

# ALINEAMIENTO AUTOMÁTICO DE NUBES DENSAS DE PUNTOS A PARTIR DE INFORMACIÓN GEOMÉTRICA Y CROMÁTICA

Raúl Feliz Alonso  
[raulfeliz@gmail.com](mailto:raulfeliz@gmail.com)

Jaime Gómez García-Bermejo  
[jaigom@eis.uva.es](mailto:jaigom@eis.uva.es)

Eduardo Zalama Casanova  
[ezalama@eis.uva.es](mailto:ezalama@eis.uva.es)

Universidad de Valladolid – ETSII  
Paseo del Cauce, s/n  
47011 – Valladolid, España

## Resumen

*En este artículo se aborda el alineamiento automático de dos nubes de puntos tridimensionales expresadas en distintas referencias y correspondientes a un mismo objeto o escena. El objetivo consiste en encontrar la matriz de rotación y traslación, que aplicadas sobre una de las dos nubes, la lleven al alineamiento con la otra nube. Para ello se propone una nueva solución basada en el algoritmo Iterative Closest Point (ICP) que incorpora una serie de mejoras significativas, tales como la consideración de la información cromática, la búsqueda automática de puntos adecuados para alineamiento (por su geometría o por su color), y el muestreo aleatorio de puntos para reducir los requerimientos computacionales.*

**Palabras Clave:** Registro de imágenes tridimensionales. Muestras representativas. Reconstrucción de superficies, ICP.

## 1 INTRODUCCIÓN

El desarrollo tecnológico en el campo electrónico y físico de los últimos años ha permitido que algunos investigadores concentren sus esfuerzos en el desarrollo de métodos y tecnologías que permitan la digitalización (3-D) mediante escáneres y posterior representación de objetos del mundo real, para diferentes aplicaciones que abarcan desde la medicina hasta el campo del entretenimiento. Para que una representación de un objeto sea realista, necesita cierto nivel de detalle, y obtener esta calidad influye directamente en los costos computacionales. Por esto, es importante resaltar que ante la manipulación de cientos de miles y a veces millones

de datos, es imprescindible contar con procesos y algoritmos robustos.

El proceso de modelado de una imagen 3-D consta básicamente de dos etapas [2], [3]: adquisición de datos, y reconstrucción del modelo a partir de los datos obtenidos. Esta segunda fase se puede dividir en otras dos, que son registro e integración.

La adquisición corresponde a la toma de imágenes de rango de diferentes vistas del objeto y el tratamiento de estas imágenes para disminuir los efectos ópticos de la cámara. Se pueden emplear distintas técnicas para medir las superficies de los objetos. La más apropiada para corta distancia es la que emplea la triangulación óptica con luz estructurada [10]. Es una técnica de no contacto que permite una rápida, precisa y densa medida. La más apropiada para larga distancia es la medición del tiempo de vuelo (o diferencia fase), también sin contacto y que proporciona precisiones de pocos milímetros [13]. Ambas técnicas permiten además la adquisición de color (con el propio dispositivo o con un captador de imágenes externo), que es en lo que se basa el algoritmo aquí propuesto.

La segunda etapa es el registro o alineamiento de las superficies obtenidas. Generalmente, para obtener una completa descripción de las superficies de los objetos con este tipo de escaneado es necesario obtener varias imágenes de rango desde diferentes vistas debido al problema de la auto-oclusión. Estas imágenes deben tener regiones de solapamiento. Esta etapa por tanto se basa en encontrar las transformadas geométricas que referencien las diferentes vistas de un objeto a un mismo sistema coordinado.

El último paso corresponde a la integración, mediante la cual se busca una aproximación geométrica de la nube de puntos en la cual se eliminan redundancias en la zona de solapamiento entre las superficies, para así terminar la reconstrucción del modelo 3D.

En el presente trabajo se propone una nueva solución para el alineamiento de vistas, basada en el algoritmo ICP. Frente al algoritmo clásico, el propuesto en el presente artículo aporta las siguientes ventajas:

- *Alineamiento por color. Se aprovecharán los datos de color por punto obtenidos con los escáneres actuales a la hora de realizar el alineamiento con el fin de buscar una mayor precisión y robustez.*
- *Búsqueda de puntos significativos. Se realizará un muestreo previo para determinar qué puntos de la nube se pueden descartar y cuáles por el contrario poseen información significativa tanto de la geometría de la superficie representada como de su color. El objetivo es trabajar con el menor número de puntos posible sin perder información.*
- *Muestreo aleatorio y repetición del algoritmo. Se elegirá de manera aleatoria una pequeña muestra de los puntos significativos y se les aplicará el algoritmo. Una vez finalizado, se elegirá otra muestra de mayor tamaño y se volverá a aplicar el algoritmo tomando como posición inicial de la nube de puntos a alinear, la posición final de dicha nube tras la aplicación anterior del mencionado algoritmo.*

## 2 TRABAJO PREVIO

Para conseguir el alineamiento existen varias técnicas, pero sin duda la más popular de todas es el algoritmo ICP descrito por Bels y McKay en [1]. Es un algoritmo iterativo que se caracteriza por el poco coste computacional en cada iteración, lo que se traduce en velocidad de cálculo. No obstante, presenta dos grandes inconvenientes: una imagen debe ser un subconjunto de la otra, ya que el algoritmo es muy sensible a la presencia de *outliers*, y por otro lado es necesario conocer una buena aproximación inicial, porque de lo contrario el alineamiento podría ser defectuoso o podría requerir mucho tiempo en caso de que se probasen distintas posiciones iniciales.

Muchos han sido los autores que han desarrollado mejoras a esta propuesta. Las más importantes son las que se citan a continuación. Se clasificarán según la etapa del ICP que modifican o según la etapa nueva que introduzcan, al igual que hizo Rusinkiewicz en [15].

*Estado de rotación y traslación inicial:* Bels y McKay en [1] proponen un muestreo uniforme de la posición del cuaternio  $\tilde{q}_0$  inicial y elegir la que proporcione menor error cuadrático medio. Otros autores prefieren utilizar algoritmos genéticos como etapa previa a la aplicación del ICP [9]. Por último, una posibilidad adicional es realizar una selección de emparejamientos con la intervención del usuario para asegurar una buena posición inicial a la que aplicar el algoritmo.

*Etapas de selección de puntos:* El algoritmo ICP original utiliza todos los puntos posibles. Esto supone la posible inclusión de *outliers*, y necesitar que una muestra fuese un subconjunto de la otra. Existen variantes que realizan una selección previa de los puntos con los que se trabajará.

Turk y Levoy en [17] proponen un selección uniforme de M puntos de entre los N que forman la muestra para obtener información de toda ella con menos puntos. Masuda en [10] propone una solución similar. Un muestreo aleatorio, con la diferencia de que en cada iteración se tomará una muestra aleatoria distinta.

Rusinkiewicz en [14] y [15] propone calcular la normal asociada a cada punto, agrupar las normales por su orientación y elegir de manera aleatoria puntos dentro de cada grupo de normales. Otros autores usan información adicional, como Weik en [18], que toma los valores de intensidad de cada punto y trabaja con los mayores.

*Etapas de emparejamiento de puntos:* Bels y McKay en [1] proponen emparejar con el punto más cercano de la otra muestra. Chen y Medioni en [2], casi al mismo tiempo, proponen calcular la normal asociada a cada punto y calcular su intersección con la otra muestra. El punto donde intersecten será el emparejamiento.

Otros autores en lugar del punto más cercano, buscan el punto compatible más cercano. Dicha compatibilidad se puede buscar de varias maneras: por intensidad [18], por color [8], [11] y [16], por orientación entre normales [12].

*Etapas de asignación de pesos a los pares:* La asignación de pesos busca realzar las cualidades de correspondencia de los pares seleccionados, disminuir el efecto de pares mal seleccionados o de baja correspondencia, dándoles o quitándoles prioridad en las etapas consecutivas del algoritmo. Existen varias formas de asignar pesos a los apareamientos, entre ellas están: asignar un peso constante, asignar pesos pequeños a los pares con distancia punto a punto mayor [5], por

compatibilidad de colores, por compatibilidad de normales [15], entre otras.

Rusinkievicz en [15] demuestra que la asignación de pesos no presenta gran incidencia en el desarrollo del algoritmo y su aporte a la optimización del mismo es mínimo.

*Etapa de rechazo de pares:* Los métodos propuestos por los autores son: eliminar los pares que tengan entre sus puntos distancias mayores a un umbral propuesto (conocido) [15], o hallar este umbral por medio del valor de la desviación estándar de las distancias entre los emparejamientos [10]. Rechazar un porcentaje fijo de pares de acuerdo a la distancia punto a punto [3], [12] y [20].

*Medida y minimización del error:* El paso final del proceso de alineamiento consiste en escoger una forma de medir el error entre los pares seleccionados y el proceso que lleve a la minimización del mismo. Diferentes métodos se han propuesto para escoger el tipo de medida del error, basándose en la suma del cuadrado de las distancias de los puntos correspondientes, error punto a punto basado en distancia o color, la suma del cuadrado de la distancia del punto origen al plano destino [2], entre otros.

Los algoritmos de minimización del error se relacionan directamente a la métrica elegida. Para el error por suma del cuadrado de las distancias se utilizan métodos de solución como: cuaternios [6], descomposición en valores singulares, matrices ortonormales [7]. Si se selecciona la métrica punto a plano [3], se deben usar métodos no lineales de minimización o linealizar el problema asumiendo que la variación en la rotación necesaria para minimizar el error es pequeña.

Frente a estas soluciones, el procedimiento propuesto en el presente artículo se basa en una mejora tanto en la etapa de selección de puntos como en la de emparejamiento de los mismos. En la etapa de selección de puntos, se hará un estudio previo de la nube de puntos que se desea alinear, descartando aquellos puntos que aporten menor información en relación a características de color como de geometría y trabajando sólo con aquellos cuya información sea relevante. Además, de entre todos los puntos seleccionados, se trabajará con muestras de menor tamaño para aligerar la carga computacional del proceso. Con respecto a la segunda etapa modificada, utilizaremos la información de color para emparejar los puntos de una superficie con el punto más compatible de la otra superficie, atendiendo a su color y posición en el espacio, y no con el más cercano, como hace el ICP.

### 3 DESCRIPCIÓN DEL ALGORITMO EMPLEADO

Como punto de partida se tienen dos nubes de puntos  $\mathcal{P} = \{\vec{p}_i\}_{i=1..N_p}$  y  $\mathcal{X} = \{\vec{x}_i\}_{i=1..N_x}$ . Se pretende alinear  $\mathcal{P}$  con  $\mathcal{X}$ . Para ello es necesario encontrar la matriz de rotación y un vector de traslación, denotando al conjunto como  $T_{LMS}$ , tal que  $T_{LMS}(\mathcal{P})$  se alinee con  $\mathcal{X}$ . El algoritmo pretende encontrar el valor de  $T_{LMS}$  de forma iterativa. Los pasos a seguir son:

- 1) Inicialización:  $T_{LMS} \leftarrow T_0$ . Estimación inicial de la matriz y vector de traslación a partir del posicionamiento inicial de las nubes de puntos o bien por selección del usuario.
- 2) Cálculo de la muestra de puntos  $\vec{s}_i \in \mathcal{P}$  que contienen la información más significativa en cuanto al color y la geometría de la superficie representada por  $\mathcal{P}$ .
- 3) Estimación de los puntos  $\vec{z}_i \in \mathcal{P}$  considerados pertenecientes a la zona de solapamiento entre ambas imágenes  $\gamma$ .
- 4) Elección de una muestra de trabajo de  $n = n_0$  puntos de entre todos los pertenecientes al conjunto  $\vec{s}_i \in \gamma$ , siendo  $n_0$  un número a elección del usuario. En este documento se ha elegido  $n_0$  de manera automática como el 10% de número de puntos que pertenecen a  $\gamma$ .
- 5) Aplicación del algoritmo ICP de la siguiente forma:
  - a) Emparejamiento de la muestra elegida en 4) con puntos  $\vec{x}_j \in \mathcal{X}$ , teniendo en cuenta tanto la posición en el espacio de los puntos como su valor de color RGB.
  - b) Cálculo de las matrices de rotación y traslación  $T_{LMS}$ .
  - c) Aplicación de dichas matrices sobre la muestra elegida y cálculo de la media del cuadrado de las distancias entre los puntos una vez aplicada la transformación.
  - d) Si se cumple que dicho valor no difiere del obtenido en la iteración anterior en más de una tolerancia establecida, detener las iteraciones. En caso contrario volver al paso a)

$$d_k - d_{k+1} < \tau_{err} \quad (1)$$

- 6) Si el número de puntos  $n$  de la muestra de trabajo corresponde con el total de puntos del conjunto  $\vec{s}_i \in \gamma$ , detener el algoritmo. En caso contrario continuar al paso 7)
- 7) Si el número de iteraciones realizadas por el algoritmo en 5) es  $N_{iter} > 2$ , hacer  $n = 2n$  y volver al paso 3). En caso contrario,  $N_{iter} \leq 2$ , volver al paso 3) con la condición de que en el paso 4) los  $n$  puntos elegidos corresponderán a la totalidad del conjunto  $\vec{s}_i \in \gamma$

### 3.1 SELECCIÓN DE LOS PUNTOS SIGNIFICATIVOS

Los puntos que contienen mayor información acerca de la geometría de la superficie son los que formarán la muestra de “puntos óptimos por geometría”. Dichos puntos serán los llamados puntos cresta y puntos valle. Para su cálculo se hace un estudio de las curvaturas locales, como describen Yoshizawa, Belyaev y Seidel en [19].

Para la selección de puntos que contienen la información más significativa en cuanto al color de la superficie de cara al alineamiento, se ha hecho una comparación entre el valor RGB de un punto y los valores RGB de los puntos de su vecindad. Se tomarán como “puntos óptimos por color” aquellos que cuya diferencia de valores RGB supere una determinada tolerancia. Dados dos puntos  $\vec{p}_1$  y  $\vec{p}_2$  con sus respectivos valores RGB  $c_1$  y  $c_2$ . Su diferencia viene determinada por la siguiente expresión:

$$c_1 = (R_1, G_1, B_1) \quad (2)$$

$$c_2 = (R_2, G_2, B_2) \quad (3)$$

$$(c_1 - c_2) = \begin{bmatrix} |R_1 - R_2| \\ |G_1 - G_2| \\ |B_1 - B_2| \end{bmatrix} = \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (4)$$

Serán óptimos todo  $\vec{p}_1$  y  $\vec{p}_2$  tal que  $R' \geq \tau_c$  o  $G' \geq \tau_c$  o  $B' \geq \tau_c$ , donde  $\tau_c$  es la tolerancia seleccionada.

### 3.2 DETERMINACIÓN DE LA ZONA COMÚN

El algoritmo presenta dos opciones. Determinación de la zona de manera manual, en la que el usuario selecciona varias correspondencias entre las superficies a alinear. Se obtienen las matrices de rotación y traslación para alinear los puntos seleccionados por el usuario. Dichas matrices formarán el  $T_0$  inicial. Una vez aplicado  $T_0(\mathcal{P})$ , se consideran puntos comunes aquellos  $\vec{p}_i$  cuya distancia euclídea a los puntos de la muestra  $\mathcal{X}$  sean menores que un cierto umbral:

$$d(\vec{p}_i, \mathcal{X}) \leq \tau_d \quad (5)$$

La tolerancia  $\tau_d$  se determina a partir de la máxima distancia a la que han quedado los puntos seleccionados entre sí después de aplicar  $T_0(\mathcal{P})$ .

$$\tau_d = \max(d(T_0(\vec{s}_i), \mathcal{X})) \quad (6)$$

Para la determinación automática, se calculan las distancias entre los puntos  $\vec{p}_i$  y  $\mathcal{X}$ , y se eligen el 10% de ellos cuya distancia sea menor. Se elige sólo el

10% como criterio conservativo, porque es preferible no utilizar puntos que sí son comunes a considerar puntos que no lo son.

Para la determinación de la zona común en las sucesivas aplicaciones, el proceso será similar al primero de los citados anteriormente. Se tomarán aquellos puntos cuya distancia a  $\mathcal{X}$  no supere una determinada tolerancia, calculada a partir de las distancias a las que han quedado los puntos  $\vec{p}_i$  de sus respectivos emparejamientos en  $\mathcal{X}$  después de aplicar  $T_{LMS}$ .

### 3.3 APLICACIÓN DEL ALGORITMO ICP

El algoritmo aplicado para el alineamiento es el descrito por Bels y McKay en [1], por lo que no se van a describir en este documento los pasos seguidos, salvo el caso del emparejamiento de puntos, que el donde radica la mayor diferencia con el trabajo original.

Ya se explicaron en la sección 2 las variantes que habían propuesto distintos autores en el emparejamiento de puntos entre ambas superficies. En este algoritmo se parte de la idea de elegir el punto compatible más cercano haciendo uso de la información de color para determinar dicha compatibilidad. No obstante, sólo se usará la información de color para los puntos considerados significativos para el alineamiento por color. Para los considerados significativos por geometría, se procederá a emparejarlo con el más cercano, igual que en el documento original.

La diferencia con respecto a lo propuesto por otros autores, como en [8] y [16], es que ellos suponían que el valor RGB de una misma zona en ambas superficies era el mismo. Sin embargo, variaciones en las condiciones de iluminación tanto natural como artificial entre los distintos escaneados de una misma zona pueden dar lugar a un tono de color distinto.

La solución aquí propuesta consiste en comparar un punto y sus puntos vecinos con otro punto y sus puntos vecinos. Se obtendrá información por medio de una comparación entre el color RGB de un punto de la superficie dato y sus vecinos inmediatos dentro de esa superficie. Dicha información consistirá en valores como la media, desviación típica y máximo de entre todos los valores obtenidos. Así mismo se obtendrá la misma información para el cada punto de la superficie modelo con los que se le pretende emparejar. Por lo tanto, en lugar de comparar sólo los valores RGB de ambos puntos, compararemos todos los valores obtenidos para cada punto.

Sea  $c_1 = (R_1, G_1, B_1)$  el valor de color RGB del punto  $\vec{p}_1$  y  $c_2 = (R_2, G_2, B_2)$  el valor de color RGB

del punto  $\vec{p}_2$ . Llamaremos “distancia de color” entre ambos a:

$$d_c(\vec{p}_1, \vec{p}_2) = |R_1 - R_2| + |G_1 - G_2| + |B_1 - B_2| \quad (7)$$

Sea  $\vec{p}_{vec}$  el vector que contiene a los  $n$  puntos más próximos de un punto  $\vec{p}_i$ . Se calcularán las distancias de color entre ese punto sus vecinos

$$\vec{d}_{c_i} = d_c(\vec{p}_i, \vec{p}_{vec_i})_{i=1..n} \quad (8)$$

De esta forma se tienen almacenadas en el vector  $\vec{d}_c$  todas las distancias de color entre un punto y sus vecinos. La información a determinar es:

$$med(p) = \frac{\sum_1^n d_{c_i}}{n} \quad (9)$$

$$\max(p) = \max(d_{c_i})_{i=1..n} \quad (10)$$

$$std(p) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_{c_i} - med(p))^2} \quad (11)$$

Se tendrán estos valores asociados a cada punto que vayamos a comparar en ambas superficies. Compararemos todos ellos además, claro está, de sus respectivos valores RGB.

Para determinar con qué puntos de  $\mathcal{X}$  hemos de comparar cada punto de  $\mathcal{P}$ , buscaremos el más cercano, como hace el algoritmo original y comparamos con todos los puntos de un entorno a ese punto más cercano. El radio de dicho entorno es variable. Inicialmente se le asigna un valor en función del tamaño de la nube de puntos para posteriormente irlo reduciendo a medida que la muestra se va alineando iteración tras iteración. Su cálculo se detalla a continuación. Sea  $\vec{p}_1 = (x_1, y_1, z_1)$  y  $\vec{p}_2 = (x_2, y_2, z_2)$  los puntos más alejados entre sí de la superficie  $\mathcal{P}$ . Se tomará como radio inicial del entorno la décima parte de la distancia geométrica entre esos dos puntos:

$$r_e = \frac{d(\vec{p}_1, \vec{p}_2)}{10} \quad (12)$$

Por razones de optimización en el cálculo computacional se sustituye el entorno esférico de radio  $r_e$  por un entorno cúbico de lado  $l_e = 2r_e$  cuyo centro coincide con el centro del entorno esférico. Dicho entorno se re-calcula iteración tras iteración con el fin de evitar la comparación entre un punto  $\vec{p}_i$  y muchos puntos  $\vec{x}_j$  cuando las superficies están ya muy cercanas. El cálculo del nuevo valor de  $l_e$  se detalla a continuación. Sean  $\mathcal{Y} = \{\vec{y}_i\}_{i=1..n}$  los  $n$  puntos pertenecientes a  $\mathcal{Y}$  con los que se está trabajando, y sea  $\mathcal{Z} = \{\vec{z}_i\}_{i=1..n}$  los  $n$  puntos

pertenecientes a  $\mathcal{X}$  con los que se han emparejado esos  $\vec{y}_i$ . Se denota  $T_{LMS}$  como el conjunto formado por la matriz de traslación y rotación obtenidas en la iteración presente a partir de estas dos nubes de puntos  $\mathcal{Y}$  y  $\mathcal{Z}$ , de tal forma que  $T_{LMS}(\mathcal{Y})$  alineará  $\mathcal{Y}$  con  $\mathcal{Z}$ . El nuevo valor del radio del entorno será la máxima de las distancias entre los puntos de  $\mathcal{Y}$ , una vez aplicada la transformación, respecto de sus respectivos emparejamientos en  $\mathcal{Z}$ .

$$r_e = \max(d(T_{LMS}(\mathcal{Y}), \mathcal{Z})) = \max(d(T_{LMS}(\vec{y}_i), \vec{z}_i)) \quad \forall i = 1..n \quad (13)$$

Por tanto el lado del cubo será nuevamente:

$$l_e = 2r_e \quad (14)$$

## 4 RESULTADOS

Para este primer ensayo se han utilizado los datos 3D-color resultantes de escanear el bajorrelieve *La Piedad* del conocido imaginero vallisoletano Juan de Juni. Se ha utilizado un escáner de triangulación Vivid 910, de Minolta. En esta primera prueba se muestran aquí los resultados correspondientes al alineamiento de dos nubes de puntos con una superficie de solapamiento entre ellas. Dichas nubes de puntos poseen información de color por punto, que será usada en el alineamiento. La determinación de la zona de solapamiento la realiza de forma automática el algoritmo, olvidando en estas pruebas la opción de selección de puntos por el usuario. Marcar puntos supondría dar información para que las superficies se sitúen en una aproximación inicial cercana a la solución final, con lo que el algoritmo original alineará bien y no se apreciará la mejora introducida por el algoritmo aquí propuesto.

### 4.1 PRIMERA PRUEBA

En esta primera prueba se pretende demostrar las ventajas del uso del color en el alineamiento de dos nubes de puntos. Para ellos se ha seleccionado una muestra a alinear en la que la zona de solapamiento presenta mucha información de color. Por el contrario, su geometría no es muy significativa. Contiene bastantes zonas planas, por lo que alinearlo solo usando la información de la posición de los puntos en el espacio dará lugar a alineamientos incorrectos, a no ser que se parta de una posición muy próxima a la solución. Para ver realmente las ventajas de aplicar este algoritmo, se partirá de posiciones no tan cercanas a la solución.

En la figura 1 se muestran las dos vistas que se van a alinear, mientras que en la figura 2 vemos la posición inicial de las mismas antes de aplicar el algoritmo.

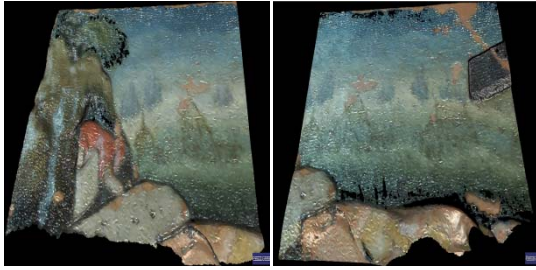


Figura 1: Muestras a alinear. La zona de solapamiento entre ambas posee mucha información de color pero se asemeja bastante a un plano.

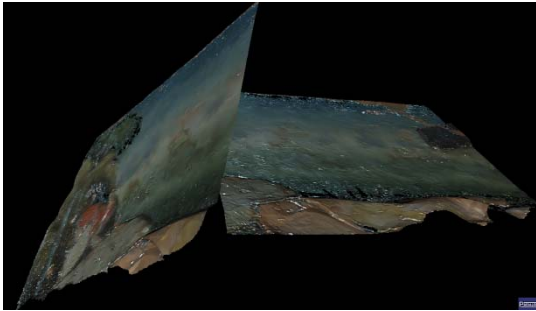


Figura 2: Posición inicial de las nubes de puntos antes del alineamiento.

Realizaremos dos ensayos para alinear las muestras. En el primero de ellos aplicaremos el algoritmo aquí propuesto. Usaremos por tanto el color para realizar el emparejamiento entre los puntos de una y otra muestra. El segundo ensayo consistirá en la aplicación del algoritmo original, haciendo por tanto el emparejamiento por el método de encontrar el punto más cercano en la otra superficie. Compararemos después los resultados.

En la figura 3 podemos ver los puntos que formarán las muestras representativas por color y por geometría con los que vamos a trabajar.

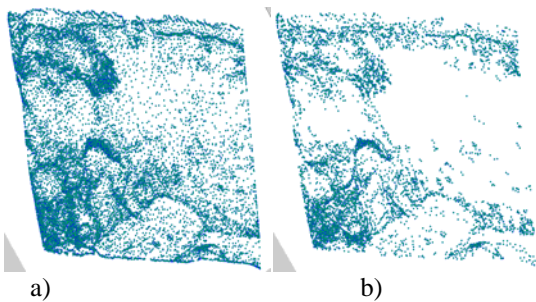


Figura 3: a) puntos significativos por color.  
b) puntos significativos por geometría.

Como se puede apreciar, la muestra de color presenta muchos puntos en la zona de solapamiento mientras que muestra de geometría casi ninguno. Por eso al aplicar el algoritmo original se obtendrán peores

resultados, ya que la geometría de todos los puntos de la zona de solapamiento está próxima a la de un plano y por tanto será difícil el alineamiento de manera exacta.

En un principio se ha usado una muestra de 50 puntos para el alineamiento, elegidos como se ha descrito en el punto 3 de este documento. Una vez alcanzado un mínimo, se ha elegido otra muestra de 100 puntos y se ha vuelto a aplicar el algoritmo, para ir afinando el alineamiento. Se ha seguido aumentando la muestra de trabajo hasta que se han cumplido los criterios de parada descritos en la sección 3.

Para el segundo ensayo se han empleado todos los puntos detectados como comunes, ya que el algoritmo ICP original no distingue puntos significativos de geometría y color.

Los resultados obtenidos después de aplicar ambos algoritmos se presentan en las figuras 4 y 5.



Figura 4: Muestras alineadas mediante el algoritmo propuesto. Los parámetros usados han sido:  
 $n_0 = 50$ ,  $\tau_c = 0.3$ ,  $\tau_{err} = 1 \times 10^{-3}$ .

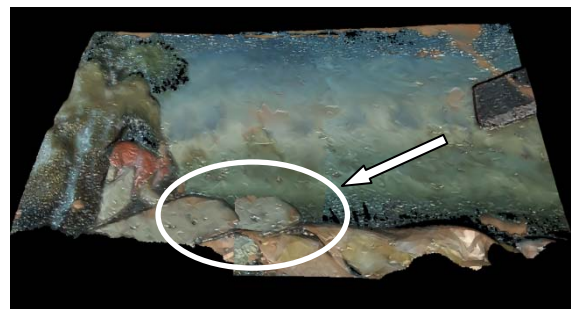


Figura 5: Muestras alineadas mediante el algoritmo original.

En la zona señalada se aprecia que el alineamiento no ha sido óptimo. Las superficies están separadas entre sí, con respecto a lo que sería la solución óptima, alcanzada en la figura 4.

Las superficies a alinear estaban en una posición muy extrema, con el fin de probar que, incluso en esas circunstancias, el uso de la información de color hace



posible el alineamiento. En el siguiente ensayo se parte de una posición inicial más cercana al alineamiento final con el fin de demostrar que el algoritmo aquí propuesto presenta ventajas incluso en el caso de que ambos alineen las muestras.

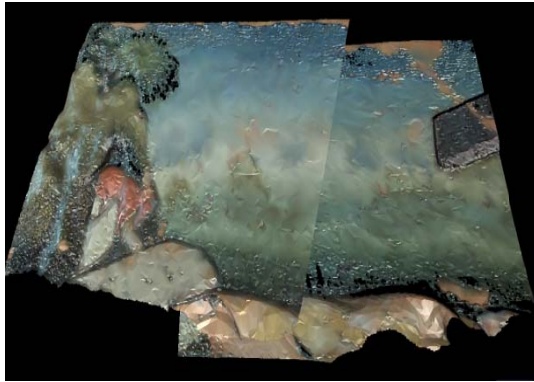


Figura 6: Posición inicial de las nubes de puntos antes del alineamiento.



Figura 7: Muestras alineadas mediante el algoritmo propuesto. Los parámetros usados han sido:  $n_0 = 50$ ,  $\tau_c = 0.3$ ,  $\tau_{err} = 1 \times 10^{-3}$ .

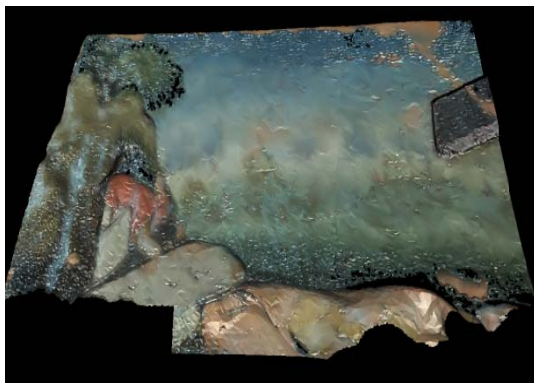


Figura 8: Muestras alineadas mediante el algoritmo original.

En la figura 9 se muestra una comparación de la evolución del error en ambos algoritmos

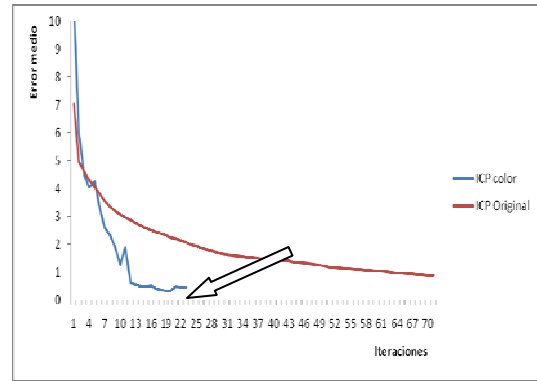


Figura 9: Comparación de la evolución del error en ambos algoritmos.

Se observa cómo a pesar de haber obtenido el alineamiento en ambos algoritmos, en el caso del ICP con información el número de iteraciones ha sido sensiblemente menor: 23 frente a 71, con lo que se demuestra que la convergencia es más rápida, en cuanto a número de iteraciones se refiere. Por otra parte, el valor final del error medio obtenido es de 0.90 para el ICP original frente al 0.45 del ICP con color propuesto. Aunque a simple vista esta diferencia no se aprecia en las imágenes, puede llegar a ser importante cuando se alinean más de dos imágenes, y el error, por pequeño que sea, se va arrastrando y puede tener un peso importante en el alineamiento global de todas las superficies.

Puede dar la impresión que el pequeño incremento del error señalado por la flecha en las últimas iteraciones va en contra de convergencia del algoritmo en la minimización del error cuadrático medio. No es así. Se debe a que en la iteración número 20 el algoritmo alcanzó el mínimo para la muestra de  $n$  puntos con la que se estaba trabajando y, como se describió en la sección 3, se aplicó nuevamente dicho algoritmo a todo el conjunto de puntos  $\mathcal{S}_i \in \gamma$ . Esto da lugar a un posible aumento del error cuadrático medio, ya que intervienen más puntos al calcularlo. La convergencia queda demostrada al comprobar que entre la iteración 21 y 23 el error cuadrático medio no aumentó.

## 4.2 SEGUNDA PRUEBA

Para este ensayo se han utilizado los datos 3D-color correspondientes a un fragmento de iglesia románica, escaneada mediante un escáner Leica HDS 3000 de tiempo de vuelo. Se pretende demostrar la robustez del algoritmo frente a muestras con una notable variación en el color de la zona de solapamiento de una de las superficies respecto de la otra. Se intentará alinear dos nubes de puntos que representan una ventana. Una de las nubes de puntos presenta una significativa modificación en el color, con el fin de reproducir una eventual variación espectral de la luz

solar durante la adquisición (debida a la presencia de nubes o a diferencia en la hora del día de las adquisiciones.).

En las figuras 6 y 7, al igual que antes, se muestran las dos vistas que se van a alinear y la posición inicial de las mismas antes de aplicar el algoritmo.



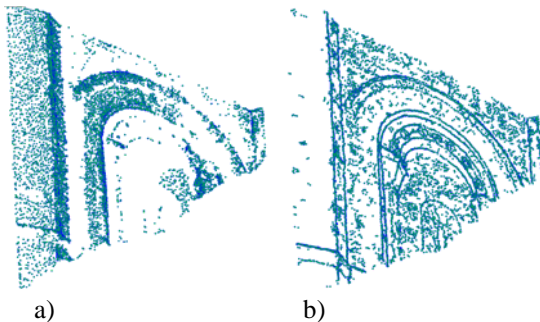
Figura 10: Muestras a alinear. Se puede apreciar entre ambas muestras la diferencia de color mencionada.



Figura 11: Posición inicial de las nubes de puntos antes del alineamiento.

Al igual que en la prueba anterior, realizaremos los dos ensayos para alinear las muestras. Compararemos después los resultados.

En la figura 12 podemos ver los puntos que formarán las muestras representativas por color y por geometría con los que vamos a trabajar.



a) puntos significativos por color.  
b) puntos representativos por geometría.

En un principio se ha usado una muestra de 100 puntos para el alineamiento, elegidos como se ha descrito en la sección 3 de este documento. Una vez alcanzado un mínimo, se ha elegido otra muestra de 200 puntos y se ha vuelto a aplicar el algoritmo, para ir afinando el alineamiento. Se ha seguido aumentando la muestra de trabajo hasta que se han cumplido los criterios de parada descritos en el punto 3.

Para el segundo ensayo se han empleado absolutamente todos los puntos detectados como comunes, ya que para aplicar el algoritmo ICP tradicional no tiene sentido andar distinguiendo entre puntos significativos por color y por geometría.

Los resultados obtenidos después de aplicar ambos algoritmos se presentan en las figuras 12 y 13.



Figura 13: Muestras alineadas mediante el algoritmo propuesto. Los parámetros usados han sido:  
 $n_0 = 100$ ,  $\tau_c = 0.3$ ,  $\tau_{err} = 1 \times 10^{-3}$ .

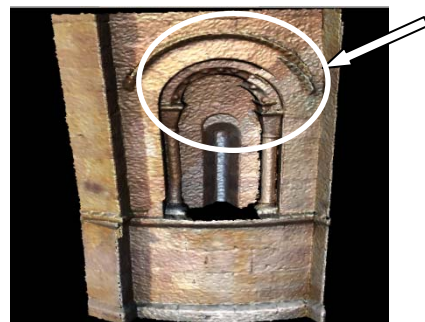


Figura 14: Muestras alineadas mediante el algoritmo original.

Como se puede ver en la zona marcada, el alineamiento no ha sido el óptimo mientras que en el ensayo anterior, el que usaba el color, sí que lo ha sido.



## 5 CONCLUSIONES

Las conclusiones a las que se han llegado después de las distintas pruebas de alineamiento, de las cuales aquí se han presentado algunos de los resultados, son las siguientes:

- El algoritmo aquí presentado permite alinear nubes de puntos 3-D con información de color por punto. Su utilidad radica en poder reconstruir un objeto a partir de muestras parciales del mismo, de forma que éste pueda ser visualizado de manera completa por un navegador.
- Otras de las posibilidades del alineamiento es la comparación de piezas con un patrón, de cara a un control de calidad.
- El alineamiento depende bastantes de cuál es la aproximación inicial de la que partamos. Cuanto más cercano esté de la solución, mejor alineará.
- Al no ser una de las muestras un subconjunto de la otra, hay que determinar la zona de solapamiento. La fiabilidad con la que se determine, condicionará el alineamiento, ya que si se seleccionan pocos puntos, se pueden desechar puntos con información muy significativa para el alineamiento, mientras que se seleccionan muchos puntos, se intentará alinear puntos de una superficie que no tienen su correspondencia en la otra y dará lugar a soluciones no deseadas. Es por eso que se ha decidido ir corrigiendo esta aproximación de la zona común a medida que se aplica el algoritmo.
- El tomar una muestra de puntos pequeña hace que pueda ser alineada en varias partes del objeto. Cuanto mayor sea la muestra, mayor será la probabilidad de que la alinee en el sitio correcto. Como contrapartida, una muestra grande supone un tiempo de cálculo más elevado y una mayor posibilidad de incluir outliers en la muestra. Por eso la solución aquí tomada es empezar con muestras pequeñas e ir aumentando su tamaño a medida que la muestra se va alineando, de tal forma que evitaremos incluir ruido y trabajaremos con suficiente información como para encontrar un alineamiento óptimo.
- El elegir muestras pequeñas es el único modo de hacerlo competitivo con el algoritmo original en lo que a tiempo de ejecución se refiere, ya que a igualdad de puntos a alinear, el algoritmo aquí propuesto requerirá mayor tiempo de cálculo por iteración, ya que la comparación de color entre puntos involucrará un mayor número de operaciones. Como ventaja adicional, el uso del color hará que el algoritmo converja en un menor número de iteraciones.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia (Proyecto CICYT DPI2005-06911), el Ministerio de Fomento (Proyecto C17/2006) y la Junta de Castilla y León (Proyecto VA011A06).

## Referencias

- [1] Besl, P.J. y McKay, N., (1992) "A method for registration of 3-d shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 239–256, vol. 14 n°2.
- [2] Chen, Y. y Medioni, G., (1991) "Object modeling by Registration of Multiple Range Images", *IEEE International Conference on Robotics and Automation*, pages 2724–2729.
- [3] Chetverikov, D., Svirko, D. y Stepanov, D., (2002) "The Trimmed Iterative Closest Point", *16th international conference on pattern recognition*, pp. 545-548.
- [4] Dorai, C., Weng, J. y Jain, A. (1998) "Registration and Integration of Multiple Object Views for 3D Model Construction", *Trtnrs. PAMI*, Vol. 20, No. 1.
- [5] Godin, G., Rioux, M. y Baribeau, R. (1994) "Three-dimensional Registration Using Range and Intensity Information", *Videometrics III*, Vol. 2350.
- [6] Horn, B., (1987) "Closed-Form Solution of Absolute Orientation Using Unit Quaternions", *Journal of the Optical Society of America A*, vol. 4, n° 4, pp. 629-642,
- [7] Horn, B., Hilden, H. y Negahdaripour, S., (1988) "Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices", *Journal of the Optical Society of America A*, Vol. 5, No. 7.
- [8] Johnson, A. y Kang, S. (1996) "Registration and Integration of Textured 3-D Data", *Conference on Recent Advances in 3-D Digital imaging and Modeling*, pp. 234–241.
- [9] Lomonosov, E., Chetverikov, C. y Anikó, E., (2004) "Fully automatic, robust and precise alignment of measured 3D surfaces for arbitrary orientations", *Computer and Automation Research Institute and Eötvös Loránd University, Budapest, Hungary*.

- [10] Masuda, T., Sakaue, K. y Yokoya, N., (1996) “Registration and Integration of Multiple Range Images for 3-D Model Construction”, *IEEE International Conference on Pattern Recognition*, pp. 879–883.
- [11] Pulli, K., (1997) “Surface Reconstruction and Display, from Range and Color Data”. *Ph. D. Dissertation, University of Washington*.
- [12] Pulli, K., (1999) “Multiview Registration for Large Data Sets”, *Second International Conference on 3-D Digital Imaging and Modeling*, pp. 160–168.
- [13] Remondino, F., El-Hakim, S. F., (2006) “Image-Based 3D Modeling: A Review”, *The Photogrammetric Record Journal*, Volume 21, Number 115, pp. 269 – 291.
- [14] Rusinkiewicz, S. M., (2001) “Real-Time Acquisition and Rendering of Large 3D Models”, *PhD thesis, Stanford University*.
- [15] Rusinkiewicz, S. M. y Levoy, M., (2001) “Efficient variants of the ICP algorithm”, *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM)*, pp. 145–152.
- [16] Sehwan, K., Kiyoun, K. y Woontack, W., (2004) “3D registration for image-based virtual environment generation using color and depth information”, *IEEE Pacific Rim Conference on Multimedia*.
- [17] Turk, G. y Levoy, M., (1994) “Zippered Polygon Meshes from Range Images”, *Proceedings of SIGGRAPH 94*, pp. 311–318.
- [18] Weik, S., (1997) “Registration of 3-D Partial Surface Models Using Luminance and Depth Information”, *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pp. 93–100.
- [19] Yoshizawa, S., Belyaev, A. y Seidel, H. P., (2005) “Fast and Robust Detection of Crest Lines on Meshes”, *Proceedings of the Ninth ACM Symposium on Solid and Physical Modeling*, pp. 227 – 232.
- [20] Zhang, Z., (1994) “Iterative point matching for registration of free-form curves and surfaces”, *International Journal of Computer Vision*, 13(2), pp: 119–152.