etc.

- **Solución mediante el conocimiento *a priori*, temporal y de interactividad de las manos**

El uso de los distintivos de color y movimiento es mandatorio para el seguimiento de las manos; sin embargo, se pueden usar otras informaciones adicionales para mejorarla :

1. El conocimiento *a priori* de un entorno poco dinámico que rodea a la mano puede usarse para buscar características de elementos estáticos que sirvan como puntos de referencia para caracterizar al entorno y así resaltar a los elementos más movibles, que son las manos [5].

2. La información de la posición de la mano en el cuadro anterior puede servir para limitar su búsqueda dentro del cuadro actual, resultando en una aceleraración en la detección de la mano. Sin embargo, la desventaja es que un error de localización en el cuadro actual puede convertirse en un problema de difícil recuperación porque el seguimiento posterior se basa en información errada. Una solución es considerar un mayor historial o intervalo de cuadros pasados para definir la detección del cuadro actual: si ocurre un error entre el cuadro anterior y el actual, se posee información previa que pueda corregirla [44, 50].

3. Se pueden modelar las interacciones entre ambas manos para así usar la sincronización e interdependencia existentes entre sus movimientos y mejorar la tolerancia frente a oclusiones mutuas [48].

4. Para mejorar la exactitud del seguimiento temporal de las manos, se puede emplear un entorno de actualización de predicciones. Ejemplos típicos encontrados en la literatura son el filtro de Kalman y el filtro de partículas [10]. El uso de estos métodos puede mejorar las consistencias temporales de las posiciones de las manos detectadas e inclusive introducir robustez frente a cambios en el ambiente o elementos distractores [46]. Un requisito importante es la modelación previa de la dinámica del movimiento de la mano.

Figura 1.3: Secuencia de ejemplo de seguimiento usando solamente distintivos de color en el trabajo [46]. Notar que existe un problema en el tercer y cuarto cuadro mostrado en la derecha, pues el rostro, al tener un color muy similar al de la mano, es detectado como posición de ella.

1.1.2.4. Ejemplos de estudio y soluciones

Se presentan seis casos de estudio y solución al seguimiento de manos, con el objetivo de entender el uso y función particular que se les da a los distintivos de movimiento y color.

Conviene solamente mencionar que varios estudios encontrados durante el desarrollo de la tesis usan métodos conocidos en la literatura del seguimiento, tales como: *Condensation* [58], *Kalman filtering* [26, 59, 60, 61, 48], *Meanshift tracking* [15, 46, 62, 26], *Camshift tracking* [63, 60], *Particle filtering* [12, 46, 47, 64, 45, 65] y *Principal Component Pursuit* [66].

- **Soluciones que emplean solamente distintivos de color**

Real Time American Sign Language Recognition Using Desk and Wearable Computer Based Video [34]

Se presenta un *barehand tracking* con la finalidad de servir para una posterior aplicación de reconocimiento del lenguaje de señas. La mano es el elemento de mayor tamaño en el cuadro, y el usuario debe portar un casco con la cámara de video.

Durante una etapa de inicialización, la posición de cada mano es conocida y en cada una se extrae el color del píxel asociado. Este píxel es etiquetado como perteneciente a la mano, y se identifica en sus 8 vecinos cuales tienen un color similar al de él, para luego también etiquetados. Este proceso se realiza iterativamente por cada píxel etiquetado hasta llegar a formar una región al cual se le aplica una dilación morfológica para obtener un *blob* por cada mano. Se calcula el centroide del *blob*, y este valor representa la posición de la mano en el cuadro actual, y servirá también para iniciar nuevamente una etiquetación en el siguiente cuadro. La detección de oclusión es inmediata: el *blob* formado durante una oclusión en el cuadro actual es mayor que el obtenido por cualquier *blob* de cada mano en cuadros anteriores. Terminada la oclusión, la posición de cada mano se obtiene mediante el uso de un historial de los centroides en cuadros anteriores.

Hand Tracking by Binary Quadratic Programming and Its Application to Retail Activity Recognition [48]

Se pretende un seguimiento de ambas manos enfocado a videos de baja resolución para aplicaciones de vigilancia en puntos de venta dentro de supermercados. El seguimiento se resuelve mediante un problema de optimización denominado Binary Quadratic Programming (BQP), cuya función costo modela tres situacio-

nes. La primera es la dificultad de que una posible ubicación de la mano pertenezca a la posición verdadera en función de su posición actual y un historial de posiciones anteriores. La segunda es modelar la dinámica del movimiento de la mano. Y la tercera es modelar las interacciones los movimientos de ambas manos para resolver casos de oclusión mutua y pérdidas temporales de las manos fuera de la visión de la cámara.

Se requiere una etapa de entrenamiento previo *offline* basado en diferentes imágenes de manos para modelar el color RGB de la mano de acuerdo a una distribución gaussiana, el cual luego se adapta continuamente *online* según los cambios de iluminación en la piel de las manos del usuario. Por cada píxel dentro del cuadro se genera una probabilidad de tener color de piel al computar la distancia Mahalanobis [11] con respecto al modelo de color de piel, para luego segmentar aquellos pẋeles que tengan mayor probabilidad de pertenecer a la mano. Estos resultados sirven para corroborar los estimados iniciales de la posición de cada mano que son obtenidos mediante un filtro de Kalman [67], lo cual se realiza dentro del problema de optimización descrito incialmente.

- **Soluciones que emplean solamente distintivos de movimiento**

Hybrid Feature Tracking and User Interaction for Markerless Augmented Reality [5]

Se presenta un *barehand tracking* en donde la mano es el elemento de mayor tamaño; destinado a aplicaciones de HCI para un sistema de Realidad Aumentada en la cual la mano ubique a los elementos virtuales por medio de su movimiento.

Luego de ser ubicada durante la inicialización, la mano es seguida mediante una combinación de características distintivas estáticas, que corresponden al entorno de la mano, y dinámicas, que corresponden a la misma mano. Las características dinámicas son distintivos ubicados por *optical flow* [10]. El seguimiento de distintivos estáticos es posible gracias al conocimiento de que el contexto que rodea a las manos no es cambiante, por lo cual se usan características SIFT (*Scale invariant feature transform*) [10]. Las características SIFT se usan para corregir aquellas características obtenidas por *optical flow* que hayan sufrido algún *drift* o que falsamente aparecieron por cambios de iluminación aparente en el cuadro. Por último, la velocidad de detección de cada característica es dispareja y para evitar que la detección de una retrase a la otra, estas se realizan sobre distintos hilos sincronizados.

Automatic 2D Hand Tracking in Video Sequences [7]

Se presenta un *barehand tracking* de ambas manos. No se utilizan características de color para evitar problemas resultantes por cambios de luminosidad en el entorno.

El distintivo de movimiento utilizado es llamado *motion residue*. Este se aplica sobre dos cuadros consecutivos, y lo que hace es dividir al primero en secciones de bloques, donde a cada uno se le trata de encontrar su mejor pareja (*match*) en el siguiente cuadro por simple traslación, dando como resultado a una imagen de flujo de bloques que estima el movimiento de cada bloque en el siguiente cuadro. Luego, se calcula el promedio de la diferencia absoluta entre el nivel de intensidad de cada bloque y su pareja encontrada en el siguiente cuadro. Los bloques con mayor valor asociado son los identificados como pertenecientes a regiones de las manos. Concluido todo este proceso, se tiene identificado a un conjunto de candidatos de las manos en el cuadro actual. Luego, el algoritmo selecciona el candidato que mejor se aproxime a la posible ubicación de las manos mediante la aplicación de un filtrado temporal que remueve a los falsos candidatos

restantes dentro del cuadro. Finalmente, si ocurre un problema de oclusión mutua, el seguimiento se diseña de modo que pueda detectarlo y reinicie la búsqueda de candidatos en el siguiente cuadro hasta encontrar un cuadro en donde no ocurra dicho problema.

- **Soluciones que emplean distintivos de color y movimiento**

Real-time hand tracking using a mean shift embedded particle filter [34]

Se plantea el *barehand tracking* de una mano mediante el uso principal del color de la piel como característica distintiva de la mano por ser invariante a escala y rotación. La imagen de cada cuadro es transformado en un mapa de probabilidad de poder encontrar en cada píxel el color de la piel, y se segmentan las regiones con mayor probabilidad. La referencia del color de la piel es actualizada durante el seguimiento para evitar problemas derivados por cambios de iluminación.

Por otro lado, con el objetivo de aumentar la robustez del seguimiento, también se usan distintivos de movimiento para diferenciar las regiones en movimiento de la mano de otras posibles regiones estáticas de color similar a la de la piel dentro del entorno. Por cada píxel se calcula una suma local de las diferencias absolutas entre el cuadro actual y el anterior, para luego segmentar los píxeles con mayor valor y definirlas como regiones en movimiento. La nueva imagen resultante sirve como máscara para el mapa de probabilidad de color y así formar una imagen que indica qué regiones están en movimiento y a la vez tienen un color similar a la de la piel. Estas regiones definen un valor de bondad que se asignan a un número de candidatos manejados por un filtro de partículas, y que por medio del método *Meanshift* [10], permiten obtener la mejor posición para la mano.

Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration [57]

El trabajo realiza un *barehand tracking* de una sola mano. Se introduce el concepto de *Flocks of Features*, el cual consiste en mantener un número fijo de pequeas regiones de imágenes o características cuya naturaleza y trayectorias a lo largo del video se determinan por *optical flow* (características *Kanade Lucas Tomasi* (KLT) [68]), y que constituyen el distintivo de movimiento de la mano. Cada característica está restringida a mantener una distancia mínima respecto a las otras para evitar su amontonamiento, como tampoco exceder una distancia máxima con respecto a la mediana de la posición de todas las características para evitar su alejamiento excesivo. Si una característica incumple alguna de las dos condiciones, se la posiciona al lugar con mayor probabilidad de color de piel y movimiento, a fin de evitar su posicionamiento en elementos estáticos del entorno con color similar a la piel o elementos en movimiento con color distinto a la piel. El distintivo de color es la probabilidad de color de piel obtenido de la evaluación de cada píxel en un histograma RGB normalizado del color de la piel obtenido durante la inicialización. Estas distancias mínimas y máximas de alejamiento se calibran para poder obtener un balance entre la importancia de los distintivos de color y movimiento. Este método evita la pérdida de características entre cuadros sucesivos y permite que todas siempre permanezcan sobre la mano.

En cada cuadro, la posición de la mano es representada por la mediana de las posiciones de las características KLT. Como resultado, el seguimiento es robusto frente a cambios en el entorno y/o posición de la cámara.

1.2. Síntesis sobre el Asunto de Estudio

El seguimiento de manos tiene gran importancia en aplicaciones en donde se requiera el reconocimiento de algún gesto o movimiento manual del usuario. Generalmente se lo encuentra como una etapa preliminar necesaria para posteriores procesamientos de mayor nivel (*post-processing*), que son requeridos para distintas tareas que no serían posibles de no contar primero con la posición adecuada de la mano en un determinado instante. Por ejemplo, una vez identificadas las posiciones de la manos, se pueden aplicar un *post-processing* para saber qué gesto o signo la mano está denotando, determinar a qué elementos del entorno pueda estar apuntando, si hay algún objeto o herramienta sostenida por ellas, etc.

Por otro lado, es posible observar que en distintas implementaciones, el seguimiento no siempre se realiza con el uso exclusivo de una cámara, sino también con ayuda de otros mecanismos adicionales, los cuales básicamente actúan de dos formas distintas:

- Proveer información sensorial adicional que complemente la información directa obtenida de la cámara. Este caso supone una mayor complejidad o especialización del *hardware* y por tanto un costo adicional; como por ejemplo, el uso de sensores de profundidad o el uso de reflectores infrarrojos.

- El uso de indumentaria adicional por parte del usuario que mejore y refuerce la detección de las manos, que abarcan desde el simple uso de capuchitas de colores en los dedos hasta el uso de guantes de color y textura especialmente diseñados.

Además, por lo general, para que estos mecanismos adicionales realmente sean útiles y se explote la información que proporcionan, suponen la existencia de una configuración especial del entorno que rodea a la mano. Esta restricción en el entorno puede convertir al algoritmo de seguimiento muy específico y particular a las aplicaciones para los cuales estuvo diseñado.

Por último, es importante destacar que existe una gran variedad de distintas técnicas para el procesamiento del seguimiento, y cuya heterogeneidad depende de cómo se lleva a cabo el procesamiento de los distinivos o *cues* obtenidos en los cuadros, los cuales tienen como común denominador estar dentro de las características básicas de color y/o movimiento de la mano. Sobre estas características se construyen los algoritmos más diversos en cuanto enfoque y complejidad.

Capítulo 2

Modelo teórico, hipótesis y objetivos

En la tesis se desarrolla un *barehand tracking* y un seguimiento basado en la apariencia de la mano, en donde ella no es necesariamente el elemento de mayor tamaño en el cuadro (ver sección 1.1.2.2.).

2.1. Marco poblemático

El seguimiento de manos se realiza a través de los cuadros del video y el resultado final por cada cuadro es la posición de un indicador visual (ventana rectangular indicadora) centrado en las coordenadas bidimensionales que represente a cada mano y el cual debe identificar a la región en donde se contenga la mayor cantidad de la mano en dicho cuadro [7]. El problema que se desea evitar es la pérdida de seguimiento: la pronta incapacidad temporal de continuar identificando a la mano en los cuadros sucesivos. Las manos son ubicadas e identificadas gracias al uso de distintivos que caracterizan a las manos por medio de alguna configuración o particularidad asociada. Estos distintivos son extraídos por medio de alguna métrica de comparación o procesamiento diverso sobre dos cualidades básicas comunes: el color (*color cues*) y el movimiento (*motion cues*) [26].

Una mejor extracción de los distintivos está asociada a un mejor seguimiento, y esto es posible en condiciones adecuadas de resolución e iluminación que mejoren la visibilidad de los elementos dentro del cuadro. La resolución es importante porque define la cantidad de píxeles pertenecientes a la región de la mano y por tanto también define la cantidad de información presente para caracterizarla (a mayor tamaño, existe mayor información sobre la mano). Asimismo, la iluminación importa porque define el contraste que existe entre el *foreground* (plano o región en donde se ubica la mano dentro del cuadro) del *background* o entorno (regiones o planos que rodean a la mano): si el contraste es menor, mayor es la dificultad de diferenciar ambas regiones y separar a la mano de cualquier otro elemento del entorno [10]. En la sección 1.1.2.3 se han descrito los problemas generales del seguimiento de manos; sin embargo, ahora se los presenta enfatizando su relación directa con el uso de los distintivos :

- **Alteración de las características que identifican a las manos**: La alteración o cambio repentino de alguna de las cualidades del *foreground* puede originar una mala caracterización de la mano y la pérdida de su identificación, lo cual trae como consecuencia una posible pérdida de seguimiento en el cuadro actual. Entre las cualidades del *foreground* que pueden modificarse, se encuentran:
  - La presencia de algún movimiento repentino de la mano brusco o violento.

- Cambios de escala y perspectiva de las manos frente a la cámara.
- Tonalidad variante de la piel a lo largo del video por cambios de iluminación y poca textura.
- Oclusiones entre ambas manos.
- Posiciones fuera del campo visual de la cámara.
- Deformidad y no-rigidez de la mano, que la vuelve ”amorfa” y justifica la gran cantidad de poses y ubicaciones que puede tomar a lo largo de la secuencia del video.

- **Presencia de características de las manos similares en el *background***: Genera la pérdida del seguimiento porque introducen falsos positivos originados por la pronta aparición de distintivos similares a la de la mano en alguna región ajena a ella dentro del *background*, y sobre las cuales erróneamente se pueden ubicar a las manos. De manera general, los falsos positivos ocurren con cambios de iluminación drásticos; sin embargo, de manera particular, ocurren en entornos dinámicos cuando los distintivos usados son de movimiento, o entornos recargados con diversos elementos de colores cuando los distintivos son de color.

Por otro lado, la implementaciń del seguimiento de manos implica considerar factores adicionales a los algorítmicos anteriormente descritos, porque el algoritmo de seguimiento puede estar funcionando correctamente pero sin ser factible de implementarse en la aplicación o contexto en el cual se le desea usar [5]. Por ejemplo, es importante considerar qué recursos o facilidades adicionales brinda el sistema sobre el cual se está implementando el algoritmo: recursos computacionales, funciones del sistema operativo, soporte de instrucciones especiales en el procesador, etc. Entre estos factores adicionales tenemos [69]:

- **Tiempo de procesamiento adecuado:** Dependiendo de cunto tiempo el procesamiento demore por cuadro, se tiene una equivalencia a cuantos cuadros por segundo (fps) se es posible mostrar las imgenes con las regiones de las manos identificadas. A mayor tasa de fps, más rápido es el procesamiento, se rechaza una menor cantidad de cuadros entrantes del video, y la implementación es más adecuada para aplicaciones*online* o en tiempo real. Por ejemplo, la cmara puede adquirir a 30fps, pero el procesamiento puede tomar un tiempo tal que solo pueda procesar y mostrar las manos identificadas en pantalla a 10 fps.

- **Máxima velocidad y/o desplazamiento por cuadro de la mano:** Es la velocidad *cm/s* o distancia máxima *cm* con la cual pueden moverse las manos del usuario entre dos cuadros consecutivos para poder ser correctamente seguidas.

- **Resolución:** Los algoritmos de seguimiento pueden varíar su tiempo de procesamiento en relación con las dimensiones de la resolución del cuadro de entrada: a mayor resolución, mayor es el tiempo de procesamiento por existir más datos de entrada; y viceversa. Esta relación depende del orden de complejidad de las etapas que conforman al algoritmo (logarítmico, polinomial, exponencial).

- **Ubicación con error tolerable:** La ubicación de la mano debe estar dentro de los márgenes de tolerancia de inexactitud considerados en el algoritmo. Por ejemplo, la ventana indicadora puede no calzar *exactamente* sobre el área perteneciente a la mano en todo el video, sino que al menos asegurar el cubrimiento constante de un porcentaje adecuado de esta área.

- **Inicialización sencilla e intuitiva:** Es una condición deseada, pues la inicialiación debe ser fácil de configurar para el usuario; de lo contrario, aumenta la posibilidad de que el usuario cometa un error de ubicación inicial en las manos y se dificulte el seguimiento posterior.

En conclusión, la problemática del seguimiento de manos se puede resumir en la realización de un seguimiento con un grado favorable de exactitud, velocidad y carga de procesamiento; a la vez de resolver problemas relacionados a la oclusión y pérdidas de seguimiento.

2.2. Modelo teórico

2.2.1. Esquema general del funcionamiento

La primera etapa del algoritmo es la inicialización que permite ubicar cada mano en una posición conocida inicial y también inicializar todas las variables necesarias para el algoritmo. Terminada la inicialización, se empieza el seguimiento cuadro a cuadro de las manos, el cual puede incluir una etapa de preprocesamiento que acondicione la imágen de entrada en cada cuadro para el posterior procesamiento del mismo, como puede ser la aplicación de una etapa de filtrado.

La posición de la mano en el siguiente cuadro puede tener diferentes niveles de incertidumbre que dependen de qué condiciones de movimiento se espera de ella y que también definen las posibles posiciones o hipótesis de su ubicación. Por ejemplo, se puede esperar que la mano se desplace como máximo a una distancia radial respecto a la posición ubicada en el cuadro anterior. Otro tipo de hipótesis es suponer que solo las regiones con una probabilidad mayor al 0.7 de tener un color similar a la de la mano, sea posiblemente una de ellas. La extracción de distintivos para la generación de hipótesis viene dada por uno o más mecanismos cuantitativos. Ejemplos de estos mecanismos cuantitativos son: la evaluación sobre alguna función de similitud o métrica, la obtención de la probabilidad de hallar una característica particular, segmentaciones, etc. Los mecanismos permiten interpretar los distintivos para ubicar las regiones con mayor probabilidad de pertenecer a una posible posición de la mano. Por ejemplo, un distintivo de color es el generar un modelo del color de la mano basado en una mezcla de gaussianas, y el mecanismo para usarlo es la simple evaluación de un píxel RGB sobre el modelo, la obtención del valor de probabilidad de pertenencia al color, y luego aplicarle un valor umbral que determina si es válido o inválido de pertenecer a la mano. En la sección 1.1.2.3se aprecia que los mecanismos cuantitativos pueden aplicarse también sobre otros recursos como algún historial variable de cuadros pasados o conocimiento explícito del *background* que brinde información que mejore el seguimiento.

La obtención final de la posición de las manos resulta de la aplicación de un mecanismo de discriminación sobre las hipótesis (resultantes de los mecanismos cuantitativos), el cual realiza un discernimiento sobre la posición más óptima para ubicar la mano dentro de todas las hipótesis. Esta es la parte del algoritmo de mayor ”inteligencia”, pues puede incluir modelos dinámicos de la mano y técnicas predictivas, la formulación de problemas de optimización, manejo de oclusiones, algoritmos para la recuperación del seguimiento luego de su pérdida, el aprendizaje de trayectorias anteriores de las manos, etc.

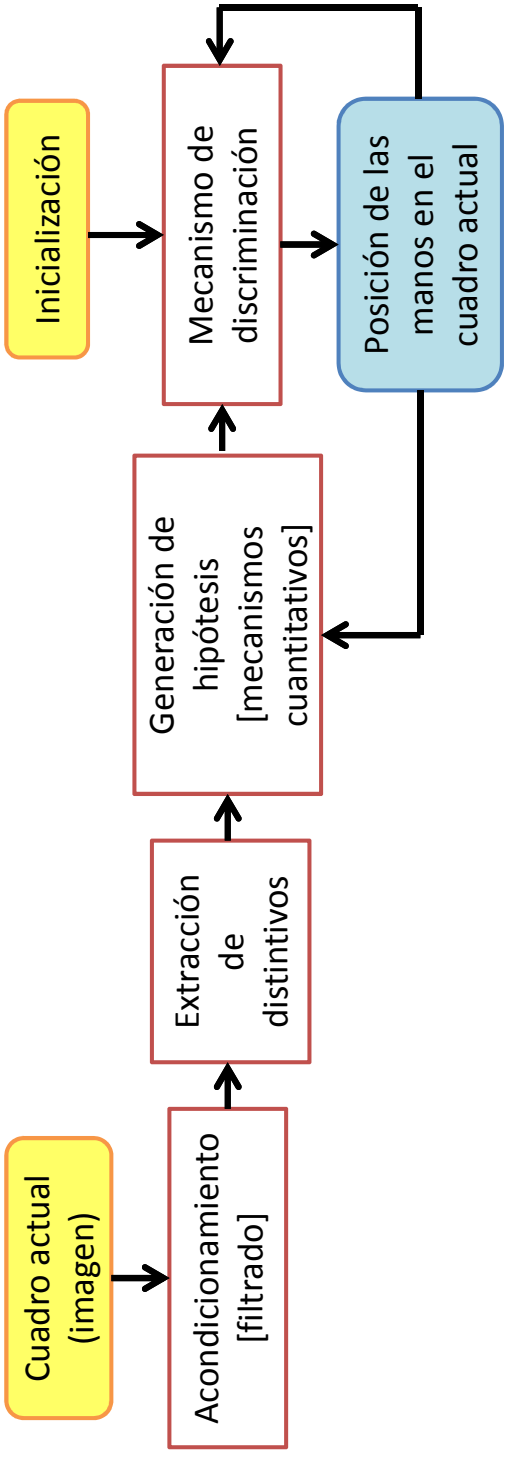


Figura 2.1: Diagrama de bloques específico.

2.2.2. Diagrama de bloques específico

El diagrama 2.1 compacta el esquema general de funcionamiento y permite articular mejor la funcionalidad a cada etapa. Los recuadros en amarillo representan una entrada para el algoritmo, el de celeste la salida y el resto son etapas de procesamiento.

Por otro lado, el diagrama 2.2 permite observar el esquema de funcionamiento del seguimiento de manos de manera muy resumida y en relación con tres problemas o requerimientos importantes en su implementación mencionados anteriormente en el marco problemático.

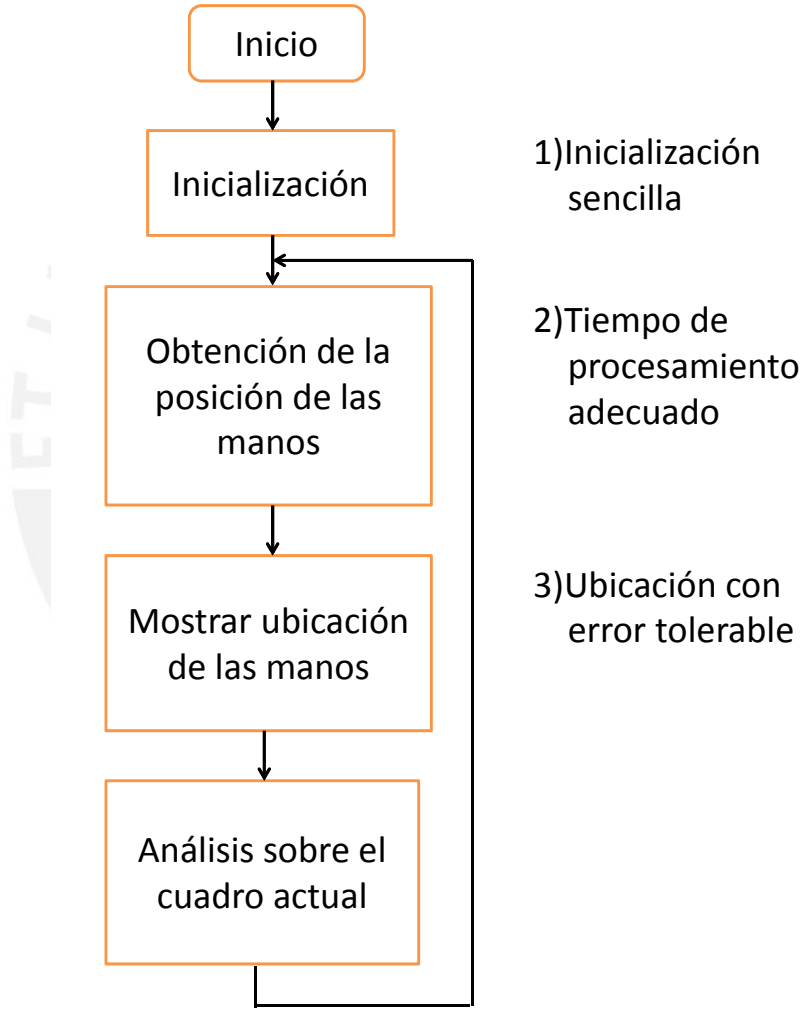


Figura 2.2: Esquema de funcionamiento y requerimientos de implementación.

2.3. Hipótesis

2.3.1. Hipótesis principal

Desarrollar un algoritmo de seguimiento de manos práctico, que requiera menor complejidad computacional, y por tanto que sea más rápida su ejecución sobre computadores estándares en el mercado y también mini-computadores, pero que que a la vez mantenga una robustez y exactitud tolerable.

2.3.2. Hipótesis secundatias

- Se podrá implementar un algoritmo *cache aware* de seguimiento de manos [70], de modo que se acelere su procesamiento y sea potencialmente adecuado para el uso de instrucciones SIMD u otras mejoras que le den una aceleracón aún mayor para alguna implementación futura.
- La implementación del algoritmo en un computador personal de características estándares en el mercado será apto para aplicaciones en tiempo real que superen los 10 fps; asimismo, se espera que también lo sea para el mini-computador MK802.
- El algoritmo desarrollado tendrá un grado de robustez adecuado frente a cambios de iluminación.

2.4. Objetivos

2.4.1. Objetivo principal

Implementar un algoritmo de seguimiento de manos en una computadora personal de características estándares en el mercado y la mini-computadora MK802, tolerante a cambios de iluminación en el ambiente de la detección, con un error aceptable de detección y eficiente en cuanto al uso de recursos de memoria y procesamiento.

2.4.2. Objetivos secundarios

- En [34] se menciona que 10 fps es lo necesario para el reconocimiento humano visible de señas, por lo tanto, se plantea en esta tesis tener dicha tasa como cota inferior a alcanzar y superar de ser posible para tener una aplicación en tiempo real.
- Realizar un algoritmo *cache aware* [70] que potencialmente podrá hacer uso de extensiones tales como SIMD entre otras para acelerar su procesamiento.
- Realizar un algoritmo capaz de resolver situaciones de oclusión y recuperarse continuamente de pérdidas de seguimiento.
- Hacer un uso exclusivo de los distintivos de movimiento en el video, por ser más robusto frente a cambios de iluminación general y aprovechar las cualidades de movimiento no rígido de las manos.

Capítulo 3

Desarrollo de la solución: Procesamiento

3.1. Consideraciones y concepción del seguimiento de manos propuesto

El algoritmo desarrollado realiza un seguimiento de ambas manos del usuario por simple visión monocular, en el cual el usuario no necesita usar indumentaria adicional y sus manos no necesariamente deben ser los elementos de mayor tamaño o protagonismo en el cuadro (ver sección 1.1.2.2.). Cada mano es seguida de manera independiente, a excepción cuando entra en condición de oclusión.

A continuación se definen cuáles son las condiciones requeridas para la correcta aplicación del algoritmo desarrollado, para luego definir su concepción de solución e implementación.

3.1.1. Condiciones y restricciones

Es muy importante definir cuáles son las supuestas condiciones del contexto y las restricciones de la aplicación del seguimiento porque estas influyen directamente en el diseño del algoritmo y su grado de complejidad operacional: dependiendo del grado de restricción, se puede alterar la complejidad de su procesamiento. Estas son las condiciones o supuestos considerados para el seguimiento [46, 2, 7, 65, 60, 61, 47, 45, 12, 64]:

1. El usuario mueve las manos aproximadamente sobre un conjunto cercano de planos paralelos frente a la cámara, de modo que no existan problemas de perspectiva o cambios aparentes del tamaño de la mano captada por la cámara.

2. La cámara permanece inmóvil en toda la secuencia de video.

3. El usuario procura no realizar movimientos inválidos que coloquen a sus manos fuera del campo de visión de la cámara, es decir, que ambas manos siempre deben estar visibles.

4. Solo existe una persona frente a la cámara.

5. Las manos son los elementos en constante y con mayor movimiento dentro del cuadro: no existen otros elementos distractores en el entorno que compitan en movimiento con las manos. Sin embargo, se considera inevitable el movimiento de codos, brazos y la cabeza del usuario.

6. El movimiento de las manos es natural, flexible y evita ser violento, brusco o raudo.

Estas seis condiciones o restricciones se cumplen naturalmente en distintas situaciones dentro de las actividades humanas; por ejemplo, cuando un ponente está exponiendo algún tema frente a una audiencia por medio de movimientos manuales naturales [7]. Por otro lado, estas seis condiciones son comunes y adecuadas en aplicaciones de *Human-computer interaction* que tengan a las manos como los elementos principales de comunicación con el computador, y por tanto, se espera de ellos un mayor movimiento protagónico dentro del video [35].

Estas seis restricciones deben tenerse presente durante todo del desarrollo de la tesis, y es en consideración a ellas que se ha optado por el desarrollo de un algoritmo que haga uso exclusivo de distintivos de movimiento (*motion cues*) para la caracterización de las manos. Esta particularidad también permite alcanzar los objetivos de un seguimiento con mayor independencia del color de la mano y robustez frente a cambios de iluminación general.

3.1.2. Concepción del algoritmo

Además del desarrollo de un algoritmo que cumpla con las seis restricciones anteriores y que realice un seguimiento con una exactitud y tasa de fps adecuadas, se conciben las siguientes cualidades:

1. **Parametrizable**: El algoritmo recibe tres parámetros como entrada: la resolución actual de los cuadros del video, la posición inicial de las manos y el tamaño de las manos proporcional a dicha resolución (en pixeles). Por lo tanto, el seguimiento se adapta a distintas resoluciones de video y se puede ejecutar sobre distintos entornos *hardware*. A menor resolución, mayor velocidad tiene el procesamiento y es más adecuado para aplicaciones *online*.

2. **Sin necesidad de entrenamiento previo**: No se requiere una etapa de aprendizaje o entrenamiento previo para modelar alguna característica o distintivo presente en la mano o en su entorno.

3. **Sin necesidad de alguna calibración previa**: No es necesario calibrar algún parámetro, constante de proporcionalidad o valor umbral para mejorar el seguimiento. Todo parámetro se redefine internamente dependiendo de los tres parámetros de entrada.

4. **Modularidad y posibilidad de inclusión de nuevas soluciones**: El algoritmo desarrollado tiene una implementación modular, claramente diferenciada en etapas en las cuales se usan distintos métodos cuantitativos y distintivos. Esto facilita el mantenimiento y la modificación del código, pero también la inclusión de nuevos métodos y distintivos dentro de los módulos.

5. **Manejo de oclusión**: Se considera la solución a posibles casos momentáneos o prolongados de oclusión mutua o juntura en las manos.

6. **Solución a pérdidas de seguimiento**: El algoritmo es capaz de volver a recuperar el correcto seguimiento luego de una pérdida del mismo, por lo cual este problema se convierte en una simple ocurrencia temporal recuperable.

3.2. Extracción y procesamiento de distintivos

La extracción y procesamiento de distintivos es el proceso por el cual se caracterizan a las manos de acuerdo a sus cualidades de movimiento con el objetivo de poder utilizarlas en distintos mecanismos cuantitativos que permitan encontrar las ubicaciones de las manos dentro del cuadro. La premisa para emplear distintivos de movimiento o *motion cues* es que las manos son los elementos con mayor *cantidad de movimiento* en el cuadro. El concepto de cantidad de movimiento se refiere a un valor, mesurado por medio de algún mecanismo cuantitativo, que indique de manera proporcional o semántica si es que hubo algún elemento en movimiento dentro de alguna región particular entre el cuadro anterior y el actual. Entonces, se postula que aquellas regiones del cuadro con mayor cantidad de movimiento corresponden a las regiones donde posiblemente se ubiquen las manos en el cuadro actual. En el algoritmo se desarrollan dos clases de distintivos: el *Mapa global de movimiento* y el *Mapa local de similitud*.

Los distintivos de movimiento empleados requieren la división de la imagen en regiones menores consistentes en bloques cuadrados de dimensión de lado fija llamados *bloques estándares*. El lado de un bloque estándar es proporcional a las dimensiones de la mano y el desplazamiento máximo deseado de la mano entre dos cuadros sucesivos, y esta proporcionalidad se define por una *constante de proporcionalidad de región*.

Los distintivos de movimiento se basan en mecanismos de diferenciación entre el cuadro actual y el anterior en escala de grises, porque lo que interesa como indicio de movimiento son los fuertes cambios de valores de intensidad de los píxeles dentro de una región en particular entre dos cuadros sucesivos. Como consecuencia, si ocurren cambios de *iluminación global* en todo el cuadro actual, debido a que la diferencia de intensidad está en misma proporción para todo píxel, esta no es interpretada como un cambio de movimiento ocurrente en alguna región particular. Esto significa que el algoritmo es robusto frente a este tipo de cambios, lo cual no es posible si se dependiese exclusivamente de distintivos de color.

A continuación se describen los dos procesos importantes dentro de la extracción y procesamiento de características: la etapa de filtrado o acondicionamiento, y el análisis de distintivos. Como anotación, el uso del símbolo $ROWS$ se refiere al ancho de la imagen del cuadro en pixeles, y $COLS$ se refiere al largo.

3.2.1. Filtrado y acondicionamiento

El primer problema que debe resolverse para la extracción de distintivos es disminuir el efecto negativo de dos fenómenos que dificultan la extracción de información de movimiento en el cuadro:

1. El nivel de ruido y sus asociados cambios de intensidad local dispersos dentro del cuadro.

2. La existencia de elementos de gran textura o de considerable variación de frecuencia local, como puede ser la ropa del usuario. Estas elementos tienen el efecto de poder incrementar la cantidad de movimiento extraído por las características de manera considerable al mínimo movimiento, pues generan altos cambios de intensidad local.

La solución para los efectos negativos de ambos fenómenos es la implementación de una etapa de preprocesamiento de filtrado y acondicionamiento. Esta etapa consiste en convertir el cuadro actual en escala de grises y aplicarle un filtro pasa-bajos con el efecto de reducir los niveles presentes de ruido y textura: reducir el importe de las altas frecuencias presentes en la imagen.

La descripción formal de esta etapa es:

1) Entradas:

$(\mathrm{I}_t)_{RGB}$: Cuadro actual a color RGB

2) Salidas:

I_t: Cuadro actual filtrado

3) Proceso:

$$\mathrm{I}_t^{'} = \mathrm{RGBtoGRAY}((\mathrm{I}_t)_{RGB}) \tag{3.1}$$

$$\mathrm{I}_t = \mathrm{I}_t^{'} * \mathrm{h}, \qquad \mathrm{h} = 1_{11x11} \tag{3.2}$$

4) Costo computacional:

$$O(2\ ROWS\ COLS) \tag{3.3}$$

Existen dos ahorros de costo computacional dentro de la implementación de esta etapa. El primero es el almacenar el cuadro filtrado en un búfer, debido a que será usado también en el procesamiento del siguiente cuadro entrante y así evitar realizar el acondicionamiento nuevamente.

El segundo ahorro computacional es en la implementación de la convolución bidimensional empleada. Sucede que para el empleo de un *kernel* cuadrado de k píxeles por lado, el orden de la convolución es $O(ROWS\ COLS\ k^2)$ si es un filtro no separable, y $O(4\ ROWS\ COLS\ k)$ si es separable. Ahora, en la presente etapa se usa un filtro promedio separable k = 11, que en teoría tiene un costo computacional por píxel: $\frac{O(4\ ROWS\ COLS\ k)}{ROWS\ COLS} = O(4\ k)$. Sin embargo, esta etapa ha sido implementada de un modo eficiente tal que el orden se reduce a $O(2ROWSCOLS)$,y el costo computacional por píxel a $O(2)$. Este es un orden constante: solo se requieren dos operaciones por píxel independientemente del tamaño del *kernel*, y por tanto el *speed-up* alcanzado tiene un factor de *2 k*.

3.2.2. Análisis de distintivos

Las ventajas del uso exclusivo de distintivos de movimiento respecto a los de color en las restricciones de la presente tesis son:

1. Aprovecha las cualidades de movimiento no rígido y deformaciones continuas de las manos para caracterizarlas mejor. Debido a que los distintivos empleados no dependen de la forma, ni estrictamente del color de la mano; la mano es libre de cambiar su apariencia, siempre y cuando se mantenga un contraste suficiente de iluminación con su entorno.

2. Permite una inicialización sencilla en la cual el usuario debe agitar sus manos constantemente durante un tiempo hasta que el sistema las reconozca. No se requiere ninguna configuración o entrenamiento previo para la extracción del color de las manos del usuario, ni de las características de iluminación global del entorno. Por otro lado, un mayor rango de usuarios son aptos para usar el sistema porque existe una mayor independencia al color de la piel.

3. Permite una mayor tolerancia a cambios iluminación global en todo el cuadro.

4. Los mismos ditintivos empleados permiten simultáneamente resolver problemas de oclusión mutuas entre las manos y la recuperación continua de pérdidas de seguimiento.

5. Permite un mejor seguimiento dentro de ambientes estáticos recargados con elementos que pueden tener colores similares a la de piel, siempre que exista un contraste relativo de intensidad entre estos elementos y las manos.

Los distintivos de movimiento tienen dos clasificaciones según usen o no información de la posición de la mano en el cuadro anterior, que a la vez está muy relacionado a las dimensiones de las regiones de trabajo que se deben analizar. La primera clasificación es el *Análisis sin información a priori*, el cual requiere como entrada un análisis sobre todo el cuadro actual y anterior para extraer la información de movimiento. El distintivo asociado a este análisis es el Mapa global de movimiento, el cual pretende obtener un valor que identifique a las regiones dentro de todo el cuadro con mayor cantidad de movimiento en donde posiblemente se encuentre la mano. Este puede ser de dos tipos: el *Mapa global de movimiento individual* y el *Mapa global de movimiento en bloques*.

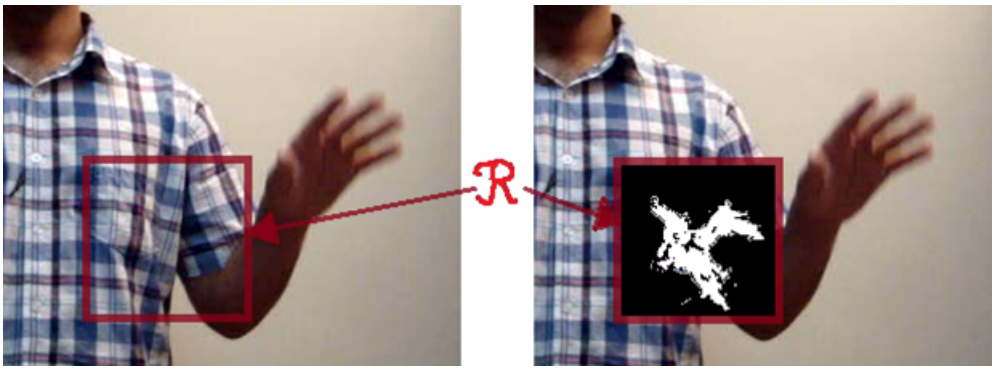
La segunda clasificación es el *Análisis de movimiento con información a priori*, el cual requiere como entrada el análisis sobre un vecindario local alrededor de la posición de la mano en el cuadro anterior dentro del cuadro actual para la extracción de movimiento. El distintivo asociado a este análisis es el Mapa local de similitud, el cual pretende extraer un valor que identifica la región donde posiblemente se ha desplazado la mano con respecto a su ubicación en el cuadro anterior.

El Mapa global de movimiento y el Mapa local de similitud se basan en la asignación de cantidad de movimiento en todo el cuadro o parte de ella mediante la división de las regiones analizadas en secciones de bloques (llamados anteriormente bloques estándares).Estas ideas son comúnmente encontradas en la literatura de Visión por Computadora [10], y son usadas en aplicaciones de compresión y reconstrucción de video [71, 72, 73]. En el contexto del seguimiento de manos, estas ideas están basadas en los trabajos de [7, 26].

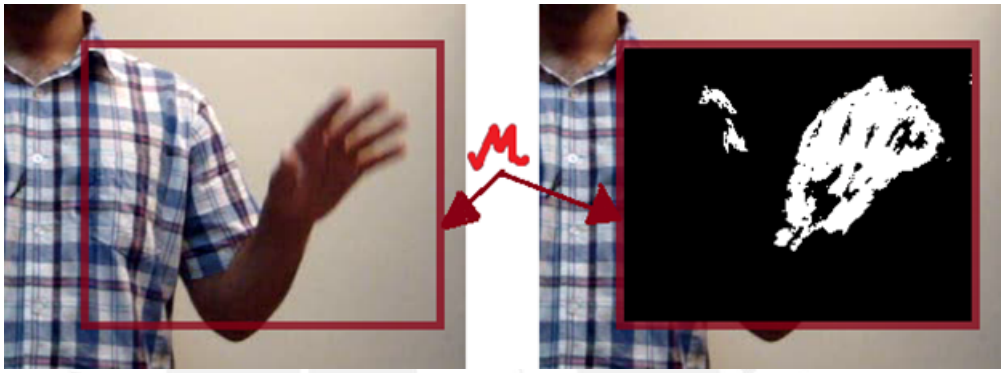
3.2.2.1. Análisis de movimiento sin información a priori: Mapa global de movimiento

Este análisis se basa en la premisa de que las manos son los elementos con mayor movimiento dentro del área de visión de la cámara de video captada en el cuadro actual, y esto puede ser suficiente para ubicar a las manos sin la necesidad de conocer donde se ubicaron anteriormente.

El análisis de movimiento se realiza sobre todo el cuadro porque, al no existir información *a priori* de la posición de la mano, esta podría estar en *cualquier* parte del cuadro. Otra razón es el hecho de que la integración de áreas más grandes en el análisis dentro del Mapa global de movimiento permite una mayor robustez y eliminación de movimientos menores presentes debido a que las regiones de las manos, al tener mayor cantidad de movimiento en todo el cuadro, tienen una *ganancia de protagonismo* frente a estos. El fenómeno de ganancia de protagonismo se encuentra explicado en la figura 3.1. El Mapa global de movimiento individual es el distintivo que segmenta los píxeles de las regiones con mayor cantidad de movimiento presente dentro de todo el cuadro. Conviene resaltar que la etapa de filtrado anterior no *elimina* las cantidades de movimiento presentes en las zonas de alta textura, sino que *reduce* sus efectos en la *cuantificación* que se realice sobre él en los distintivos.



(a) Región de análisis de cantidad de movimiento ℜ.



(b) Región de análisis de cantidad de movimiento *M*.

Figura 3.1: En estas figuras se representa el fenómeno de ganancia de protagonismo. Cada región de análisis tiene segmentados los píxeles de mayor cantidad de movimiento. Se observa que cuando se analiza la región *M*, la mano cobra mayor importancia de movimiento sobre otros elementos que aparecían tenerlo en ℜ.

En conclusión, la integración de todo el cuadro permite un análisis en el cual solo importan aquellas regiones con mayor protagonismo de movimiento, que en teoría deberían ser solamente las manos. Sin embargo, un problema que ocurre es que junto al movimiento de las manos también está el movimiento de otras partes del cuerpo, como los brazos, codos, hombros, el cuello, la cabeza, etc. Todas estas partes también aportan una cantidad de movimiento en el análisis, y en ocasiones particulares pueden equipararse al aportado por las mismas manos; lo cual se traduce en la aparición de *falsos positivos*, es decir, de regiones que erradamente podrían considerarse como pertenecientes a la manos. Para reducir estos falsos positivos, se formula el Mapa global de movimiento en bloques.

La *cadena de caracterización* es el proceso por el cual se extrae el distintivo del Mapa global de movimiento individual (MGMI), y consiste en tres etapas consecutivas: *Diferencia Absoluta temporal-local*, *Varianza local* y *Segmentación unimodal natural*. A continuación, se describen estas etapas con mayor detalle. Cuando se referencie *dim(I)*, se refiere a las dimensiones del la imagen *I* del cuadro actual.

1. Diferencia absoluta temporal-local

La descripción formal de esta etapa es:

1) Entradas:

I_t: Cuadro actual filtrado resultante de la etapa de filtrado y acondicionamiento.

I_{t-1}: Cuadro anterior filtrado resultante de la etapa de filtrado y acondicionamiento (extraído del búfer).

2) Salidas:

A: Imagen resultante de la aplicación de la Diferencia Absoluta temporal-local.

3) Proceso:

$$A = |I_t - I_{t-1}| * h, \qquad h = 1_{7x7} \tag{3.4}$$

4) Costo computacional:

$$O(4\ ROWS\ COLS) \tag{3.5}$$

En primer lugar, se calcula la diferencia absoluta píxel por píxel entre el cuadro filtrado actual y el anterior, para así generar una imagen de diferencias absolutas [34]. Luego, por cada píxel (i,j) de la imagen de diferencias absolutas, se realiza una sumatoria sobre un vecindario local de dimensiones de 7 x 7 píxeles centrada en (i,j), y el resultado es almacenado en dicha posición dentro de la imagen resultante *A*.

Todo valor no nulo dentro de la imagen de diferencias absolutas indica una variación de *intensidad* en el tiempo comprendido entre dos cuadros consecutivos (como la variación es relativa, se aplica la función valor absoluto). A mayor valor de diferencia, mayor probabilidad de que la variación haya sido producida por el movimiento de algún elemento en dicha región, pues existen cambios resultantes del ruido y cambios de iluminación. Sin embargo, con el propósito de atenuar los efectos del ruido remanente, se aplica nuevamente un filtro promedio (sumador). Por otro lado, el filtrado también permite esparcir los valores de variación de intensidad que se encuentren acumulados en pequeñas regiones del cuadro, es decir, permite un esparcimiento de la información *concentrada de movimiento* hacia sus regiones vecinas, lo cual es aprovechado en las siguientes etapas consecutivas.

2. Varianza local

La descripción formal de esta etapa es:

1) Entradas:

A: Imagen resultante de la etapa anterior.

2) Salidas:

B: Imagen resultante de aplicar la etapa de Varianza local.

3) Proceso:

$$\mu = \frac{1}{|s|} \sum_{k,l \in S} A_{kl}, \qquad S = N^{sym}_{c(i,j)}, dim(S) = 5x5 \tag{3.6}$$

$$B_{ij} = \frac{1}{|s| - 1} \sum_{k,l \in S} (A_{kl} - \mu)^2,\ dim(B) = dim(A) \tag{3.7}$$

4) Costo computacional:

$$O(75\ ROWS\ COLS) \tag{3.8}$$

Sobre la imagen resultante de la etapa anterior, se aplica un proceso de extracción de la varianza local sobre una ventana de análisis de 5x5 píxeles. La varianza local tiene la propiedad de generar un elevado valor numérico en la posición (i,j) de *B* si esta corresponde a un píxel de *A* cuyos píxeles a su alrededor presenten una gran variedad de cambios de valor, es decir, que pertenezcan a una región cercana a las fronteras donde ocurrió algún movimiento en el cuadro. De lo contrario, se genera un valor bajo si los píxeles que rodean a (i,j) mantienen un bajo o nulo cambio de valor entre ellos. En conclusión, la varianza local presenta un valor alto en aquellos píxeles que se encuentren sobre alguna región que presente una situación de frontera (variabilidad o gradiente considerable) entre zonas de alta y baja cantidad de movimiento, mejorando la delimitación entre ellas.

3. Segmentación unimodal natural

La descripción formal de esta etapa es:

1) Entradas:

B: Imagen resultante de la etapa anterior.

2) Salidas:

$\mathrm{I}_{\mathrm{MGMI}}$: El Mapa global de movimiento individual.

3) Proceso:

$$\text{tr'} = umbral\_natural(\mathrm{B}) \tag{3.9}$$

$$\text{tr} = (max(\mathrm{B}) - \text{tr'}) * 0{,}1 + \text{tr'} \tag{3.10}$$

$$(\mathrm{I}_{MGMI})_{ij} = \begin{cases} 1, & B_{ij} \geq tr \\ 0, & \text{en otro caso} \end{cases}, \qquad dim(\mathrm{I}_{MGMI}) = dim(\mathrm{B}) \tag{3.11}$$

Esta es la última etapa dentro de la cadena de caracterización y culmina con la generación del Mapa global de movimiento individual o MGMI. El MGMI es el resultado de aplicar una segmentación binaria sobre *B*, en la cual el valor de "1" en el píxel (i,j) significa que este píxel pertenece a una región de cantidad de movimiento considerable, y el valor "0" lo contrario. Experimentalmente se encuentra que *B* contiene mayoritariamente elementos nulos o de bajo valor numérico, debido a que existen pocas regiones con movimiento considerable dentro del cuadro. De este modo, se encuentra experimentalmente que un histograma de la distribución de valores de *B* es unimodal, con la moda en el extremo de menores valores de cantidad de movimiento. Esta condición es ideal para aplicar la segmentación unimodal descrita en [74], cuya ventaja reside en el uso de un valor umbral generado de manera particular para cada cuadro del video (a diferencia de los umbrales *hard-coded*, el cual es denominado *umbral natural*. Este umbral natural es aumentado en un valor de 10 % respecto a la diferencia entre este valor y el máximo valor encontrado en *B*, para luego proceder a la segmentación respectiva.

Finalmente, conviene realizar un análisis sobre la importancia de cada etapa dentro de la cadena de caracterización sobre el MGMI: se analiza qué sucedería si se omite ciertos procesos dentro de ella. Este análisis puede ser apreciado gráficamente en la figura 3.2. En primer lugar, si se omite la etapa de Varianza local, sucede que dentro del MGMI se forman *clusters* relativamente gruesos y toscos de píxeles etiquetados con "1", debido a que la etapa de Varianza local se encargaba de limitar las regiones en movimiento. Un primer problema que puede ocurrir es que los *clusters* pueden formarse en zonas como codos o brazos, y ser lo suficientemente grandes como para fusionarse con el *cluster* de la mano y formar uno solo: la mano sería indistinguible del brazo, y su ubicación podría ser errónea. Un segundo problema puede ser que los mismos *clusters* identificados para cada mano lleguen a ser tan grandes de modo que resulten juntándose y fusionándose en uno solo, a pesar de que no exista oclusión observable en el cuadro actual (lo ideal es tener ambos clusters separados para que cada una identifique a una mano independiente de la otra).

Si se omite el filtrado posterior al cálculo de la imagen de diferencias absolutas dentro de la etapa de Diferencia absoluta temporal-local, se fomenta la acumulación o focalización de información de movimiento en zonas dispersas del cuadro. Esto resulta en la formación y aumento de regiones fragmentadas que no pueden formar un *cluster* compacto dentro del MGMI, y por lo tanto, la cantidad de movimiento detectado en las manos pueden perder área y protagonismo de movimiento (por contraste, elementos ajenos a las manos podrían ganar mayor protagonismo).

En la etapa del filtrado y dentro de las etapas de la cadena de caracterización se ha mostrado el costo computacional. Es importante resaltar la observación de que todas las operaciones que están relacionados con convoluciones se implementan de manera muy eficiente. Por otro lado, la etapa de mayor demanda computacional teórica encontrada es la Varianza local, por lo cual ella debe ocupar gran parte del tiempo de procesamiento del algoritmo; sin embargo, no es lo suficientemente complejo para evitar una implementación total eficiente del algoritmo descrito en la tesis.

- **Generación del Mapa de movimiento global en bloques (MGMB)**

La descripción formal de esta etapa es:

1) Entradas:

I_{MGMI}: El Mapa global de movimiento individual.

2) Salidas:

I_{MGMB}: El Mapa global de movimiento en bloques.

3) Proceso:

$$\text{lado\_bloque\_estandar} = \text{dimension\_x\_mano} * \text{constante\_de\_proporcionalidad} \tag{3.12}$$

$$m1 = floor(ROWS/\text{lado\_bloque\_estandar}) \tag{3.13}$$

$$n1 = floor(COLS/\text{lado\_bloque\_estandar}) \tag{3.14}$$

$$(\mathrm{I}_{MGMB})_{ij} = \sum_{k,l \in \mathrm{Bl(i,j)}} (\mathrm{I}_{MGMI})_{kl}, \quad \begin{matrix} dim(\mathrm{I}_{MGMB}) = (m1)x(n1), \\ dim(\mathrm{Bl(i,j)}) = (tam\_bloque)^2 \end{matrix} \tag{3.15}$$

4) Costo computacional:

$$O(ROWS\ COLS) \tag{3.16}$$

(a) Imagen del cuadro.

(b) MGMI.

(c) MGMI sin la etapa de filtrado dentro de la Diferencia absoluta temporal-local.

(d) MGMI sin la etapa de Varianza local.

Figura 3.2: Efectos de las modificaciones dentro de la cadena de caracterización.

El Mapa global de movimiento en bloques o MGMB es una imagen que representa las distintas regiones del cuadro divididas en bloques que tienen asignados valores numéricos en relación con la cantidad de movimiento detectada dentro de ellos. El MGMB es el segundo distintivo generado a partir del análisis sin información *a priori* y se construye a partir del MGMI.

El MGMB se forma como una imagen de dimensiones *m1 x n1*, resultado de dividir al MGMI en una cuadrícula de *m1 x n1* bloques estándares, los cuales son cuadrados con un tamaño de lado dependiente de la constante de proporcionalidad y las dimensiones de la mano. Luego, por cada bloque *Bl(i,j)*, se obtiene un valor consistente en la suma de todos los píxeles que esta región abarque sobre el MGMI. Este valor es ubicado en la posición (i,j) del MGMB.

Anteriormente se menciona que la utilidad usar este distintivo es la ganancia de protagonismo de las regiones con mayor movimiento. Debido a que la cantidad de movimiento se encuentra cuantificada en bloques que representan a regiones particulares del cuadro, las regiones con píxeles segmentados dispersos tendrán un menor valor asociado, y pierden importancia frente a otra regiones con píxeles segmentados más concentrados de movimiento. Por este motivo, este distintivo se complementa con el MGMI y permite reducir los falsos positivos originados por la mayor cantidad de hipótesis de los datos de entrada. De este modo, los bloques que pertenezcan a las regiones de las manos, tienen un valor numérico mayor dentro del Mapa global de movimiento en bloques en relación con su cantidad de movimiento.

- **Uso del Mapa global de movimiento individual: El Análisis de movimiento subyacente**

El Análisis de movimiento subyacente es el proceso por el cual se analiza si un píxel o conjuntos de píxeles asociados a la ubicación de alguna posición de interés (como puede ser la mano) tienen el valor ”1” dentro del MGMI y saber si pertenecen a una región con cantidad de movimiento considerable. Se realiza de dos maneras distintas:

1. **Análisis de movimiento subyacente individual:** Determina si el píxel que identifica a la posición de la mano tiene el valor ”1”. Esta manera es usada cuando se desea analizar si existe una pérdida de seguimiento.

2. **Análisis de movimiento subyacente vecino:** Se analiza a un conjunto de píxeles dentro de una ventana alrededor de alguna posición de interés en búsqueda de la existencia de al menos un píxel que tenga el valor ”1” dentro de ella. Este proceso se usa para resolver problemas de oclusión.

- **Uso del Mapa global de movimiento en bloques: El Desplazamiento global**

Debido a los valores numéricos en el MGMB, este puede ser usado para desplazar la ubicación de las manos hacia alguna nueva posición en función de la cantidad de movimiento asociados a las regiones de bloques a las que fue dividido el cuadro. Este procedimiento de ubicación de las manos es llamado *desplazamiento global* porque las manos pueden posicionarse en *cualquier* región dentro del cuadro.

Se puede definir la posición (k,l) como el centro del bloque (i,j) sobre la imagen del cuadro actual:

$$k = i * \text{lado\_bloque\_estandar} + floor(lado\_bloque\_estandar/2)$$

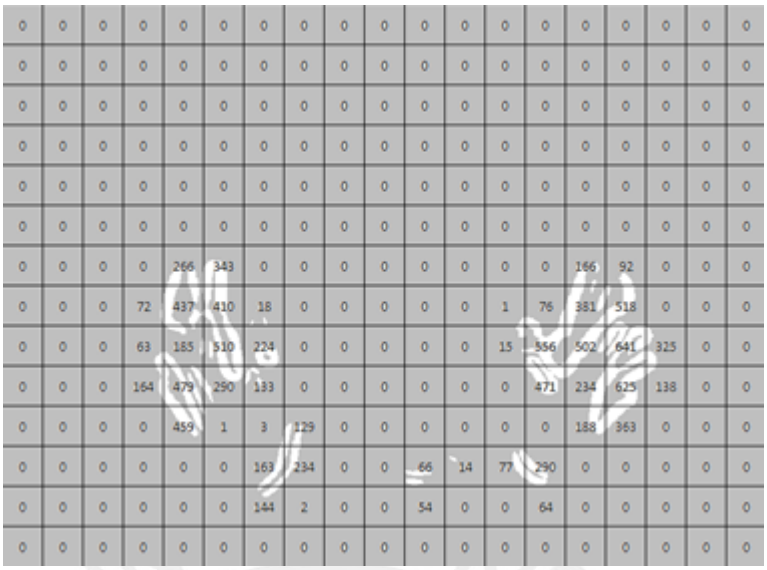


Figura 3.3: Mapa global de movimiento en bloques asociado al cuadro de la figura 3.2a.

$$l = j * \text{lado\_bloque\_estandar} + floor(lado\_bloque\_estandar/2)$$

El algoritmo de desplazamiento global es el presentado en la figura 3.4. El desplazamiento global puede servir para ubicar ambas manos o una de ellas en particular, lo cual solo es posible cuando las manos no presentan oclusión mutua, porque es necesario que dentro del MGMB se tengan diferenciados dos *clusters* de valores no nulos según un vecindario *8-neighborhood* (ver figura 3.5). Cada uno de estos dos *clusters*, al corresponder a las regiones con mayor cantidad de movimiento, tendrá ubicado a una mano distinta.

Los *clusters* se ubican de manera secuencial. En el caso del desplazamiento global de ambas manos, primero se ubica el bloque (i,j) de mayor valor numérico en el MGMB y luego se ubica a la mano en su posición (k,l) asociada. Después, se enmascara el *cluster* al cual el bloque (i,j) pertenece por medio de una eliminación propagativa de los bloques 8-neighborhoodd de valores no nulos, asignándoles un valor nulo a los bloques eliminados (la forma de anular los bloques del *cluster* se implementa mediante un *backtracking* recursivo). Finalizado este proceso, queda en teoría un *cluster* adicional dentro del MGMB, razón por la cual se ubica nuevamente el bloque (i,j) de mayor valor numérico en el MGMB y en l se ubica la posición de la segunda mano. Conviene resaltar que en el caso del desplazamiento global de solamente una mano, simplemente se procede a la eliminación propagativa del *cluster* asociado a la mano que no se desea desplazar y se continúa el resto del procedimiento del mismo modo a cómo se ha descrito anteriormente.

Puede ocurrir que se tengan más de dos *clusters* aislados, en cuyo caso, se puede afirmar que por las condiciones del seguimiento vistas en la sección 3.1.1. , este tercer *cluster* tendrá valores relativamente menores comparados los otros dos principales porque posiblemente se ha originado por algún movimiento indeseado colado durante la etapa de segmentación unimodal del MGMI. Esta es otra ventaja del fenómeno de ganancia de protagonismo presente en el MGMB.

Por último, puede ocurrir la indeseada posibilidad de que exista un solo *cluster* pese a no existir una oclusión aparente entre las manos. Esto puede suceder porque ambos *clusters* hipotéticos relacionados a cada mano no están separados por bloques de valores nulos, sino que lo están por bloques con valores no nulos muy pequeños debido a algún elemento de movimiento indeseado. Como resultado, en el proceso de

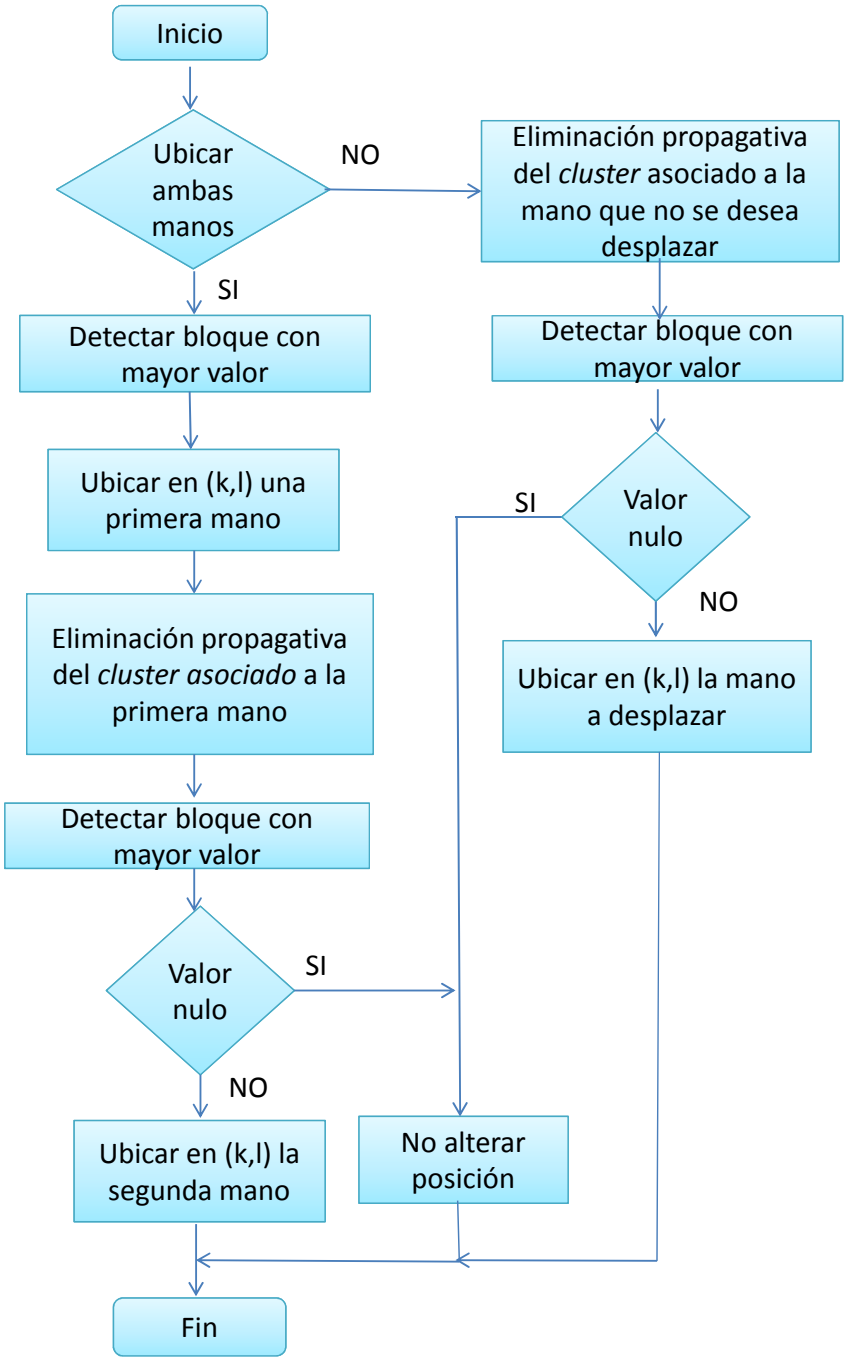


Figura 3.4: Algoritmo de desplazamiento global.

eliminación propagativa, se elimina a todo este único cluster y la posición de la segunda mano no es alterada.

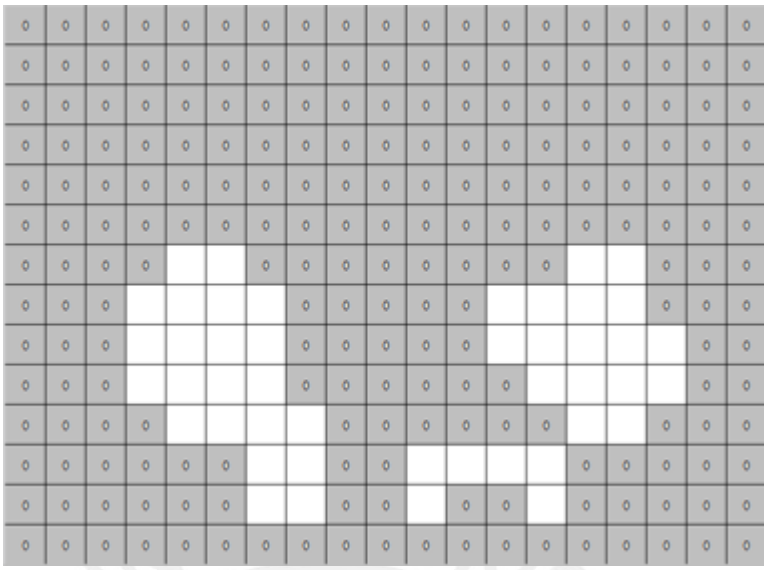


Figura 3.5: MGMB de la figura 3.3 con dos *clusters* de valores no nulos diferenciados según un vecindario *8-neighborhood*.

3.2.2.2. Análisis con información a priori: Mapa local de similitud (MLS)

El análisis de movimiento con información *a priori* es posible gracias a que el algoritmo conoce el desplazamiento máximo permitido para las manos respecto al cuadro anterior. Por tal motivo, la mano se ubica en el cuadro actual mediante un análisis del movimiento dentro de un vecindario de posiciones alrededor de la ubicación que la mano tenía en el cuadro anterior, y el cual se define en función del desplazamiento máximo permitido. El Mapa local de similitud o MLS es el distintivo que expresa numéricamente la cantidad de movimiento analizada dentro de este vecindario, y cuando se ubican a las manos mediante su uso, se refiere a que hubo un *desplazamiento local*.

La ventaja del uso de información *a priori* por medio del desplazamiento local es que permite un análisis dentro de un menor espacio de hipótesis porque hay un rango menor de posibles posiciones de la mano comparado a lo que ocurre en un desplazamiento global, y por lo tanto, se infiere que el error de ubicación debe ser menor.

Las limitaciones en el análisis con información *a priori* se relacionan con la oclusión o movimientos muy raudos de la mano que originen una pérdida de seguimiento e inhiban al desplazamiento local de poder seguirla.

- **Generación del Mapa local de similitud (MLS)**

La descripción formal de esta etapa es:

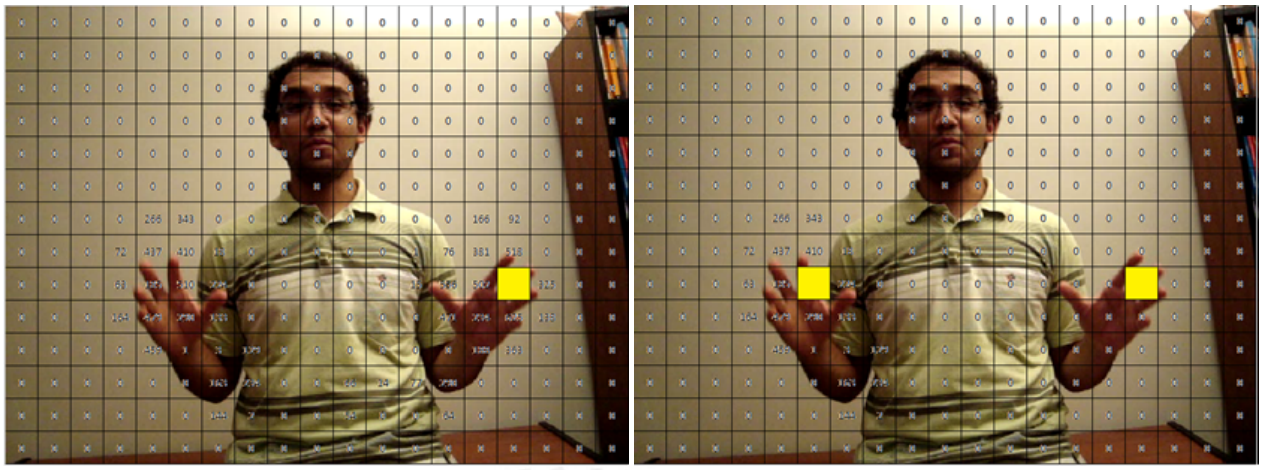
1) Entradas:

I_t: Cuadro actual luego de la etapa de filtrado y acondicionamiento.

I_{t-1}: Cuadro anterior luego de la etapa de filtrado y acondicionamiento.

x_{t-1}: Coordenada abscisa *x* de la posición de la mano en el cuadro anterior.

y_{t-1}: Coordenada abscisa *y* de la posición de la mano en el cuadro anterior.



(a) Ubicación de la primera mano. (b) Ubicación de la segunda mano.

Figura 3.6: Ubicación de ambas manos usando el MGMB. El cuadro es el mismo de la figura 3.2a. Se resalta de color amarillo el bloque en cuyo centro se ubicará la posición de cada mano.

d_max: Distancia máxima de la ventana exploratoria para el *matching* de cada sección del MLS, relacionado al valor máximo que puede desplazarse una mano entre cuadros sucesivos.
lado_bloque: Dimensión del lado del bloque estándar.

2) Salidas:

I_{MLS}: El Mapa local de similitud.

3) Proceso:

$$(\mathrm{I}_{MLS})_{ij} = min\{\xi_{ab}\}, \quad \begin{matrix}(\mathrm{I}_{MLS})_{ij} \in S, dim(S) = 3x3, S = N^{sym,sep=lado\_bloque}_{c(\mathrm{x}_{t-1},\mathrm{y}_{t-1})} \\ \xi_{ab} \in S_1, dim(S_1) = (\mathrm{d\_max} * 2 + 1)^2, S_1 = N^{sym}_{c((\mathrm{I}_{MLS})_{ij})}\end{matrix} \tag{3.17}$$

$$\xi_{ab} = \sum_{m,n \in S_2} ((\mathrm{I}_t)_{mn} - (\mathrm{I}_{t-1})_{mn})^2, \; dim(S_2) = (\mathrm{lado\_bloque})^2, S_2 = N^{sym}_{c(\xi_{ab})} \tag{3.18}$$

4) Costo computacional:

$$O(18 \text{ d\_max[ROWS,COLS]}^2 \text{ lado\_bloque[ROWS,COLS]}^2) \tag{3.19}$$

El primer paso dentro del análisis es ubicar la posición de la mano en el cuadro anterior como el centro de una región de bloque estándar en el cuadro actual. Luego, se ubican las regiones asociadas a ocho bloques adicionales dentro del *8-neighborhood* del bloque ubicado inicialmente. Cada uno de estos nueve bloques está asociado a una entrada del I_{MLS} y en conjunto constituyen la región de análisis (hipótesis) para la ubicación de la mano por desplazamiento local: el MLS es una matriz de 3x3 elementos. Por cada bloque (i,j) asociado a la entrada (i,j) del I_{MLS}, se ubica un vecindario de *d_max* posiciones simétricamente alrededor de la posición del centro del bloque (i,j) en el cuadro anterior. Por cada posición (a,b) de este vecindario, se centra un bloque y se calcula la suma de diferencias al cuadrado (SSD) entre cada píxel del bloque (i,j) y el de (a,b) que acaba de ubicarse. Este valor resultante es llamado ξ_{ab} y es almacenado dentro de un arreglo en memoria. El SSD es un proceso diferencial que permite obtener un valor de similitud: a menor valor, mayor es la similitud del bloque (i,j) del cuadro actual con respecto al (a,b) ubicado en el cuadro anterior. Por tal

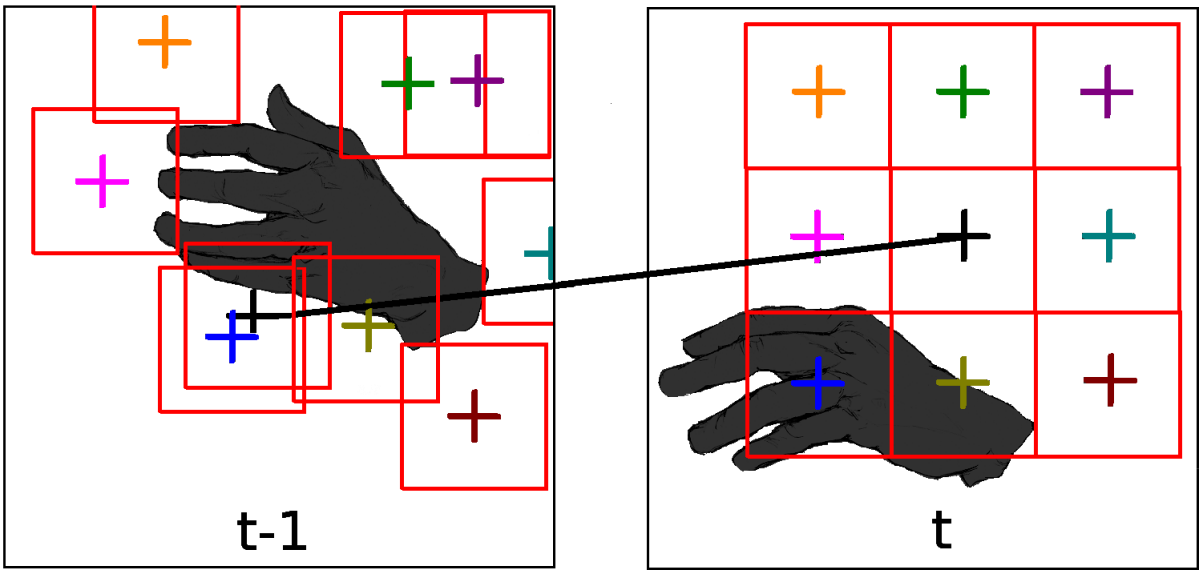


Figura 3.7: Figura en el cual es posible apreciar como a partir del MLS formado en la posición anterior de la mano dentro del cuadro actual "t", en cada una de sus celdas (con su centro identificado por una cruz de color distinto) se ubica la región con mayor similitud (*matching*) en el cuadro anterior "t-1".

motivo, luego se realiza un *matching*: se ubica el mínimo valor ξ_{ab} dentro del arreglo en memoria y lo asigna como el valor de similitud para la posición (i,j). Este proceso se realiza para cada uno de los nueve bloques constituyentes de la región de análisis y así completar los valores dentro de la matriz I_{MLS}. Otro trabajo que usa el SSD en el contexto de aplicaciones de seguimiento es [26], aunque también es usualmente usado en diversas aplicaciones de *matching* [10].

En conclusión, el MLS captura cuantitativamente la cantidad de movimiento en base a qué tan similar en apariencia es una región en particular del cuadro actual con su posición más probable dentro de un vecindario en el cuadro anterior. Sin embargo, también puede analizarse como un mapa de *disimilitud*: un mayor valor en alguna entrada del MLS indica que la región asociada a ella tiene un menor parecido (o una mayor variación) a cómo fue en el cuadro anterior, y por lo tanto, una mayor probabilidad de pertenecer a una región que ha sufrido el movimiento de alguún elemento en él respecto al cuadro anterior. La bondad de usar el SSD es que eleva cuadráticamente las diferencias de intensidades durante el *matching*, con el propósito de amplificar sus disimilitudes.

- **Desplazamiento local: Ponderación local de movimiento**

La descripción formal de esta etapa es:

1) Entradas:

I_{MLS}: El Mapa local de similitud.

lado_bloque: Dimensión del lado del bloque estándar.

x_{t-1}: Coordenada abscisa *x* de la posición de la mano en el cuadro anterior.

y_{t-1}: Coordenada abscisa *y* de la posición de la mano en el cuadro anterior.

2) Salidas:

x_local$_t$: Abscisa de la nueva posición de la mano en el cuadro actual.

y_local$_t$: Ordenada de la nueva posición de la mano en el cuadro actual.

3) Proceso:

$$\text{pos\_x}_i = \text{coordenada } x \text{ del centro de posición } ij \text{ del MLS} \tag{3.20}$$

$$\text{pos\_y}_j = \text{coordenada } y \text{ del centro de posición } ij \text{ del MLS} \tag{3.21}$$

$$\begin{aligned} \text{x\_local}_t &= \frac{\sum_{i,j\in S} \text{pos\_x}_i * (\text{I}_{MLS})_{ij}}{\sum_{i,j\in S} (\text{I}_{MLS})_{ij}} \\ \text{y\_local}_t &= \frac{\sum_{i,j\in S} \text{pos\_y}_j * (\text{I}_{MLS})_{ij}}{\sum_{i,j\in S} (\text{I}_{MLS})_{ij}} \end{aligned} , \qquad dim(S) = 3x3, S = N^{sym}_{c(\text{x}_{t-1},\text{y}_{t-1})} \tag{3.22}$$

4) Costo computacional de la implementación:

$$O(1) \tag{3.23}$$

La *ponderación local de movimiento* es el mecanismo cuantitativo que permita desplazar la mano en función a los valores de disimilitud asociado a las nueve regiones de bloques estándares del MLS y las posiciones centrales de cada una de ellas. La ponderación ubica la nueva posición de la mano como el cálculo del centroide de toda la región explorada en el MLS según los pesos que se contengan en los valores asociados a cada uno de sus bloques que la conforman. Cada centro de bloque se considera como una posición en el cálculo del centroide, donde cada peso de ponderación viene dado por el valor de disimilitud asociado a dicho bloque. El centroide tiene las importantes ventajas de poderse mover *virtualmente* en cualquier dirección y magnitud menor al desplazamiento máximo para poder ubicarse sobre el punto donde exista mayor disimilitud o equilibro de cantidad de movimiento presente en el área de análisis. Este método es entonces ideal para el seguimiento de cuerpos no rígidos como las manos, cuyos movimientos particularmente no están limitados en cuanto dirección.

El Mapa local de similitud tolera y es robusto frente a cambios de iluminación global en todo el cuadro, porque este fenómeno solo introduce una variación de intensidades *similar* sobre todos los valores resultantes del SSD asignado en el MLS. Como resultado, aquellas celdas que hubiesen obtenido un mayor protagonismo dentro del cálculo de ponderación sin que ocurra un cambio de iluminación global, lo seguirán manteniendo pese a que este cambio sí ocurriese. Conviene resaltar que solo se refiere a cambios de iluminación *no drásticos*: aquellos que no cambian ningún signo de la diferencia entre cualquier par de píxeles analizados entre el cuadro anterior y actual; de lo contrario, no existe certeza en poder predecir un comportamiento correcto del desplazamiento local.

Por otro lado, existe un problema con la ponderación local de movimiento: el tamaño de la región de análisis asociado al MLS permite la existencia del problema de *drifting*. Si la ventana aumenta de tamaño, habría una mayor posibilidad de direcciones y tamaños de posibles desplazamientos locales; sin embargo, también se introduciría mayor información de movimiento de regiones que anteriormente no se consideraban. Como consecuencia, este aumento de análisis también influye en la ponderación y puede generar un *drift* o arrastre del desplazamiento hacia regiones que no sean las de interés. Existen dos ejemplos notorios de *drifting*. El primer ejemplo se presenta cuando ocurre bastante acercamiento entre ambas manos sin ocluirse, pero que son suficientes como para que la región de análisis del MLS considere a regiones de la otra

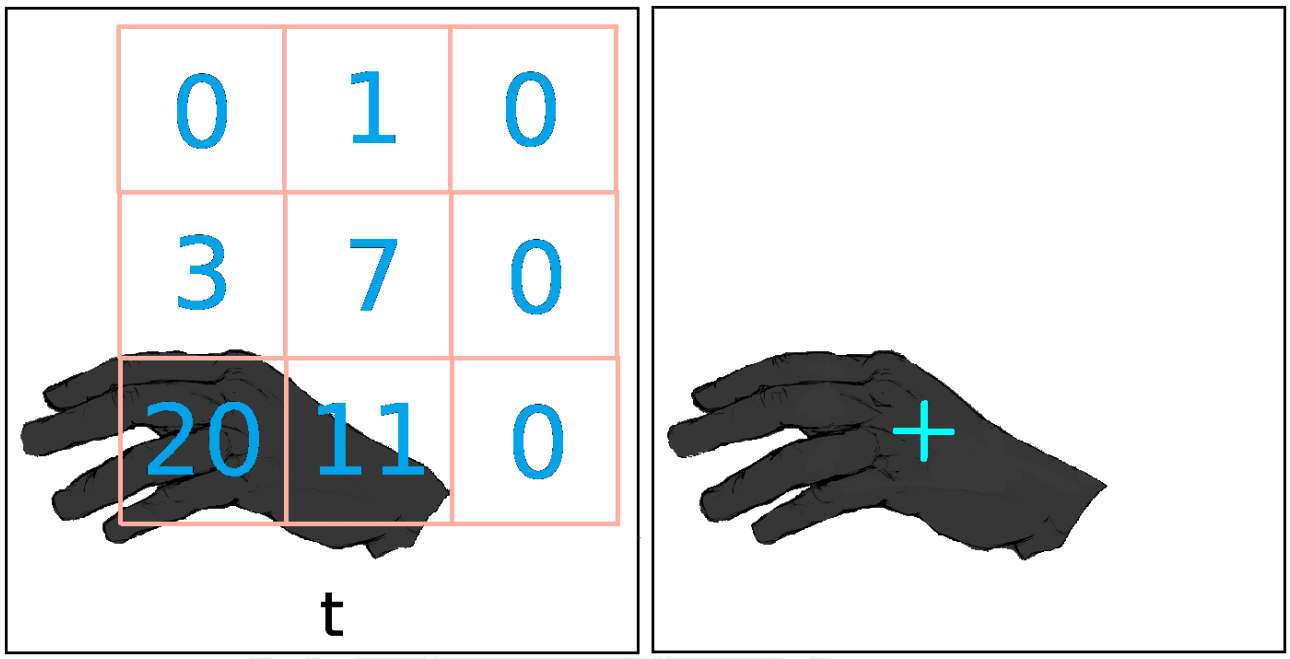


Figura 3.8: En esta figura se muestra gráficamente la ubicación de la mano en el cuadro actual (derecha) resultante de ubicar el centroide por ponderación del MLS asociado (izquierda) para la situación descrita en 3.7.

mano como parte de su hipótesis y generar un *drift* hacia la región de la otra mano. El segundo ejemplo es que la región de análisis podría abarcar otras partes del cuerpo del usuario que también están en movimiento y que se desean evitar, como son los brazos o codos, pero que también pueden originar un importe de *drift* hacia su ubicación.

En la presente tesis, el tamaño de la región de análisis del MLS se define por el propio algoritmo en relación con el tamaño de la mano y su máximo desplazamiento, y por lo tanto, se reduce la presencia del fenómeno de *drifting*.

Capítulo 4

Desarrollo de la solución: Algoritmo

4.1. Descripción del algoritmo

4.1.1. Diagrama de flujo

En la figura 4.1 se presenta el diagrama de flujo general de la solución de la tesis, en el cual conviene precisar lo siguiente:

1. Existen dos recuadros de color verde, los cuales pertenecen a una etapa general de *Extracción de distintivos*, en la cual se generan en paralelo el distintivo *Mapa global de movimiento* y, por cada mano, el *Mapa local de similitud*.

2. Los recuadros de color amarillo y rojo engloban procesos que pertenecen a la etapa de *Procesamiento de distintivos*. Los recuadros amarillos engloban a distintas etapas que procesan al Mapa gobal de movimiento individual (MGMI) y al Mapa global de movimiento global en bloques (MGMB), mientras que el rojo engloba a una etapa que procesa al Mapa local de similitud (MLS).

3. La mayoría de etapas son de color celeste e indican un único proceso. Sin embargo, las de color naranja indican que dentro de ellas existen más procesos, por lo cual pueden tener más de una flecha como salida. Las etapas de color naranja se exponen posteriormente en los diagramas de flujo *Manejo de oclusión* (figura 4.2) y *Análisis de movimiento circundante* (figura 4.3); dentro de los cuales, los rectángulos de color violeta representan etapas mostradas en el diagrama de flujo general de la figura 4.1.

4. Todas las etapas que poseen un asterisco (*) son realizadas en cada mano de manera independiente, es decir, que todos los procesos dentro de dichas etapas se repiten por cada mano. Caso contrario, los procesos se realizan una sola vez para ambas manos conjuntas.

Se explica el funcionamiento del algoritmo en base a lo presentado en los diagramas de flujo. En el primer comienzo del seguimiento se encuentra la etapa de inicialización, en la cual el algoritmo conoce la ubicación inicial y las dimensiones constantes de las manos dentro del cuadro. El resto del seguimiento es la tarea repetitiva de localizar las posiciones de ambas manos del usuario en los cuadros sucesivos, y en cada uno de estos cuadros, el *seguimiento finaliza* cuando se halla y muestra la posición de la mano en la pantalla del computador, y espera volver a empezar el seguimiento con la entrada del siguiente cuadro.

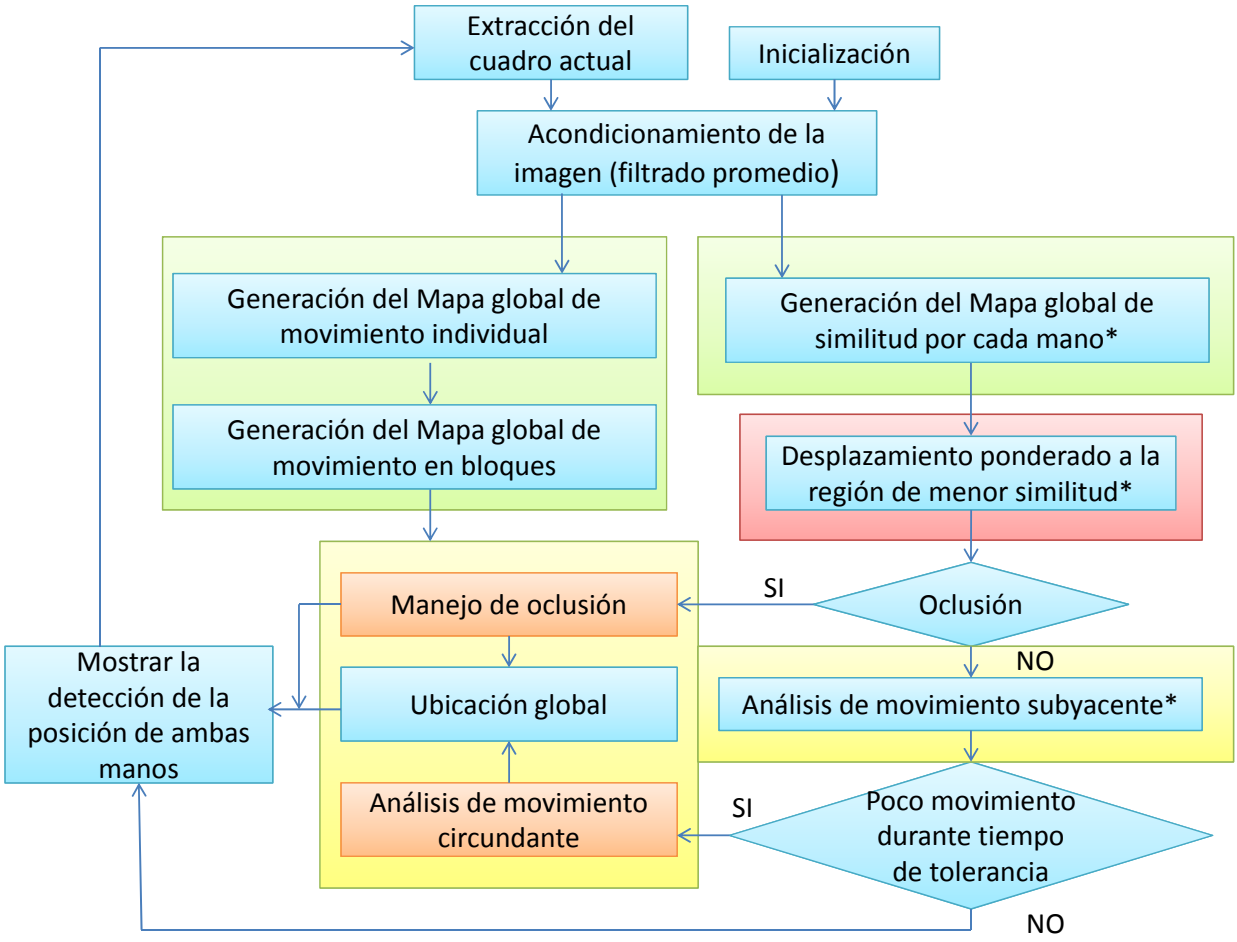


Figura 4.1: Diagrama de flujo general de la solución al seguimiento de manos presentado en la presente tesis.

Los efectos negativos para los distintivos de movimiento originados por el nivel de ruido o regiones de alta textura presentes en el cuadro, son reducidos con la etapa de acondicionamiento o filtrado de la imagen. Luego, se procede a la etapa de Extracción de distintivos, en la cual se extraen simultáneamente el MGMI y MGMB sin el uso información *a priori*, y del MLS con el uso de información *a prori*.

El MLS se obtiene independientemente por cada mano y con ella se obtiene una primera ubicación de la mano por desplazamiento local en base a la posición que tuvo en el cuadro anterior. Esta nueva ubicación es llamada posición de *referencia local* o *prl*. La *prl* no es la nueva ubicación definitiva con la cual finaliza el seguimiento, sino que es una primera aproximación de posible desplazamiento, la cual es alterada o reafirmada de acuerdo a tres eventos que pueden ocurrir luego de este desplazamiento:

1. La mano puede haber estado en una situación de pérdida de seguimiento desde el cuadro anterior, por lo tanto, el desplazamiento local ha ubicado la mano en una zona errónea y la mano simplemente se desplazó por efectos del ruido o *drift*.

2. Luego del desplazamiento local, se encuentra que las manos están lo suficientemente cercanas para considerarlas en oclusión.

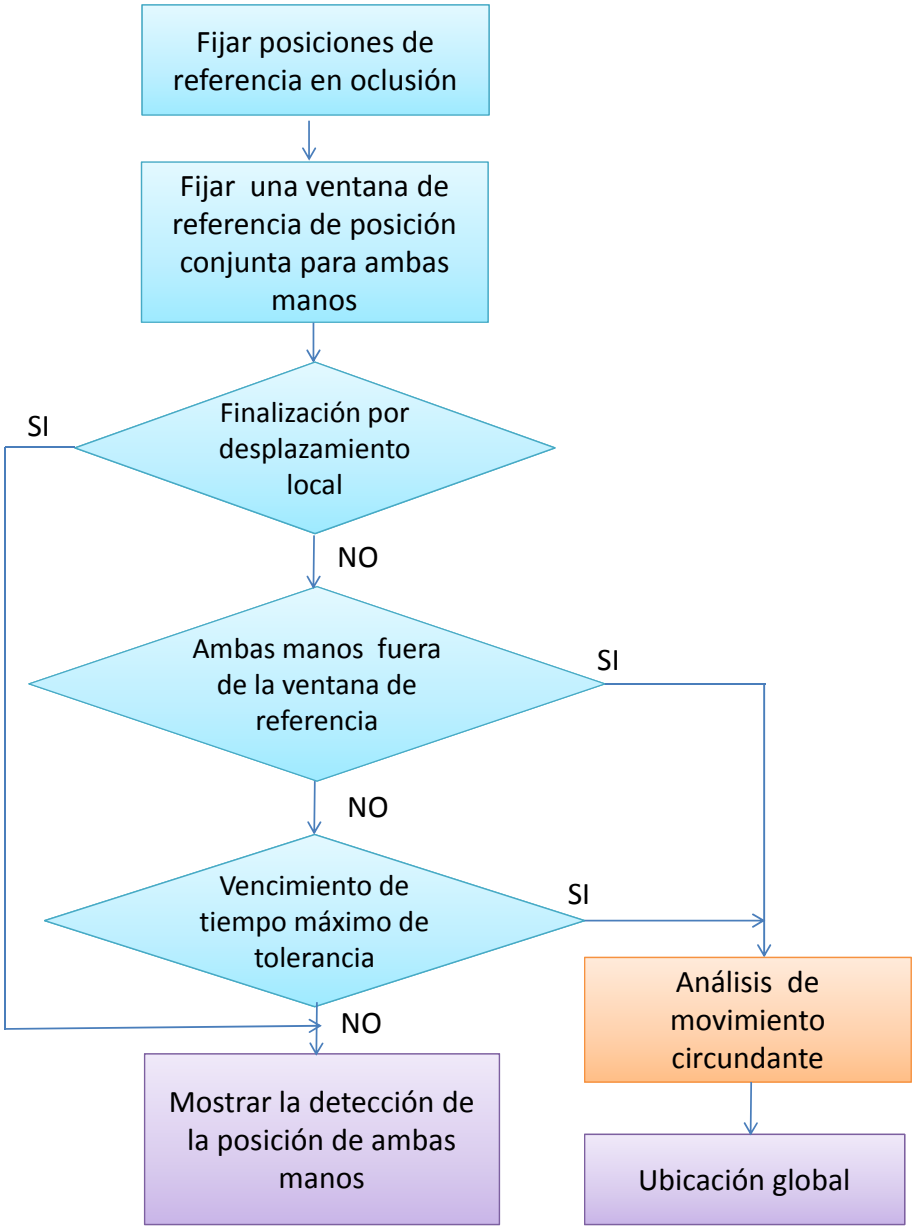


Figura 4.2: Diagrama de flujo específico de Manejo de Oclusión.

3. Si no ocurre ninguna de las dos posibilidades anteriores, se considera al desplazamiento local como *correcto* para representar la ubicación de la mano y se finaliza el seguimiento.

En el caso que ocurra la primera posibilidad, la pérdida de seguimiento no es detectada con el simple análisis de dos cuadros consecutivos, sino que es detectada luego de transcurrir una cantidad de cuadros presentes en un intervalo de tiempo llamado *tiempo de tolerancia*. El tiempo de tolerancia se inicia cuando por medio del análisis de movimiento subyacente individual (ver sección 3.2.2.1) sobre la mano se detecta un valor de ”0”, y sigue transcurriendo en los siguientes cuadros siempre y cuando se continúe con este

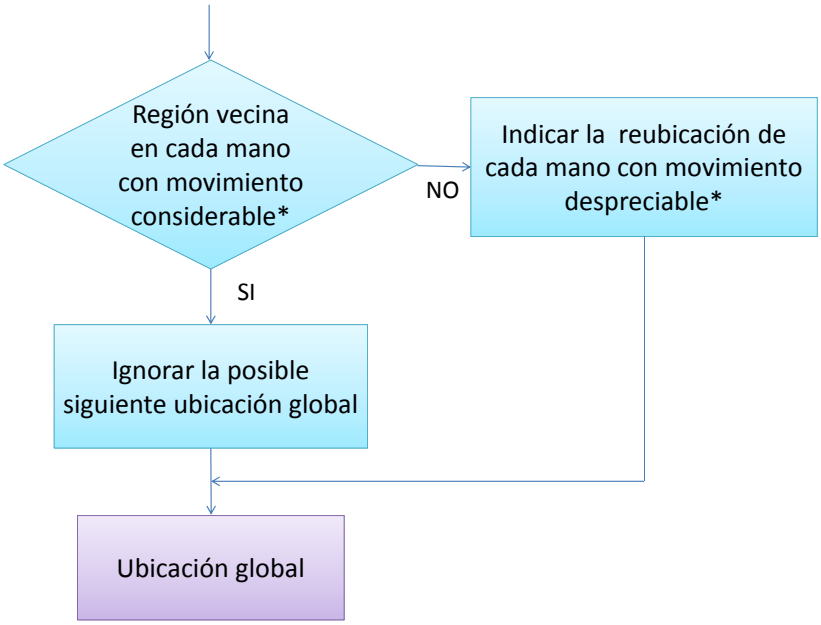


Figura 4.3: Diagrama de flujo específico de Análisis de movimiento circundante.

valor detectado; de lo contrario, se lo reinicia. Durante el tiempo de tolerancia, el desplazamiento local se considera *correcto*, siempre y cuando no ocurra alguna oclusión. Cuando se finaliza el tiempo de tolerancia, se afirma que la detección de la mano está posiblemente sobre una región sin una cantidad de movimiento considerable dentro del cuadro y en una situación de posible pérdida de seguimiento. Entonces, se utiliza al MGMB para realizar un *análisis de movimiento circundante* que consiste en determinar si existe una cantidad de movimiento considerable dentro de alguna de las regiones vecinas a la ubicación de la mano posiblemente perdida. Si el análisis de movimiento circundante afirma la presencia de movimiento vecino, se finaliza el seguimiento con la posición obtenida inicialmente por el desplazamiento global; caso contrario, se reafirma la existencia de una pérdida de seguimiento y se finaliza el seguimiento con un desplazamiento global que ubique a la mano en alguna región con mayor cantidad de movimiento dentro del cuadro. De este modo, el análisis de movimiento subyacente dentro del tiempo de tolerancia permite cumplir con el objetivo de que el algoritmo desarrollado está continuamente recuperándose de las pérdidas de seguimiento.

Por otro lado, en el caso que ocurra la segunda posibilidad, el desplazamiento local no puede actualizarse como ubicación de las manos, sino que se fijan dos *posiciones de referencia en oclusión* o *pro*'s por cada mano, para que ellas representan a las manos durante todo el tiempo que ocurra la oclusión. Esta fijación permite separar la identificación de ambas manos durante la oclusión, evitando que las posiciones de ambas manos detectadas converjan a una sola y sea más difícil diferenciarlas en la continuación del seguimiento. La oclusión puede terminar de tres formas distintas. La primera es que las manos puedan separarse por simple desplazamiento local a partir de las *pro*s y finalizar así el seguimiento. La segunda es que se detecte la salida de las manos fuera de un vecindario alrededor de las *pro*s por medio de un análisis de movimiento subyacente vecino (ver sección 3.2.2.1). Y la tercera forma de salir es por el vencimiento de un tiempo máximo de tolerancia a la oclusión. Luego de terminar la oclusión por la segunda o tercera forma, se realiza un análisis de movimiento circundante y su posible desplazamiento global para finalizar el seguimiento. En

caso de que la oclusión no se finalice en el cuadro actual, simplemente se finaliza el seguimiento mostrando los *pro*s como ubicaciones de las manos.

En conclusión, el algoritmo trata de finalizar el seguimiento del cuadro actual con la correcta ubicación de la mano dentro de la región que mejor la caracterice en base a los distintivos de movimiento presentes. Pueden ocurrir problemas de pérdida de seguimiento, pero las posiciones son recuperadas siempre después de un tiempo de tolerancia. Por último, el algoritmo permite modelar la situación de oclusión y continuar el seguimiento después de que ella termine.

4.1.2. Procesos o etapas principales

Se describen los procesos principales observados en el diagrama de flujo en términos de su utilidad y funcionalidad dentro del seguimiento de manos.

4.1.2.1. Inicialización

La inicialización es la primera etapa del algoritmo y es la responsable de recibir la información inicial necesaria para comenzar el seguimiento de manos. Es la etapa responsable de la cualidad parametrizable del algoritmo desarrollado, puesto que el usuario requiere ingresar los parámetros de la resolución actual de los cuadros y el tamaño de las manos (largo y ancho) dentro de ella: todo esto permite que el seguimiento pueda darse sobre cuadros de distintas resoluciones y exista una mayor independencia de la plataforma de implementación. Además, el usuario solo requiere agitar las manos para que el sistema automáticamente inicialice las posiciones de las manos. Todas estas características permiten que la inicialización sea sencilla e intuitiva de configurar, a la vez que reduce el ingreso de datos iniciales errados al procesamiento del algoritmo. Además, el usuario no necesita ingresar ningún factor o procedimiento de calibración inicial, ni realizar algún entrenamiento previo al algoritmo. En este punto conviene recordar las restricciones y condiciones de aplicación del seguimiento descritas en la sección 3.1.1. .

4.1.2.2. Extracción de distintivos

Este es el proceso por el cual se extraen los distintivos de movimiento necesarias para caracterizar las manos por medio de su movimiento. La premisa es que aquellas regiones con mayor cantidad de movimiento corresponden a las regiones donde posiblemente se ubique la mano en el cuadro actual según las condiciones y restricciones necesarias para el seguimiento correcto en la presente tesis. En la figura 4.4 se presenta un diagrama resumen en donde se relaciona la etapa de Extracción de distintivos con el resto de etapas o procesos realizados en el seguimiento.

4.1.2.3. Procesamiento de distintivos

El procesamiento de distintivos es la etapa por la cual se interpreta la información de cantidad de movimiento presente en regiones del cuadro y en base a ellas se obtiene un resultado para poder ubicar las posiciones de las manos.

- **Complementariedad del análisis *sin* y *con* información *a priori***

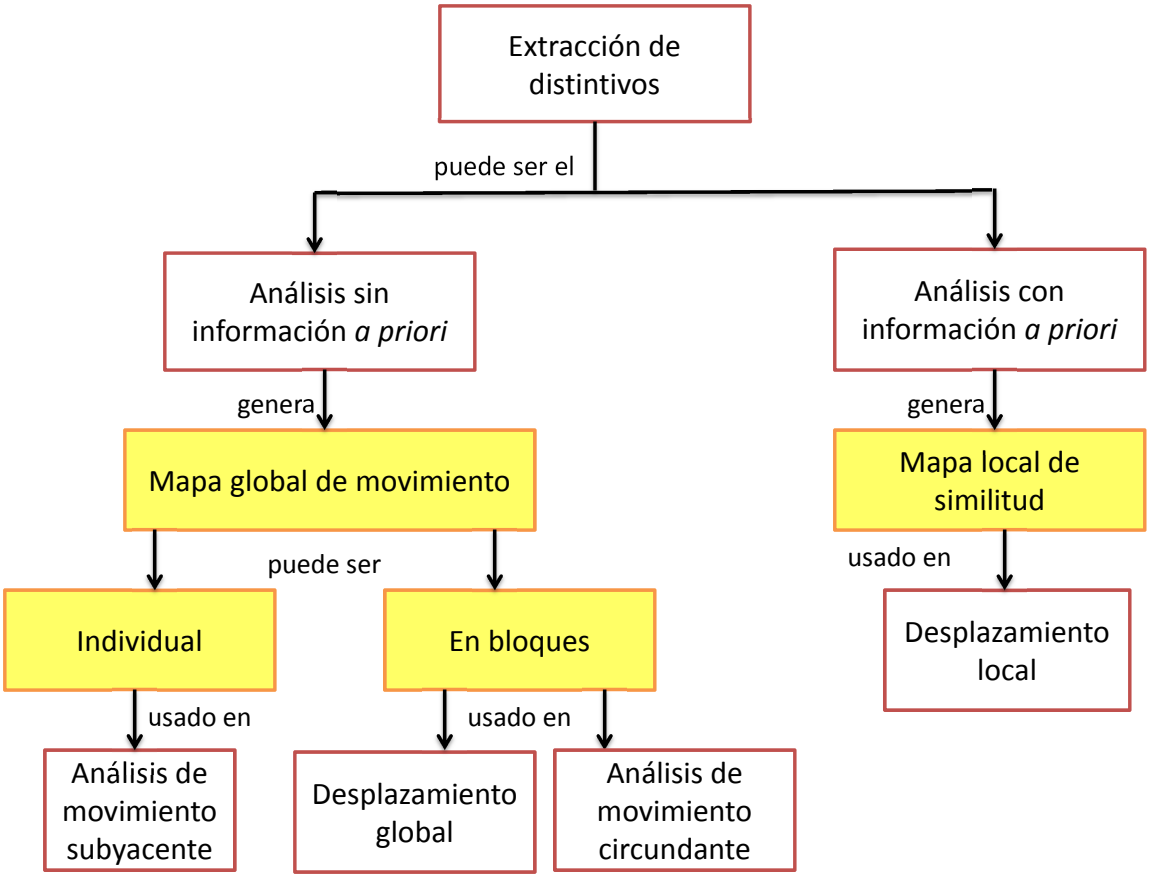


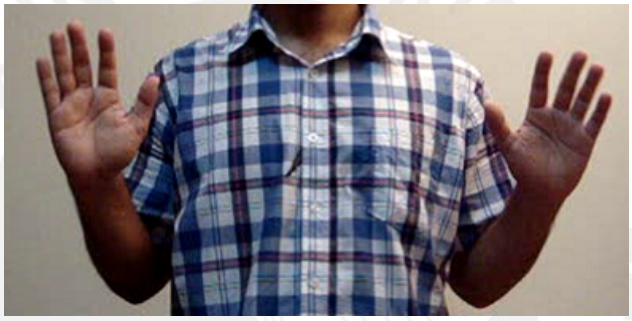
Figura 4.4: Relación de la etapa de Extracción de características con las otras etapas relacionadas al seguimiento.

Anteriormente se puede observar que el desplazamiento global es solamente usado para remediar las dos situaciones problemáticas resultantes del desplazamiento local: oclusión y pérdida de seguimiento. Se observa que el uso de ambos tipos de desplazamientos, diferenciados por el uso o no de información *a priori*, recae en una relación de *complementariedad* mutua que presentan dentro del seguimiento. La premisa básica de esta complementariedad es: ”solo cuando el desplazamiento local no sea *suficiente* para ubicar la mano, utilizar el desplazamiento global”. Dicho de otro modo: ”solo ubicar la mano mediante el uso de información *a priori* de la posición de la mano en el cuadro anterior, cuando dicha información sea confiable”.

El desplazamiento global no puede tener preferencia sobre el desplazamiento local para la ubicación de las manos debido a que la primera requiere de un mayor espacio de análisis de hipótesis (todo el cuadro). Empíricamente se pudo establecer que esta mayor cantidad de hipótesis, al implicar la existencia de un mayor espacio de búsqueda para encontrar a la mano, también implica la posible existencia de un mayor *error* de precisión en la ubicación correcta de la mano. Este problema obliga a restringir el uso del desplazamiento global solamente cuando sea estrictamente necesaria, con prioridad al desplazamiento local que considera un menor espacio de búsqueda o hipótesis. Recordar que el espacio de búsqueda del desplazamiento local no puede exceder cierto desplazamiento máximo teórico respecto a la posición de la mano en el cuadro anterior. Sin embargo, el desplazamiento real obtenido por el MLS nunca llegue a este valor máximo teórico debido

a dos problemas limitantes:

1. **Cambio de intensidad aparente y degradación de la forma:** Para el análisis de los distintivos de movimiento solo importan los valores de intensidad (escala de grises) de las manos; sin embargo, existe una correspondencia entre color e intensidad: cuando la mano pierde la información de color que le permite contrastarse y delinearse con el entorno que la rodea, también le sucede lo mismo con sus valores de intensidad. La delineación es importante porque brinda una mayor compacidad a la región correspondiente a la mano, de lo contrario, sería más difusa, se degrada su forma y parte de su área podría mezclarse y difuminarse con el entorno. Este problema ocurre cuando la mano se mueve lo suficientemente rápido para perder su delineación, lo cual tiene el efecto de generar una falsa cantidad menor de movimiento dentro de los distintivos usados, de modo que se degrada la caracterización de las manos y su correcta detección. Observar la figura 4.5, en la cual hay un ejemplo de movimiento lento de la mano en la figura 4.5a y uno más rápido en 4.5b.



(a) Movimiento lento de la mano.

(b) Movimiento más rápido de la mano.

Figura 4.5: Ejemplos de movimientos de la mano. En la figura superior, de movimiento más lento, es posible observar que se mantiene la forma y la intensidad del color de la mano de forma más distinguible e íntegra comparada con la figura inferior, en donde los bordes de la mano y el color se encuentran difuminados.

2. **Cambio del área de la mano aparente frente a la cámara:** La degradación de la forma de la mano introduce también un cambio de su área aparente u observable frente a la cámara. Por otro lado, este cambio del área también sucede cuando la mano toma ciertas poses relativas frente a la cámara, como por ejemplo, pasar de mostrar la palma a mostrar su perfil. Sucede que, mientras la mano tenga una mayor región visible frente a la cámara, existe *mayor información expuesta* a ser detectada, como cuando la mano muestra la palma o su contraparte. Lo contrario sucede cuando la mano está de perfil o apuntando directamente a la cámara, debido a que el área o región que abarca disminuye. A mayor información expuesta, mejor es la caracterización de la cantidad de movimiento detectada

(a) Poses de la mano con mayor área frente a la cámara.

(b) Poses de la mano con menor área frente a la cámara.

Figura 4.6: Ejemplos de distintas poses de la manos.

porque si la mano tiene menor área, es como si se hubiese encogido y producirá menor cantidad de movimiento comparado a otros elementos movibles como los brazos, codos o cabeza: la mano puede perder protagonismo de movimiento. Observar la figura 4.6 para ver ejemplos de distintas poses de las manos frente a la cámara.

El resultado de la reducción del máximo desplazamiento teórico es el consecutivo aumento de la tasa de pérdidas de seguimiento debido a que la mano podría estar desplazándose en una mayor cantidad a lo que realmente puede tolerarse debido a los dos problemas expuestos anteriormente. Además, recordar que existe otro problema que puede acompañarlo: el *drift* presente en el MLS. Como resultado, se genera una desconfianza en el desplazamiento local y se requiere del uso de información de movimiento de regiones más lejanas a las inmediaciones de la posición errada detectada de la mano: se necesita explorar el cuadro actual en búsqueda de una mejor posición para la mano, es decir, ubicar a la mano sin el uso de información *a priori*.

En conclusión, se ha descrito con mayor detalle la complementariedad entre el análisis *sin* y *con* información *a priori*: el desplazamiento global corrige los posicionamientos dados por el desplazamiento local cuando es necesario; y el desplazamiento local trata de ser suficiente en la mayor cantidad de veces para poder disminuir los errores de preción posibles a presentarse en el desplazamiento global. Este estilo de articulación entre distitivos que varían respecto a la dimensión de sus hipótesis, es suficiente para la naturaleza determinística del seguimiento presentado en la tesis.

- **Manejo de oclusión**

En el diagrama de flujo se observa una etapa dedicada al manejo de oclusión entre ambas manos dentro del seguimiento. El estado de oclusión ocurre luego de verificar que las posiciones obtenidas por

desplazamiento local para cada mano están lo suficientemente cercanas para considerarse en oclusión, más precisamente, tienen una distancia de cercanía menor a un cuarto de la longitud máxima de la mano.

Anteriormente se menciona que durante todo el tiempo que ocurre la oclusión, la detección de cada mano se ubica sobre la posición de referencia en oclusión o *pro*, con el objetivo de evitar que eventualmente las dos posiciones distintas de las manos se junten como una sola luego de algún desplazamiento local posterior. Este procedimiento de ubicación se observa en la figura 4.7, el cual se explica a continuación. Las aspas rojas indican la posición de cada mano en el cuadro anterior y las aspas celestes son el resultado de aplicarles el desplazamiento local en el presente cuadro. Luego, se ubica la posición media de estas dos posiciones, la cual es denotada por el aspa amarillo. Finalmente, por cada posición de la mano en el cuadro anterior y el aspa amreillo, se ubica su punto medio: esta es la *pro* y está denotado por el aspa verde para cada mano.

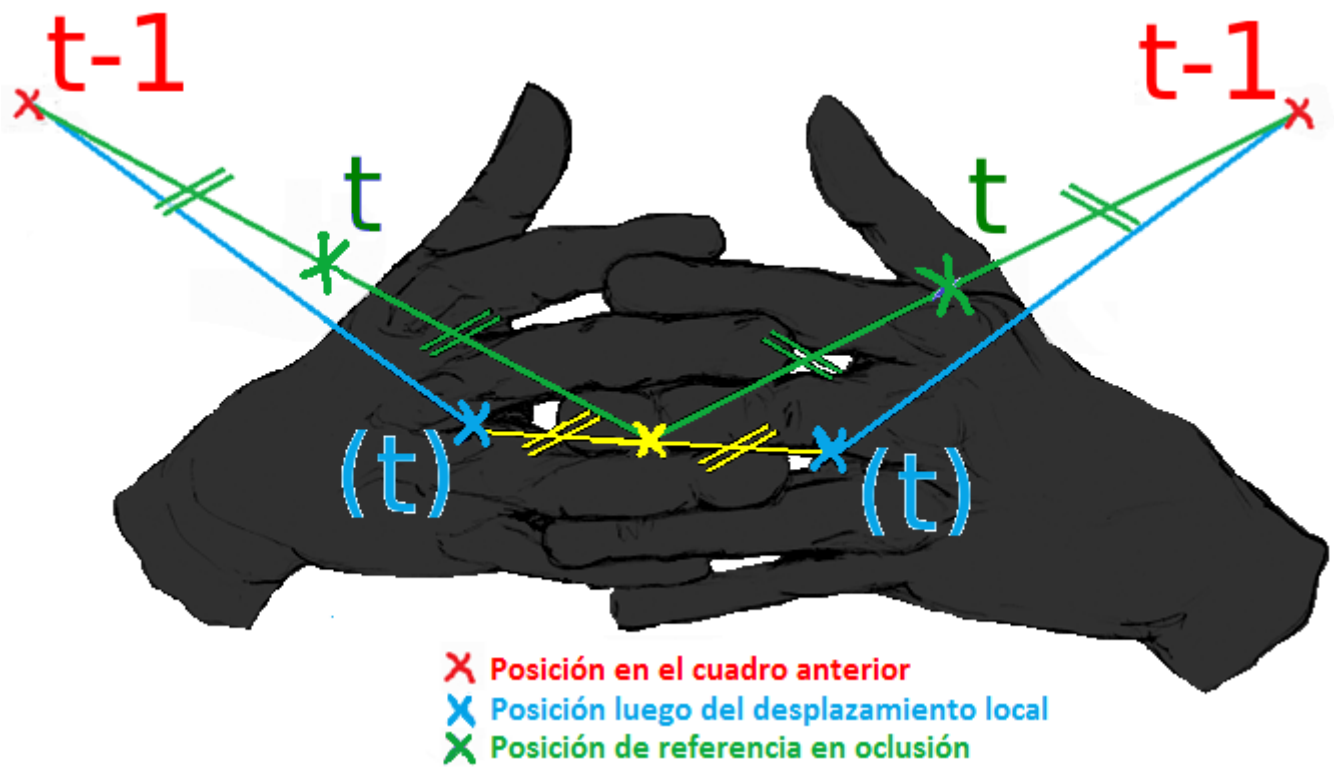


Figura 4.7: Establecimiento de la posición de referencia en oclusión.

El algoritmo cuenta con tres *mecanismos de salida* para poder finalizar el estado de oclusión. Dos mecanismos están relacionados a un análisis del movimiento de las manos, y son la *condición de movimiento externo* y el *análisis de ventana estacionaria*. El tercer mecanismo depende del vencimiento de un tiempo llamado *tiempo máximo de tolerancia*, el cual, comienza desde el instante en que se está en estado de oclusión y que una vez concluido, se realiza un análisis de movimiento circundante muy particular. Este análisis especial consiste en determinar si existe un solo *cluster* de movimiento dentro del MGMB, y si tal es el caso, se supone que las manos siguen en oclusión y por tanto permanece en dicho estado; de lo contrario, se procede a un desplazamiento global.

1.- Condición de movimiento externo

Es el mecanismo de salida de preferencia. Luego de entrar por primera vez al estado de oclusión, se analiza en lo sucesivo si es que luego de un desplazamiento local respecto a las posiciones de referencia en oclusión, las manos estarán separadas a una distancia mayor a un cuarto de la longitud mayor de la mano. Si este es el caso, se concluye la oclusión y se actualizan las posiciones de las manos según el desplazamiento para finalizar el seguimiento, de lo contrario, se mantienen las posiciones de referencia en oclusión y se procede a analizar los otros dos mecanismos de salida restantes.

2.- Análisis de ventana estacionaria

Si es que luego del análisis de la condición de movimiento externo no se finaliza la oclusión, conviene realizar otro análisis de movimiento que valide la situación de oclusión actual. El análisis anterior tiene una deficiencia: si es que las manos se han desplazado lo suficientemente rápido para no ser debidamente seguidas por desplazamiento local, los *pro*s seguirán siendo considerados como las posiciones actuales a pesar que las manos pueden estar lo suficientemente lejanas para no estar en oclusión. Esta deficiencia origina pérdidas de seguimiento, y lógicamente por el fenómeno de complementariedad mutua, deberá ser recuperada por algún posible desplazamiento global posterior.

Este mecanismo se basa en la formulación de una ventana cuadrada que tiene un lado de longitud del doble de un lado del bloque estándar, la cual es llamada *ventana estacionaria*. Se comprueba que si las manos están ubicadas en lados opuestos y fuera de esta ventana, se mantiene entre ellas una distancia suficiente para no estar en oclusión. Entonces, se realiza un análisis de movimiento subyacente vecino (ver sección 3.2.2.1.) respecto a la posición media entre las *pro*'s que determine si existe algún "1" dentro de la ventana estacionaria. En el caso sea afirmativo, es porque ambas manos posiblemente sigan en oclusión y se procede a analizar el tercer mecanismo de salida restante para descartar la posibilidad de que la cantidad de movimiento detectada sea resultado de elementos de textura o ajenos a las manos. De lo contrario, si no existe movimiento considerable dentro de la ventana estacionaria, es porque se terminó la oclusión y se repite el proceso análisis de movimiento circundante y desplazamiento global.

3.- Tiempo máximo de tolerancia

Existe una situación para la cual el mecanismo de salida del análisis de ventana estacionaria falla. Esta situación es cuando una mano puede permanecer dentro de la ventana estacionaria moviéndose e introduciéndole una cantidad de movimiento considerable, mientras que la otra se desplaza rápidamente hacia alguna región lo suficientemente lejana para no estar en oclusión. Para solucionar este defecto, es que se cuenta con el tercer mecanismo de salida: si se vence el tiempo máximo de tolerancia y en el MGMB se encuentran dos *clusters* diferenciados, es posible terminar con la situación de oclusión.

Capítulo 5

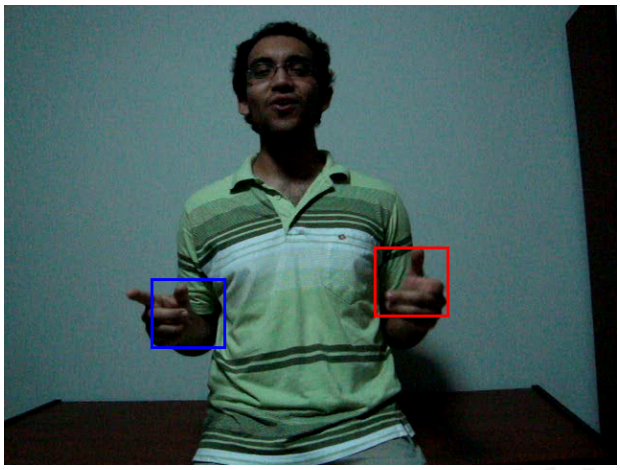
Experimentación y Resultados

El algoritmo realiza un seguimiento determinístico mediante el uso exlusivo de distintivos de movimiento con el objetivo de ubicar las manos dentro del cuadro actual con una precisión y un tiempo de procesamiento adecuados para las implementaciones en computadores y mini-computadores. En [34] se encuentra que 10 fps es lo necesario para el reconocimiento humano visible de señas, por lo tanto, se plantea en esta tesis tener dicha tasa como cota inferior a alcanzar y superar en lo posible.

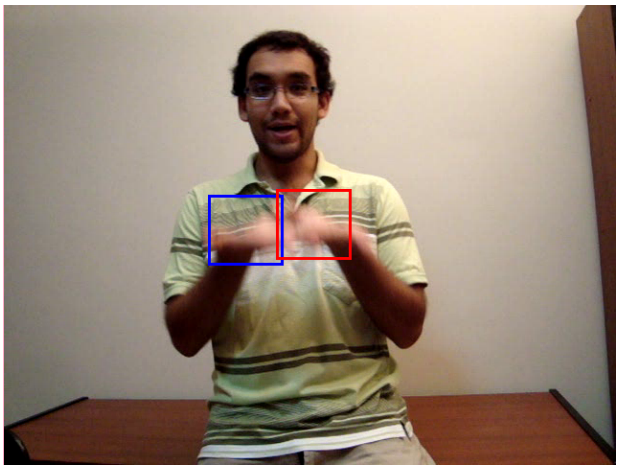
La primera implementación preliminar del algoritmo se hizo en el paquete científico de Matlab, con el objetivo de servir como una etapa de prueba de conceptos en la cual se pudiese verificar directamente el desarrollo algorítmico del seguimiento de manos desarrollado. Esta implementación se desarrolló en una computadora personal o *desktop* cuyas características se encuentran en el cuadro 5.1. El algoritmo se probó con con tres videos de prueba de 37 segundos de duración en la resolución de 640x480 píxeles, e identificados como 3078, 3083 y 3114. En la figura 5.1 se puede apreciar un cuadro representativo por cada uno de estos videos. En cada video se contiene al usuario con un polo o camisa de textura considerable, mostrando completamente el torso, cabeza y extremidades superiores; dentro de un contexto de movimiento libre de manos, brazos y cabeza en un entorno de conversación gestual. En las pruebas de video, el usuario no usa ropa de mangas largas, lo cual es problemático si se usaran distintivos de color, pues los brazos son elementos distractores. Se obtuvo un tiempo de procesamiento promedio por cuadro de 7.60 segundos, un tiempo no apto para aplicaciones en tiempo real. Por este motivo, en la continuación de esta sección, el algoritmo es implementado bajo el lenguaje C y usando la librerá gratuita ”ffmpeg” [75].

5.1. Evaluación de la velocidad de procesamiento

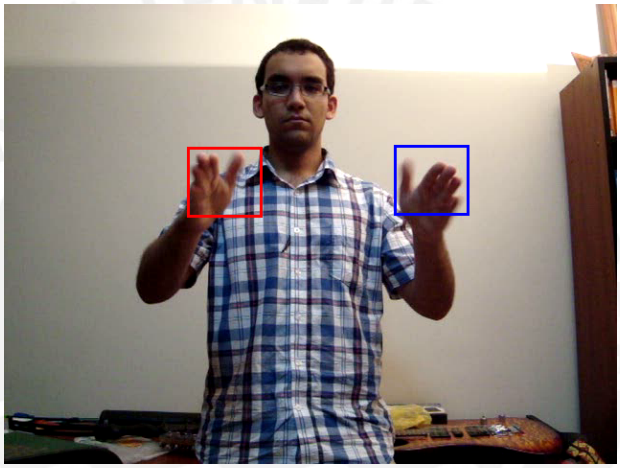
En primer lugar, se presentan los resultados de las implementaciones del algoritmo sobre los tres videos de prueba en términos del tiempo de procesamiento por cuadro, para las resoluciones de 640x480 y 320x240 píxeles en una computadora personal estándar en el mercado y la mini-computadora MK802. El tiempo de procesamiento por cuadro permite obtener la tasa de cuadros por segundo o fps promedio que sirve para caracterizar la velocidad de procesamiento del algoritmo. En el cuadro 5.1 se muestra una comparación técnica entre el computador y el mini-computador utilizados en la presente tesis según las especificaciones de sus procesadores y capacidades computacionales.



(a) Video 3078.



(b) Video 3083.



(c) Video 3114.

Figura 5.1: Videos de prueba.

	Desktop	**MK802**
Procesador	AMD Athlon II P320 Dual-Core	ARMv7 (Cortex-A8)
Reloj (GHz)	2.1	1.0
Memoria Caché	2x64KB I-caché 2x64KB D-caché 2x512KB L2 caché	32KB I-caché 32KB D-caché 256KB L2 caché
FLOPS promedio	1.10G	16.05M

Cuadro 5.1: Comparación técnica entre la computadora personal (*desktop*) y la mini-computadora MK802 según las especificaciones de sus procesadores y capacidades computacionales.

Respecto a la computadora personal, esta se basa en un procesador AMD Athlon^{TM} Dual Core de 2.1Ghz, 2GB de memoria RAM, en un S.O Linux Ubuntu 12.04 de 64 bits. Por otro lado, respecto a la mini-computadora MK802, esta se basa en un procesador Allwinner A10 1.0GHz Cortex-A8 (arquitectura ARM), 1GB de memoria RAM, en un S.O Linux Linaro 12.07.

Dentro del tiempo de procesamiento total del seguimiento por cuadro, es necesario saber cuál o cuáles

etapas toman mayor tiempo de procesamiento porque si se desea mejorar la velocidad del procesamiento, se debe empezar por modificar la implementación de dichas etapas: haciéndolas más eficientes o inclusive cambiando la naturaleza del procesamiento dentro de ellas. Con el objetivo de facilitar la exposición de los resultados obtenidos, se decidió dividir las etapas del algoritmo en dos conjuntos: se denota por ”P1” a la etapa de extracción del Mapa global de movimiento individual y el Mapa global de movimiento en bloques, y se denota por ”P2” al resto de etapas restantes del algoritmo (ver sección 3.2.2.).

5.1.1. Implementaciones para las resoluciones de 640x480 y 320x240 píxeles

Los estadísticas obtenidas de las mediciones realizadas de los tiempos de procesamiento por cuadro y el cálculo resultante de la tasa de fps asociado se observan en el cuadro 5.2 para ambas implementaciones hardware en ambas resoluciones de prueba.

Plataforma	Desktop	MK804
Tiempo mínimo (ms)	404.3	3.2e+003
Tiempo máximo (ms)	462.9	3.9e+003
Tiempo promedio (ms)	455.2	3.3e+003
Desviación estándar	3.2	79.0
fps promedio	**2.2**	**0.3**

(a) Resolución de 640x480 píxeles.

Plataforma	Desktop	MK804
Tiempo mínimo (ms)	75.8	521.8
Tiempo máximo (ms)	86.0	642.1
Tiempo promedio (ms)	76.1	531.9
Desviación estándar	0.5	15.0
fps promedio	**13.1**	**1.9**

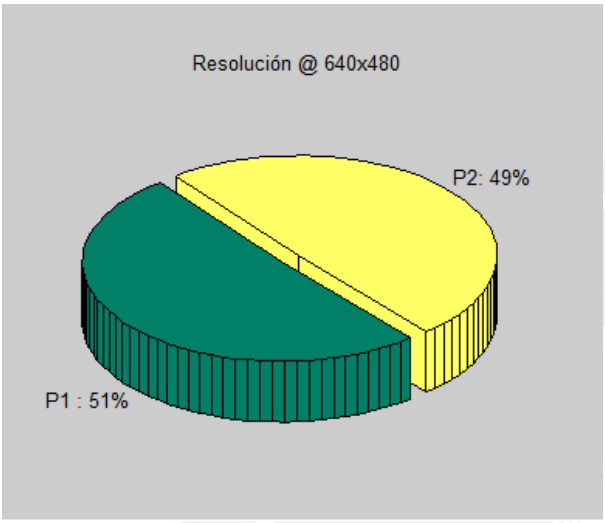
(b) Resolución de 320x240 píxeles.

Cuadro 5.2: Resultados de las distintas estadísticas obtenidas de los tiempos medidos de procesamiento por cuadro en las resoluciones de 640x480 y 320x240 píxeles, y los valores de la tasa de cuadros por segundo (fps) promedio asociada. La implementación *desktop* en la resolución de 320x240 px es la única en obtener una tasa de fps apta para aplicaciones en tiempo real.

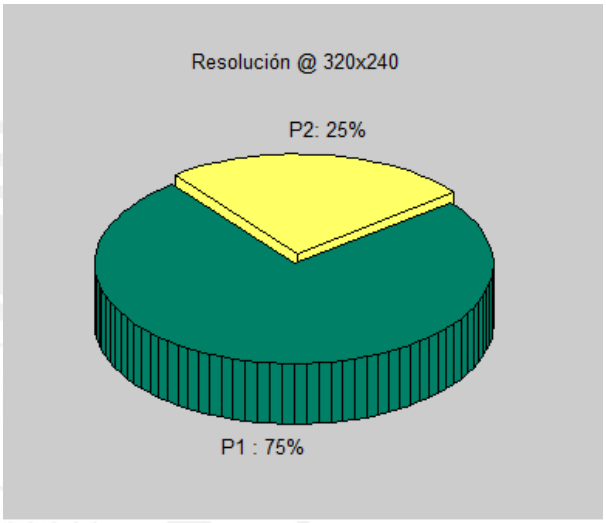
Se concluye que para una resolución de 640x480 píxeles no es posible poder realizar un seguimiento de manos en tiempo real para ninguna de las implementaciones realizadas en ambas plataformas hardware, pues no se alcanza la cota inferior de 10 fps planteada en la tesis. Sin embargo, para la resolución de 320x240 píxeles, se alcanza una tasa de 13.1 fps en la implementación sobre la computadora personal, la cual es apta para aplicaciones en tiempo real. Más aún, el hecho de tener una desviación estándar de 0.5 en los valores de fps, indica que se tiene una velocidad de procesamiento prácticamente constante por cuadro. La implementación en la mini-computadora MK802 no resultó apta para aplicaciones en tiempo real en ninguna resolución implementada.

Observar la figura 5.2 que muestra el tiempo promedio que ocupa P1 comparado a P2 en las secuencias de video para ambas resoluciones. La proporción promedio que ocupa P1 y P2 en relación con el tiempo total por cuadro es prácticamente idéntica entre las implementaciones de la computadora personal y el MK802. En la resolución de 640x480 píxeles, el tiempo tomado por P1 y P2 son prácticamente de igual valor: la mitad del tiempo de procesamiento total tomado por el algoritmo. En la resolución de 320x240 píxeles, el tiempo promedio que ocupa P1 es tres veces mayor al tomado por P2. Respecto a los distintivos usados (sección 3.2.2.), se puede analizar cuales de ellos demandan mayor complejidad computacional, y que por tanto necesitan una implementación más eficiente pare reducir el tiempo total de procesamiento tomado por el algoritmo. En el caso de P1, se puede identificar que la etapa de varianza local es la más notoria. En el

caso de P2, la etapa de generación del Mapa local de similitud tiene un orden de complejidad que depende de dos términos cuadráticos dependientes de la resolución del cuadro analizado (ver sección 3.2.2.2.). Por este motivo, P2 toma un mayor protagonismo en el tiempo tomado por el algoritmo al trabajar con mayor resolución, llegando a ocupar la mitad del tiempo en la resolución de 640x480 px, y tal vez sea inclusive necesario modificar algorítmicamente su naturaleza de procesamiento para reducir su costoso orden computacional. En conclusión, a menor resolución, la mejora sobre P1 es más crítica para alcanzar una mayor tasa de procesamiento, mientras que P2 será más importante al trabajar con mayor resolución de video.



(a) Diagrama para la resolución de 640x480 px.



(b) Diagrama para la resolución de 320x240 px.

Figura 5.2: Porcentaje del tiempo de procesamiento que ocupa P1 (extracción del Mapa global de movimiento individual y el Mapa global de movimiento en bloques) respecto al resto de etapas del algorimo P2 para ambas resoluciones implementadas (ver sección 5.1.).

5.1.2. Comparación con otros trabajos

En el cuadro 5.3 se presenta una comparación respecto a la tasa de fps (velocidad de procesamiento) de la tesis propuesta respecto a otros trabajos actuales [65, 48, 46]. Sin embargo, en el análisis de estos resultados es importante considerar las caracterśticas del procesador sobre el cual se implementa el algoritmo de seguimiento: un procesador más moderno y de mayor reloj procesará la información con mayor rapidez. Por este motivo, en el cuadro 5.3 se especifican el modelo del procesador y la frecuencia del reloj del sistema usado. Todos los procesadores con los cuales se compara la presente tesis tienen mayor velocidad de reloj y dos de ellos son más modernos. De este modo, se observa que los trabajos [65, 48] tienen una mayor tasa de fps respecto al trabajo propuesto, con el trabajo [48] en un valor cercano al de la presente tesis. Por otro lado, el trabajo [46] muestra una tasa de fps elevado dentro de una resolución menor, pero el seguimiento está diseñado para una sola mano. Por último, es importante resaltar que se ha podido implementar un seguimiento en tiempo real en una computadora de menor capacidad de procesamiento, y es de esperar que al implementarse en un procesador más moderno como el de los trabajos comparados, se puedan obtener mejores valores respecto a ellos.

	Propuesto	Trabajo [65]	Trabajo [48]	Trabajo [46]
fps @ 240x180 px	–	–	–	**35.71**
fps @ 320x240 px	**13.10**	–	**14.20**	–
fps @ 640x480 px	**2.20**	**(15.00 – 25.00)**	–	–
Procesador	AMD Athlon II P320	Intel Core i5	Pentium Quadcore	Pentium
Reloj (GHz)	**2.1**	**3.1**	**2.53**	**2.4**
RAM (GB)	3.6	8	–	–
Año de publicación	2013	2012	2012	2006
Ambas manos	**Sí**	**Sí**	**Sí**	**No**
Color/Movimiento	M	C/M	C	C/M

Cuadro 5.3: Comparación respecto a la tasa de procesamiento (fps). Resaltar la penúltima fila del cuadro que se refiere a si el algoritmo del trabajo analizado realiza un seguimiento sobre ambas manos del usuario, y la última fila que se refiere a si se usan distintivos de color (C) y/o movimiento (M)

5.2. Evaluación de la precisión o exactitud del seguimiento

Los algoritmos de seguimiento de manos cuentan con dos métodos principales para medir la precisión de la ubicación de la mano a lo largo de un video, las cuales necesitan del establecimiento manual *a priori* del *ground truth* de la posición de la mano en cada uno de sus cuadros. Ambos métodos requieren medir la distancia euclidiana entre la posición *detectada* y la posición *verdadera* de la mano, la cual es denominada *error en píxeles* por cuadro [12, 48].

El primer método consiste en contar todos los cuadros del video cuyo error en píxeles por cuadro sea menor a un valor establecido según los requerimientos de aplicación del seguimiento, y luego dividirlo entre el número de cuadros total del video para obtener un porcentaje o *ratio* del video con detección adecuada [12].

El segundo método consiste en usar una métrica que mida el error de detección de la mano por todo el video analizado, y que por lo tanto, se exprese en píxeles. Se usa la medida de error RMS [48], la cual consiste en promediar el error en píxeles de todos los cuadros del video analizado. Mientras menor sea el error RMS, más preciso es el seguimiento en promedio. Este es el método de evaluación que se usa en la presente tesis.

5.2.1. Implementación en base de datos de prueba

Para evaluar la precisión o exactitud del seguimiento se hizo uso de la base de datos gratuita llamada ”Intelligent Biometric Group Hand Tracking Database” [76], en la cual se usaron 35 videos de seguimiento de diferentes movimientos y condiciones de iluminación (entorno), como también con distintos usuarios que pueden variar en color de piel, ropa y poses frente a la cámara. Solo se utilizaron los videos que cumplen con las restricciones y condiciones para el uso del algoritmo de seguimiento de manos propuesto en la presente tesis (ver sección 3.1.1.). Realmente no interesa en qué plataforma, computadora personal o mini-computadora, se implementen estas pruebas, pues lo único que se desea evaluar es la precisión del algoritmo: se decidió implementarlo en la computadora personal.

Las ventajas de usar una base de datos de prueba es que brinda la cualidad de imparcialidad a la evaluación, además de brindar situaciones de evaluación que no están en exclusivas condiciones de laboratorio. Por

último, esta base de datos provee al investigador de información del *ground truth* de la verdadera posición de la mano en los cuadros de todos los videos.

5.2.2. Comparación con otros trabajos

En el cuadro 5.4 se presenta una comparación de la precisión o exactitud del seguimiento de la tesis propuesta respecto a otros trabajos actuales [65, 46, 48, 77]. En el caso de los trabajos [65, 46], se presentan dos implementaciones: el propuesto por los mismos autores y otro adicional que ellos presentan en la misma publicación . El error RMS es una cantidad que tiene sentido si se la relaciona con el tamaño de la mano: un algoritmo puede tener un error RMS mayor respecto a otro trabajo, pero tal vez el tamaño del cuadro y de la mano tienen dimensiones mayores que justifiquen un mayor error. Entonces, debido a que no existen al menos dos trabajos presentados que usen los mismos videos de prueba, se coloca en el cuadro 5.4 la resolución de los videos de prueba y el largo promedio de la mano dentro de los videos. Así, en la quinta fila del cuadro 5.4 se obtiene un *ratio de error* resultante de dividir el error RMS entre el largo promedio de la mano, y este valor sirve para comparar la precisión del seguimiento entre los distintos trabajos. Mientras menor sea este valor, más precisa es la detección. Por ejemplo, un error menor al 50 % significa que la distancia promedio entre la posición detectada y la verdadera posición de la mano es menor a la mitad de su longitud. Lamentablemente, no existe ningún consenso con respecto a que tan exacto o preciso debe ser el seguimiento de manos. Por lo tanto, se debería evaluar en aplicaciones específicas si el desempeño del algoritmo desarrollado es suficiente o no. Sin embargo, esto escapa del alcance de los análisis en la presente tesis, y las comparaciones expuestas se limitan a hacerse en base al ratio de error para todos los trabajos por igual.

Los trabajos [65, 46] obtienen un mejor desempeño medido por el ratio de error, llegando a tener una mejora en un factor de un poco más que dos con respecto al algoritmo propuesto. En principio, es esperable que métodos que emplean mayor cantidad de información, como lo es el uso de ambos tipos de características de color y movimiento, tengan un mejor desempeño que técnicas que emplean un solo subconjunto de la misma. Por tal motivo, es apreciable que los trabajos [65, 46] tengan un mejor performance en términos de exactitud que la presente tesis y el resto de trabajos [48, 77]. Sin embargo, cabe resaltar que los trabajos [65, 46] son los únicos en no emplear una bases de datos públicamente disponible en sus evaluaciones, y por lo tanto la comparabilidad de los resultados listados en el cuadro 5.4 es limitada. Además, este problema hace difícil el reportar qué situaciones *complejas* se presentaron en las pruebas de sus algoritmos de seguimiento. Para obtener una comparación más relevante, se debería realizar una evaluación de todos los algoritmos empleando la misma base de datos, o al menos bases de datos con características similares. Notar que el presente trabajo tiene un mejor seguimiento comparado a los trabajos [48, 77] y un ratio de error de exactitud menor al 50 %, con lo cual se cumple con el objetivo de haber desarrollado un algoritmo con una precisión de seguimiento tolerable. Por último, es importante resaltar que se obtuvo una desviación estándar de 4.38 con respecto al error RMS promedio de 19.21. Esto indica cierta robustez en el seguimiento, pues existe poca variación en los errores RMS determinados en cada cuadro respecto al valor medio.

	Propuesto	Trabajo [65]		Trabajo [46]		Trabajo [48]	Trabajo [77]
		Propuesto	OpenNI	Propuesto	Filtro de pártículas		
Resolución (px)	**352x288**	**640x480**		**240x180**		**320x240**	**320x240**
RMS (px)	**19.21**	**8.70**	**21.00**	**5.8**	**8.7**	**14.1**	**21.22**
Desviación estándar	4.38	4.90	12.20	–		–	–
Largo de la mano (px)	46	49		35		15.00	38
Ratio de error	**41.45**	**19.18**	**52.44**	**16.57**	**24.86**	**94.00**	**55.84**
fps	13.10	(15.00 – 25.00)	–	35.71	15.87	14.20	–
Año de publicación	2013	2012		2006		2012	2012
Ambas manos	**Sí**	**Sí**		**No**		**Sí**	**No**
Color/Movimiento	M	C/M		C/M		C	M

Cuadro 5.4: Comparación respecto al error RMS. Resaltar la penúltima fila del cuadro que se refiere a si el algoritmo analizado realiza un seguimiento sobre ambas manos del usuario, y la última fila que se refiere a si se usan distintivos de color (C) y/o movimiento (M)

5.3. Problemas y posibles modificaciones al sistema

Las implementaciones realizadas del algoritmo ha permitido identificar de manera más precisa a las causas de los problemas que originan una mala ubicación de las manos o pérdidas de seguimiento, muy relacionados a los movimientos raudos de las manos y la ganancia de protagonismo de movimiento que pueden alcanzar otros elementos como la ropa, codos, brazos y cabeza del usuario. Si bien es posible recuperarse de una pérdida de seguimiento en las condiciones de operación adecuadas, se trata de disminuir su frecuencia de incidencia. El análisis de estos problemas y el planteamiento de sus soluciones ayudan a entender mejor el alcance y limitaciones del algoritmo. Toda modificación y/o mejora planteada debe mantener la cualidad de no requerir ninguna etapa de entrenamiento previo *offline*, ni la calibración de algún parámetro adicional por parte del usuario en la inicialización, como tambieén el mantener una carga de procesamiento adecuada para una implementación *online*.

El algoritmo posee un estilo de orden y codificación modular que permite una adecuada expansión de cada una de sus partes, y por tanto, la fácil implementación de mejoras dentro de ella. De este modo, el algoritmo se comporta como una plantilla de diseño, en donde lo invariante dentro de ella es la complementariedad del análisis sin y con información *a priori*, junto con el manejo de la oclusión y la recuperación de pérdidas de seguimiento. Por otro lado, lo mutable dentro del algoritmo es el contenido de dichas etapas: los distintivos usados y la articulación que puede existir entre ellos para dar soporte a las distintas etapas del algoritmo.

Una posible mejora general es la inclusión de un *framework* predictivo (probabilístico) dentro del seguimiento que permita obtener la ubicación de las manos en base a estimaciones y a un modelamiento dinámico de ellas, pero que además utilice a los distintivos descritos en la tesis u otros adicionales. Por ejemplo, se pueden emplear filtros de Kalman o filtros de partículas. Sin embargo, es importante que estas mejoras mantengan un tiempo de procesamiento adecuado para aplicaciones *online* [46], especialmente crítico para sistemas de entornos computacionales limitados que son de interés en la presente tesis.

En la presente sección se exponen los problemas encontrados en las partes inmutables del algoritmo, y cómo posiblemente solucionarlos mediante mejoras dentro del contenido de ellas. Estas mejoras son resumidas posteriormente en la sección de recomendaciones en la presente tesis. Finalmente, los resultados del cuadro 5.4 sugieren que la inclusión del uso de distintivos de color pueda mejorar el seguimiento, lo cual es

de esperar, puesto que brindan mayor información acerca de las manos. Por este motivo, su uso es incluido en varias de las mejoras propuestas; sin embargo, entender que su uso está basado en algún modelo del color actual de la mano del usuario establecido debidamente durante la inicialización y con la posibilidad abierta de poder ir actualizándose adaptativamente durante el transcurso del seguimiento. La inclusión de distintivos de color debe ser de carácter adicional y accesorio, pues lo principal como caracterización de la mano deben ser los distintivos de movimiento por las ventajas expuestas a lo largo de la tesis.

5.3.0.1. Con respecto al desplazamiento global

- **Errores de ubicación sobre elementos en movimiento**

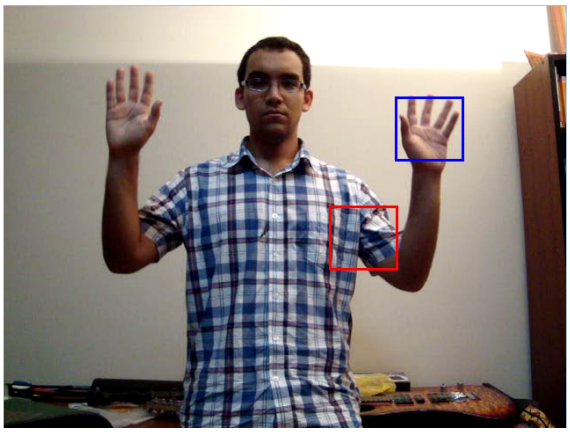
El desplazamiento global puede fallar al detectar las regiones pertenecientes a los codos, brazos o la cabeza del usuario como si fueran las manos debido a que presentaron una cantidad de movimiento considerable dentro del Mapa global de movimiento en bloques. Inclusive puede ubicarse sobre regiones que no tengan color de piel, como puede ser la ropa del usuario.

Modificación y posible solución

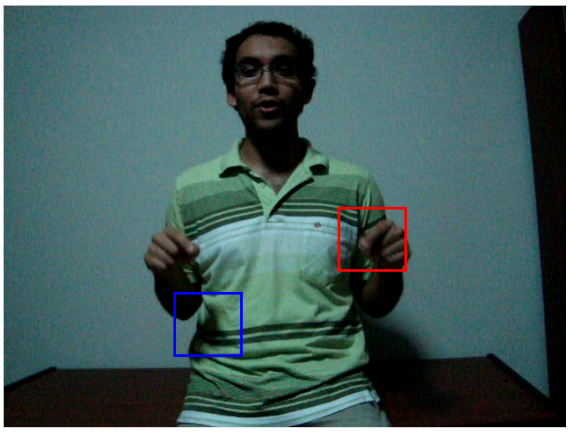
- Se plantea usar *información temporal* con la implementación de algún *mecanismo temporal de rechazo*. Este mecanismo divide al cuadro en secciones de bloques estándares (similar al MGMB) y por cada región dividida se realiza un conteo estadístico según cuantas veces se han detectado las manos dentro de ella durante algún determinado intervalo de tiempo previo. De acuerdo a este conteo, se asigna un valor proporcional de probabilidad a cada una de estas regiones que sirve como referencia para saber cuáles de ellas han tenido mayor presencia de las manos anteriormente y formar así un *Mapa global de conteo estadístico* (MGCE). De este modo, cuando se realice un desplazamiento global, cada posición del MGCE puede servir como factor de peso multiplicador o máscara del MGMB, de modo que puedan rechazarse zonas con baja probabilidad de encontrar a las manos, pero que ocasionalmente obtuvieron una cantidad de movimiento considerable por el movimiento de algún elemento indeseado.

- Mediante el uso de características de color se plantea el formular un *Mapa global de color de piel individual* (MGCPI), en el cual se segmenten aquellos píxeles dentro del cuadro actual cuyo color tenga una considerable similitud respecto al color de la piel; y en base a este formular también un Mapa global de color de piel en bloques (MGCPB) análogo a como se hizo con el MGMI. Esto puede servir de dos formas distintas:

  - En el momento del desplazamiento global, cada posición del MGCPB puede servir como un factor de peso multiplicador o máscara hacia el MGMB, de modo que solo se pueda ubicar a la mano en las regiones que tengan simultáneamente movimiento y gran similitud de color de piel. Esto evita el desplazamiento global hacia zonas con movimiento de elementos con color distinto al de la piel.
  - El MGCPB puede utilizarse conjuntamente con el mecanismo temporal de rechazo descrito anteriormente, de modo que permita detectar qué regiones de color de piel en los cuadros anteriores



(a) Cuadro del video de prueba 3114. Un recuadro se ubica incorrectamente sobre la axila del usuario debido a que este presenta un elemento de gran textura gracias a la ropa del usuario.



(b) Cuadro del video de prueba 3078. Un recuadro se ubica incorrectamente cerca al codo del usuario debido al gran contraste de intensidad que existe entre su sombra respecto al entorno y la ropa del usuario.

Figura 5.3: Dos cuadros de distintos videos de prueba en donde se aprecia problemas con el desplazamiento global relacionado a errores de ubicación sobre elementos en movimiento debido a algún elemento ajeno a las manos que ha generado una cantidad de movimiento considerable en el cuadro.

no han presentado cantidades de movimiento considerables. Esto permite detectar, por ejemplo, la cabeza del usuario y usar esta información para su posible rechazo de ubicación en el desplazamiento global.

- Se puede reducir la búsqueda de la posible ubicación de la mano dentro del desplazamiento global, a favor de una región de análisis limitada por ciertas restricciones antropomorfas relacionadas a las posiciones anteriores de la mano. De este modo, se evitan ubicaciones lejanas que puedan resultar inverosímiles para ubicar la mano, bajo la premisa de que luego de detectarse una pérdida de seguimiento, la posición verdadera de la mano no puede ser más lejana a lo que el brazo pudo haber desplazado a la mano.
- Se puede incluir etapas de operaciones morfológicas sobre el MGMI. Estas operaciones pueden unir píxeles con cantidad de movimiento considerable para que formen *clusters* más grandes o elementos cerrados de mayor área. Como resultado, se refuerza aún más el protagonismo de movimiento sobre aquellas regiones que ya la tenían desde antes de dichas operaciones.

- **No se identifican a ambas manos luego de un desplazamiento global**

En el desplazamiento global de ambas manos puede ocurrir que la primera de ellas es ubicada correctamente, mientras la posición de la otra mano no es alterada. Conforme se analizado en la sección 3.2.2.1. , esto se debe a que en el MGMB no se *observan* dos *clusters* diferenciados, sino que en realidad, puedan haber dos existentes por cada mano pero que están desafortunadamente unidos por al menos un bloque de valor no nulo.

Modificación y posible solución

Una posible solución es la aplicación de una etapa adicional de segmentación sobre el MGMB, con la particularidad de que los bloques que estén por debajo del valor umbral tengan valor nulo, y aquellos que no, mantengan su valor sin alteración. Como resultado, los *clusters* se vuelven más compactos y se eliminan de aquellos bloques de valores despreciables. Esta misma solución se puede emplear sobre el MGCE y MGCPB expuestos anteriormente.

5.3.0.2. Con respecto al desplazamiento local

- **Errores por *drift* en la ubicación local de la mano**

En el desplazamiento local existen casos en el cual se origina una *drift* de la ubicación de la mano hacia posiciones localmente cercanas que no correspondan a la mano misma, pero a cualquier otro elemento ajeno con movimiento dentro del MLS (ver sección 3.2.2.2.).

Modificación y posible solución

- Luego de obtener las posición de la mano por desplazamiento local, se puede analizar dentro de una ventana cuadrada alrededor de ella, cual es la posición cuyo vecindario tenga un color con mayor similitud al de la piel para ubicar en ella la nueva posición de la mano. De este modo, los distintivos de color sirven para refinar la ubicación obtenida por desplazamiento local. Una mejora adicional es definir que el tamaño del lado de la ventana sea proporcional a la magnitud del desplazamiento local: recordar que un mayor movimiento, implica una mayor pérdida de la calidad del color en la mano y por lo tanto es menos confiable. De este modo, si hubo un mayor desplazamiento local, el tamaño del lado de la ventana se reduce porque se confía menos en el color; de lo contrario, aumentar el lado de la ventana si el desplazamiento fue menor. Por último, considerar que esta solución puede limitar el alcance del desplazamiento máximo permitido entre cuadros consecutivos.

- En el momento de generar el MLS, puede además generarse un mapa que cuente la cantidad de ”1’s” subyacentes en las posiciones del MGCPI dentro de cada bloque del MLS. Esta cantidad contada puede procesarse para ser usada como peso o factor multiplicador en el cálculo del centroide del MLS para dar mayor protagonismo a aquellas regiones de bloques que tengan además un color más similar al de la mano.

NOTA: En el caso del desplazamiento local, es importante que el uso de distintivos de color deba ser accesorio y complementario porque también pueden ocurrir problemas de *drift* o desviaciones en la detección hacia otras regiones con color similar al de la piel (brazos, cabeza, etc.).

5.3.0.3. Con respecto al seguimiento en general

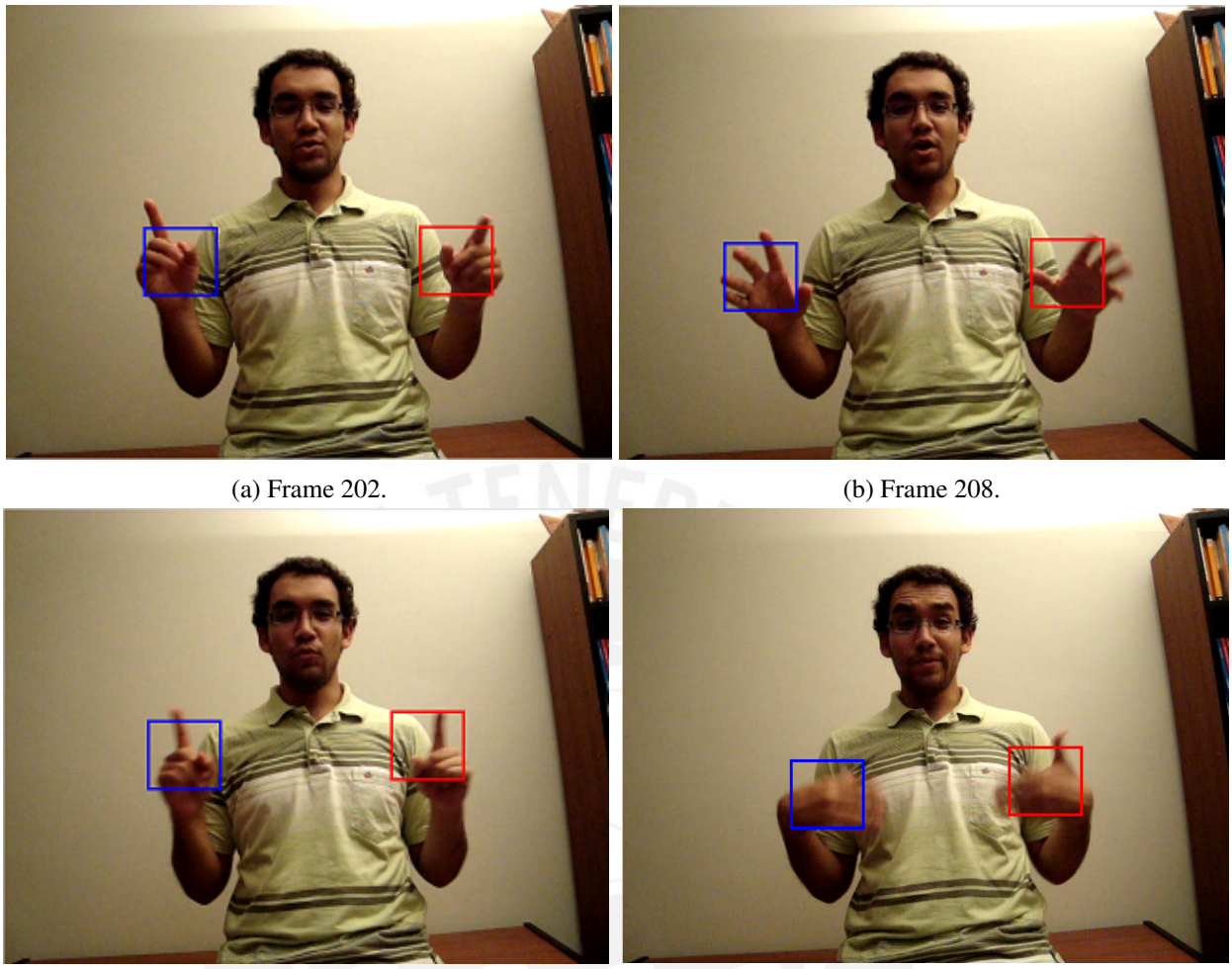
- **Problema con movimientos dentro de la ventana estacionaria durante la oclusión**

Durante la oclusión, el mecanismo de salida del tiempo máximo de tolerancia permite resolver los casos problemáticos en los cuales alguna mano introduce cierta cantidad de movimiento considerable dentro de la ventana estacionaria, a pesar de que estas podrían no seguir en oclusión (ver sección 4.1.2.3.). Sin embargo,

para obtener un seguimiento más fluido y menos erróneo, es importante el considerar reducir el índice de incidencia de este problema.

Modificación y posible solución

- Se puede modificar el modo de ubicar las posiciones de referencia en oclusión (*pro's*). Por ejemplo, el uso de algún método de extrapolación de posiciones que permita definir una pro más adecuada para casos en los que posiblemente las manos salgan de la oclusión en la misma trayectoria por la cual entraron en ella.
- Mejorar los mecanismos de salida con información del color de la mano. Por ejemplo, cuando se detecte que dentro de la ventana estacionaria exista una disminución considerable de píxeles de gran similitud al color de la piel (usando el MGCPI), se pueda entender que tal vez haya terminado la condición de oclusión.
- Un posible mecanismo de salida es almacenar la información de color que existe inicialmente dentro de la ventana estacionaria en algún modelo, e irla comparando en cada cuadro sucesivo con el modelo obtenido de la información contenida dentro de la ventana estacionaria en dicho cuadro posterior. Entonces, se debe terminar la condición de oclusión cuando exista una gran diferencia en la comparación, porque esto tal vez se debió a que las manos se retiraron de la ventana estacionaria y no siguen en oclusión. En este caso no es necesario conocer el color de las manos.



(a) Frame 202. (b) Frame 208.

(c) Frame 214. (d) Frame 220.

Figura 5.4: Secuencia de frames del video de prueba 3082 con las manos detectadas.

Conclusiones

De manera general, se ha logrado desarrollar e implementar una solución al seguimiento de manos capaz de ubicar a las manos en un tiempo de procesamiento y precisión adecuadas para aplicaciones en tiempo real dentro de una computadora personal bajo la hipótesis de que las manos son los elementos presentes con mayor movimiento, y con la potencialidad de también serlo en entornos con recursos computacionales más limitados. Además, la solución es paramatrizable para poder funcionar con distintas resoluciones de video; y también es sencilla de usar, pues tiene una inicialización intuitiva y no requiere alguna calibración adicional por parte del usuario.

Por otro lado, se pueden concluir los siguientes aspectos y objetivos más particulares:

- Bajo las condiciones en que las manos son los elementos de mayor movimiento en el video (ver sección 3.1.1.), se ha podido realizar un seguimiento de naturaleza determinística. Al mismo tiempo, estas condiciones han permitido resolver los problemas de oclusión mutua y pérdidas de seguimiento sin la necesidad de implementar un modelo dinámico previo de la mano o métodos de seguimiento probabilísticos más complejos.

- El seguimiento de manos se basa completamente en el uso de distintivos de movimiento para caracterizar a las manos y poderlas ubicar en el cuadro actual. Esto ha permitido que el seguimiento pueda realizarse sobre entornos con distintas condiciones de iluminación global y tonalidades de color en las manos.

- El algoritmo de seguimiento de manos desarrollado también constituye una plantilla modular. Lo inalterable dentro de esta plantilla es la forma cómo se relaciona el desplazamiento local con el global, es decir, el modo cómo se decide confiar o desconfiar del uso de la información *a priori* de la posición de la mano en el cuadro anterior. También es inalterable cómo se relacionan las etapas de detección de oclusión y recuperación de las pérdidas de seguimiento dentro del algoritmo. Sin embargo, los módulos modificables son los distintivos o cualquier tipo de información que permita caracterizar a las manos y que sean usados en cualquiera de las partes inalterables de la plantilla.

- Respecto a la velocidad del procesamiento, se concluye lo siguiente:

  - En la implementación sobre la computadora personal, se obtuvo una tasa promedio de 13.1404 fps para la resolución de 320x240 y 2.1970 fps para la resolución de 640x480. Esto significa un procesamiento en tiempo real para la resolución de 320x240, y que es posible incluir etapas posteriores de procesamiento para alguna otra aplicación particular, sin ocasionar que el video sea ilegible para el usuario en cuanto identificación visual de las seas representadas por sus manos por superarse la cota inferior de 10 fps [34].

  - En la implementación sobre el mini-computador MK802, se obtuvo una tasa promedio de 1.8798 fps para la resolución de 320x240 y 0.3067 fps para la resolución de 640x480. Este resultado restringe su uso para aplicaciones *off-line*.

- Respecto a la precisión o exactitud del seguimiento, se obtuvo un error RMS de 19.21 px para la resolución de 320x240 px, para un largo de la mano de 46.34 px y dentro de videos con distintas situaciones de iluminación y movimiento. Esto significa, que en promedio, existe un error de ubicación menor a la mitad del largo de la mano. Este es un valor aceptable de precisión, además que, al compararlo con otros cuatro trabajos actuales [65, 46, 48, 77], se ha logrado un mejor resultado comparado a dos de ellos.

- Ciertas partes demandantes computacionalmente del algoritmo, como son las convoluciones, tienen un diseño que las vuelve *cache aware* y que potencialmente podrán hacer uso de extensiones tales como SIMD entre otras para acelerar su procesamiento. En particular, la etapa de Varianza local, que según los experimentos tiene una demanda computacional y tiempo de procesamiento considerable dentro del algoritmo, se presta para una implementación más eficiente mediante el uso de instrucciones SIMD. Por otro lado, la etapa de generación del MLS, que tiene un orden de complejidad considerable, también tiene la posibilidad de modificarse para una mayor eficiencia. Como resultado, el algoritmo presentado tiene la posibilidad de ser más adecuada para entornos computacionales de recursos más limitados.

Recomendaciones

- La inclusión de distintivos de color dentro de la caracterización de la mano, y analizando cuanta carga computacional adicional involucraría su uso. Una posible opción es usarlos de manera auxiliar y solamente en situaciones puntuales donde resuelvan problemas presentes en el algoritmo desarrollado actualmente, pero siempre de forma que los distintivos de movimiento sigan siendo los principales a usar:
  - Se pueden resolver los problemas relacionados a situaciones donde exista poco o nulo movimiento de las manos, en la cual temporalmente se de preferencia a los distintivos de color para la ubicación de las manos sobre los distintivos de movimiento y así evitar pérdidas de seguimiento.
  - Se puede detectar la zona que corresponda a la cabeza del usuario (la cual tiene poco movimiento relativo comparado a las manos) y usar esta información como ayuda para ubicar a las manos por medio de alguna restricción de naturaleza anátomica o espacial.
- Incluir las siguientes mejoras para generar una implementación más robusta del algoritmo frente a los problemas que actualmente posee:
  - Implementar un mecanismo que permita adecuar al seguimiento frente a situaciones en donde la mano pueda haber cambiado de tamaño aparente frente a la cámara, además de adaptar el tamaño de la ventana de detección a las nuevas dimensiones de la mano.
  - Implementar un mecanismo que permita considerar mayor información temporal o un historial pasado de posiciones de las manos para generar alguna decisión respecto a la validez de la posición determinada de la mano en el cuadro actual o predecir algún estimado de la posición actual que sirva para corregir alguna pérdida de seguimiento o situación problemática diversa.
  - Acotar la búsqueda de la posible ubicación de la mano dentro del desplazamiento global, a favor de una región de exploración limitada por ciertas restricciones antropomorfas relacionadas a las posición anterior de la mano. Esto permite una aceleración del algoritmo.
- Realizar una implementación usando extensiones como SIMD u otras que permitan implementar más eficientemente las etapas del algoritmo relacionadas a la extracción de distintivos, aprovechando la caracterśtica *cache aware* que se posee. Si se desea aumentar etapas de procesamiento relacionados a distintivos adicionales, procurar que sigan manteniendo este tipo de diseño eficiente.

Bibliografía

[1] N. B. Bo, M. N. Dailey, and B. Uyyanonvara, "Robust hand tracking in low-resolution video sequences," in *Proceedings of the third conference on IASTED International Conference: Advances in Computer Science and Technology*, ser. ACST'07. ACTA Press, 2007, pp. 228–233.

[2] M. Donoser and H. Bischof, "Real time appearance based hand tracking," in *Proceedings of the 19th International Conference on Pattern Recognition*, ser. ICPR'08. IEEE, 2008, pp. 1–4.

[3] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, vol. 14, no. 1, pp. 30–39, Jan. 1994.

[4] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," in *Proceedings of the 12th international conference on Human-computer interaction: intelligent multimodal interaction environments*, ser. HCI'07. Springer-Verlag, 2007, pp. 728–737.

[5] T. Lee and T. Höllerer, "Hybrid feature tracking and user interaction for markerless augmented reality," in *Proceedings of the 2008 IEEE Virtual Reality Conference*, ser. VR '08. IEEE, 2008, pp. 145–152.

[6] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proceedings of the 6th European Conference on Computer Vision*, ser. ECCV '00, vol. 2, no. 1843. Springer-Verlag, 2000, pp. 3–19.

[7] Q. Yuan, S. Sclaroff, and V. Athitsos, "Automatic 2d hand tracking in video sequences," in *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, ser. WACV-MOTION '05, vol. 1. IEEE Computer Society, 2005, pp. 250–256.

[8] OMRON Corp., "Recognizing motion of faces and people," 2013, página de la empresa OMRON acrca de su aplicación para dispositivos móviles que permite el reconocimiento del movimiento del rostro y manos. [Online]. Available: http://www.omron.com/ecb/products/mobile/okao02.html

[9] GestureTek Corp., "3d depth camera hand, face and body tracking solutions," 2013, página de la empresa GestureTek sobre el sistema de *tracking* 3D que ofrecen para seguir el movimiento de manos, rostro y torso del usuario. [Online]. Available: http://www.gesturetek.com/3ddepth/introduction.php

[10] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. Springer-Verlag New York, Inc., 2010.

[11] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Addison-Wesley Longman Publishing Co., Inc., 2001.

[12] S. Ongkittikul, S. Worrall, and A. Kondoz, "Two hand tracking using colour statistical model with the k-means embedded particle filter for hand gesture recognition," in *Proceedings of the 7th Computer Information Systems and Industrial Management Applications*, ser. CISIM '08. IEEE Computer Society, 2008, pp. 201–206.

[13] T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank, *ACM SIGCHI Curricula for Human-Computer Interaction*. ACM, 1992.

[14] S. C. W. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873–891, Jun. 2005.

[15] R. Y. Wang and J. Popović, "Real-time hand tracking with a color glove," in *ACM Transactions on Graphics*, ser. SIGGRAPH '09, no. 63. ACM, 2009, pp. 1–8.

[16] AugmenteDev Corp., "Augment 3d system," 2013, página de la aplicación para dispositivos móviles Augment que permite interacciones de Realidad Aumentada. [Online]. Available: http://www.augmentedev.com

[17] P. Mistry and P. Maes, "Sixthsense: a wearable gestural interface," in *ACM Transactions on Graphics*, ser. SIGGRAPH ASIA '09, vol. 11. ACM, 2009, pp. 1–1.

[18] G. C. Burdea and P. Coiffet, *Virtual Reality Technology*, 2nd ed. John Wiley & Sons, Inc., 2003.

[19] Advance Real-Time System, "The motion capture system," 2013, página de una empresa alemana que ofrece soluciones de hand tracking orientados a aplicaciones médicas, como la captura de movimiento para mediciones de la ergonomía humana, etc. [Online]. Available: http://www.ar-tracking.com/applications/motion-capture

[20] Play Station, "Playstation eye brings next-generation communication to playstation 3," 2007, página acerca del desarrollo del producto Playstation Eye para la consola de juego Playstation. [Online]. Available: http://us.playstation.com/corporate/about/press$-$release/396.html

[21] Xbox, "Xbox 360 + kinect," 2013, página que introduce el producto Kinect en conjunto con su uso en la consola de juego Xbox 360. [Online]. Available: http://www.xbox.com/en-US/KINECT

[22] X. Live, "Componentes del sensor de kinect," 2013, página que menciona y explica los distintos componentes que constituyen el sensorintroduce del producto Kinect de Xbox 360. [Online]. Available: http://support.xbox.com/es-ES/xbox-360/kinect/kinect-sensor-components

[23] Kinect for Windows, "Developer solutions for the kinect sensor windows user," 2013, página donde es posible obtener el SDK para manejar el sensor Kinect con el SO Windows y acceder a la documentación respectiva. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx

[24] V. Frati and D. Prattichizzo, "Using kinect for hand tracking and rendering in wearable haptics," in *Proceedings of the 2011 IEEE World Haptics Conference*, ser. WHC'11. IEEE Computer Society, 2011, pp. 317–321.

[25] Leap Motion Inc., "Leap motion," 2013, página de la empresa Leap motion en la cual expone su principal producto del mismo nombre, el cual consiste en un hardware dedicado para la adquisicón del movimiento de las manos y conforme a un software, poder tomar acciones que permitan la interaccón entre la persona y el computador. [Online]. Available: https://www.leapmotion.com

[26] R. V. Babu, P. Pérez, and P. Bouthemy, "Robust tracking with motion estimation and local kernel-based color modeling," in *Proceedings of the 2005 International Conference on Image Processing*, ser. ICIP'05, vol. 1, no. 8. IEEE, 2005, pp. 717–720.

[27] C. Malerczyk, "Interactive museum exhibit using pointing gesture recognition," in *Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2004*, ser. WSCG'04, 2004, pp. 165–172.

[28] LM3LABS Corp., "Interactive museum technologies," 2013, página de la empresa LM3LABS que brinda servicios de implementación de sistemas interactivos para museos, los cuales están basados en teconologás de Visión por Computadora. [Online]. Available: http://www.lm3labs.com/museum/technologies

[29] Agilence Inc., "Agilence auditories solutions," 2013, página de la empresa Agilence que ofrece servicios de auditoría y capacitación sobre seguridad en los establecimientos de trabajo. [Online]. Available: http://www.agilenceinc.com

[30] StopLift Checkout Vision Systems Co., "StopLift Checkout Vision Systems: ScanItAll," 2013, página de la empresa Stoplift en la cual se expone su producto estrella ScanItAll para la detección de distinas modalidades de robo o pérdidas de productos en autoservicios por medio de la aplicación de técnicas de Visión por Computadora. [Online]. Available: http://www.stoplift.com

[31] StopLift Co., "ScanItAll: How it works. Rocket Science for Loss Prevention," Jun. 2013, página de la empresa Stoplift en la cual se expone de manera detallada el completo funcionamiento del producto ScanItAll. [Online]. Available: http://www.stoplift.com/how-it-works

[32] H. Trinh, Q. Fan, J. Pan, P. Gabbur, S. Miyazawa, and S. Pankanti, "Detecting human activities in retail surveillance using hierarchical finite state machine," in *2011 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ser. ICASSP'11. IEEE, 2011, pp. 1337–1340.

[33] B. S. Parton, "Sign language recognition and translation: A multidisciplined approach from the field of artificial intelligence," *Journal of Deaf Studies and Deaf Education*, pp. 94–101, 2006.

[34] T. Starner, A. Pentland, and J. Weaver, "Real-time american sign language recognition using desk and wearable computer based video," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12. IEEE Computer Society, Dec. 1998, pp. 1371–1375.

[35] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, ser. GW '99. Springer-Verlag, 1999, pp. 103–115.

[36] S. Lee, V. Henderson, H. Hamilton, T. Starner, H. Brashear, and S. Hamilton, "A gesture-based american sign language game for deaf children," in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '05. ACM, 2005, pp. 1589–1592.

[37] Georgia Tech University, "Center for accessible technology in sign," 2010, página del proyecto entre la Escuela para sordos de Atlanta y el Georgia Institute of Technology con el fin de desarrollar sistemas computacionales que permitan mejorar el aprendizaje del lenguaje de seas para los nios sordos de esta escuela. [Online]. Available: http://www.cats.gatech.edu

[38] C. von Hardenberg and F. Bérard, "Bare-hand human-computer interaction," in *Proceedings of the 2001 workshop on Perceptive user interfaces*, ser. PUI '01. ACM, 2001, pp. 1–8.

[39] J. J. Wang and S. Singh, "Video analysis of human dynamics - a survey," *ELSEVIER Real Time Imaging*, vol. 9, pp. 321–346, 2003.

[40] Y. Wu, J. Lin, and T. Huang, "Capturing natural hand articulation," in *Proceedings of the Eighth IEEE International Conference on Computer Vision*, ser. ICCV '01, vol. 2. IEEE, 2001, pp. 426–432.

[41] J. M. Rehg and T. Kanade, "Visual tracking of high dof articulated structures: an application to human hand tracking," in *Proceedings of the Third European Conference on Computer Vision*, ser. ECCV '94, vol. 2. Springer–Verlag, 1994, pp. 35–46.

[42] M. de La Gorce, N. Paragios, and D. J. Fleet, "Model-based hand tracking with texture, shading and self-occlusions," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'08. IEEE Computer Society, 2008, pp. 1–8.

[43] J. Rehg and T. Kanade, "Digiteyes: Vision-based hand tracking for human-computer interaction," in *Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*. IEEE Computer Society, 1994, pp. 16–22.

[44] M. Strring, T. Moeslund, Y. Liu, and E. Granum, "Computer vision-based gesture recognition for an augmented reality interface," in *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, 2004, pp. 766–771.

[45] H. Fei and I. Reid, "Probabilistic tracking and recognition of non-rigid hand motion," in *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, ser. AMFG '03. IEEE Computer Society, 2003, pp. 60–67.

[46] C. Shan, Y. Wei, and T. Tan, "Real-time hand tracking using a mean shift embedded particle filter," in *Pattern Recognition*. Elsevier Science Inc., 2007, pp. 1958–1970.

[47] A. Sen, I. Cheng, and A. Basu, "Hand and face tracking under occlusion with anthropomorphic constraints," in *Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics*, ser. SMC'12. IEEE, 2012, pp. 242–247.

[48] H. Trinh, Q. Fan, P. Gabbur, and S. Pankanti, "Hand tracking by binary quadratic programming and its application to retail activity recognition," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '12. IEEE Computer Society, 2012, pp. 1902–1909.

[49] F. Khan and S. Baset, "Real-time human motion detection and classification," in *Proceedings of the 2002 IEEE Students Conference*, ser. ISCON'12, vol. 1. IEEE, 2002, pp. 135–138.

[50] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, and H. Ney, "Tracking using dynamic programming for appearance-based sign language recognition," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, ser. FGR '06. IEEE Computer Society, 2006, pp. 293–298.

[51] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking multiple and compound visual objects," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ser. PAMI'01, vol. 23, no. 6. IEEE Computer Society, Jun. 2001, pp. 560–576.

[52] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proceedings of the GraphiCon 2003*, 2003, pp. 85–92.

[53] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, Jan. 2002.

[54] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *Proceedings of the 9th European conference on Computer Vision*, ser. ECCV'06, vol. 2. Springer-Verlag, 2006, pp. 589–600.

[55] T. Sayantan, P. Sayantanu, M. Ankur, and D. Swagatam, "Face detection using skin tone segmentation," in *Proceedings of the 2011 World Congress on Information and Communication Technologies*, ser. WICT'11. IEEE, 2011, pp. 53–60.

[56] M. Donoser and H. Bischof, "ROI-SEG: Unsupervised color segmentation by combining differently focused sub results." in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'07. IEEE Computer Society, 2007, pp. 1–8.

[57] M. Kolsch and M. Turk, "Fast 2D hand tracking with flocks of features and multi-cue integration," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, ser. CVPRW '04, vol. 10. IEEE Computer Society, 2004, pp. 158–.

[58] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[59] J. Martin, V. Devin, and J. L. Crowley, "Active hand tracking," in *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, ser. FG'08. IEEE Computer Society, Apr. 1998, pp. 573–578.

[60] S. Lee, A. Sang, and K. Hyoung, "Mawupc algorithm for tracking face and hands in complex background," in *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, ser. ISPA'01. IEEE, 2001, pp. 295–300.

[61] K. Barhate, K. Patwardhan, S. Roy, S. Chaudhuri, and S. Chaudhury, "Robust shape based two hand tracker," in *Proceedings of the 2004 International Conference on Image Processing*, ser. ICIP'04, vol. 2. IEEE, 2004, pp. 1017–1020.

[62] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'00, vol. 2. IEEE Computer Society, 2000, pp. 142–149.

[63] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, vol. 1, pp. 1–15, 1998.

[64] K. Hyejin, K. Keun-Chang, and L. Jaeyeon, "Fast 2D both handstracking with articulate motion prediction," in *Proceedings of the 8th International Conference Advanced Communication Technology*, ser. ICACT'06, vol. 1. IEEE, 2006, pp. 242–248.

[65] B.-J. Chen, C.-M. Huang, T.-E. Tseng, and L.-C. Fu, "Robust head and hands tracking with occlusion handling for human machine interaction," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, ser. IROS'12. IEEE, 2012, pp. 2141–2146.

[66] W. Du and J. Piater, "Hand modeling and tracking for video-based sign language recognition by robust principal component analysis," in *Proceedings of the 11th European conference on Trends and Topics in Computer Vision*, ser. ECCV'10, vol. 1. Springer-Verlag, 2012, pp. 273–285.

[67] G. Welch and G. Bishop, "An introduction to the kalman filter," Chapel Hill, NC, USA, 1995, una breve introducción al Filtro de Kalman ofrecida por la institución University of North Carolina at Chapel Hill. [Online]. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[68] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, ser. IJCAI'81, vol. 2. Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[69] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 101.1–101.11.

[70] M. Kowarschik and C. Weiss, "An overview of cache optimization techniques and cache-aware numerical algorithms," in *Lecture Notes in Computer Science*, Springer, Ed., 2002, pp. 213–232.

[71] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Proceedings of the 17th International Conference on Pattern Recognition*, ser. ICPR '04, vol. 2. IEEE Computer Society, 2004, pp. 521–524.

[72] P. E. Eren, M. I. Sezan, and A. M. Tekalp, "Robust, object-based high-resolution image reconstruction from low-resolution video," in *Proceedings of the IEEE Transactions on Image Processing*, vol. 6, no. 10. IEEE, 1997, pp. 1446–1451.

[73] P. Metkar and N. Talbar, "Dynamic motion detection technique for fast and efficient video coding," in *IEEE Region 10 Conference TENCON*. IEEE, 2008, pp. 1–5.

[74] P. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, 2001.

[75] FFmpeg, "FFmpeg library," 2013, página de la librería gratuita FFmpeg, una solución completa y multi-plataforma para grabar, procesar y reproducir audio y video. [Online]. Available: http://www.ffmpeg.org

[76] B. A. R. Mohd Shahrimie, Mohd Asaari and S. A. Suandi, "Intelligent biometric group hand tracking (IBGHT) database for visual hand tracking research and development," *Multimedia Tools and Applications*, 2012.

[77] C. Shan, T. Tan, and Y. Wei, "Hand tracking using optical-flow embedded particle filter in sign language scenes," in *Proceedings of the 2012 International Conference on Computer Vision and Graphics*, ser. ICCVG'12. Springer-Verlag, 2012, pp. 288–295.

[78] G. Sperling, M. Landy, Y. Cohen, and M. Pavel, "Intelligible encoding of ASL image sequences at extremely low information rates," in *Proceedings from the second workshop on Human and Machine Vision*, vol. 13, no. 3. Academic Press Professional, Inc., 1986, pp. 256–312.

[79] Major Nelson, "Kinect price drop," 2012, página del director y gerente de la división de Xbox Live de Microsoft, y contiene información diversa sobre los distintos avances y productos desarrollados para la consola Xbox. [Online]. Available: http://majornelson.com/2012/08/22/kinect-price-drop

[80] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*, 4th ed. Academic Press, 2008.

[81] P. Gabbur, S. Pankanti, Q. Fan, and H. Trinh, "A pattern discovery approach to retail fraud detection," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '11. ACM, 2011, pp. 307–315.

[82] M. Isard and A. Blake, "Icondensation: Unifying low-level and high-level tracking in a stochastic framework," in *Proceedings of the 5th European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, vol. 1, no. 1406. Springer-Verlag, 1998, pp. 893–908.

[83] K. Takahashi and T. Kodama, "Remarks on simple motion capture using heuristic rules and monte carlo filter," in *Proceedings of the Fifth International Conference on Image and Graphics*, ser. ICIG'09. IEEE Computer Society, 2009, pp. 808–813.

[84] Y. Song, L. Goncalves, and P. Perona, "Learning probabilistic structure for human motion detection," in *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'01, vol. 2. IEEE Computer Society, 2001, pp. 771–777.

[85] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing 2002*, ser. ICIP'02, vol. 1. IEEE, 2002, pp. 900–903.

[86] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. IEEE Computer Society, 1998, pp. 839–846.

[87] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using non-parametric belief propagation," in *Proceedings of the 2004 Computer Vision and Pattern Recognition Workshop*, ser. CVPRW '04, vol. 12. IEEE Computer Society, 2004, pp. 189–197.

[88] T.-L. Liu and H.-T. Chen, "Real-time tracking using trust-region methods," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ser. PAMI'04, vol. 26, no. 3. IEEE Computer Society, 2004, pp. 397–402.

[89] M. Donoser and H. Bischof, "Efficient maximally stable extremal region (mser) tracking," in *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '06, vol. 1. IEEE Computer Society, 2006, pp. 553–560.

[90] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'01, vol. 1. IEEE, 2001, pp. 511–518.

[91] T.-J. Cham and J. M. Rehg, "A multiple hypothesis approach to figure tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'99, vol. 2. IEEE Computer Society, 1999, pp. 239–244.

[92] S. Goldenstein and C. Vogler, "When occlusions are outliers," in *Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition Workshop*, ser. CVPRW '06. IEEE Computer Society, 2006, pp. 98–105.

[93] N. Jacobson, Y. Freund, and T. Nguyen, "Occlusion boundary detection using an online learning framework," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP'11. IEEE, 2011, pp. 913–916.

[94] J. L. Crowley, J. M. Bedrune, M. Bekker, and M. Schneider, "Integration and control of reactive visual processes," in *Proceedings of the Third European Conference on Computer Vision*, ser. ECCV '94, vol. 2, no. 801. Springer-Verlag, 1994, pp. 47–58.

[95] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.