

Universidad Nacional de La Plata
Facultad de Informática



Segmentación espectral de imágenes obtenidas con cámaras de tiempo de vuelo

Autor: Luciano Lorenti
Director: Javier Giacomantone

22 de abril de 2014



El trabajo realizado en la presente tesina se vincula con un proyecto acreditado por la Universidad Nacional de La Plata dentro del Programa de Incentivos llevado a cabo por el III-LIDI (Instituto de Investigación en Informática - LIDI) de la Facultad de Informática.



- Segmentación de imágenes TOF
- Segmentación espectral. Su aplicación en imágenes TOF
- Librería para la cámara SR4000
- Entorno de captura y visualización para la cámara SR4000
- Software de prueba y análisis



- El propósito de un método de segmentación es descomponer una imagen en sus partes constitutivas.
- Han sido utilizados diversos métodos de segmentación en ambientes industriales, con colores e iluminación controlada.
- Estos métodos no obtienen resultados similares en contextos diferentes.



- Las imágenes obtenidas con cámaras tradicionales no pueden capturar toda la información de un escena 3D.
- Se proyecta una escena 3D para obtener una representación bidimensional.
- El proceso de captura crea ambigüedades entre el tamaño de los objetos y la distancia con respecto al sensor.
- Dispositivos de captura pasivos.
- Una alternativa para otorgarle robustez a los algoritmos de segmentación es **incorporar información de profundidad**.



- Mapa de Profundidad
- Interferometría
- Triangulación
- Tiempo de vuelo



Se utilizó la cámara MESA SR 4000 de la empresa suiza SwissRanger.

- Cámara Activa: Utiliza su propia fuente de iluminación. Diodos emiten luz infrarroja modulada en amplitud.
- Los sensores de la cámara detectan la luz reflejada en los objetos y genera dos imágenes:
 - 1 Imagen de Intensidad: Proporcional a la amplitud de la onda reflejada
 - 2 Imagen de Rango: Generada a partir de la diferencia de fase entre la onda emitida y reflejada



- Obtención en forma simultánea de imágenes de rango e intensidad
- Velocidades de capturas compatibles con aplicaciones de tiempo real
- Obtención de nubes de puntos 3D a partir de un sólo punto de captura



- En las imágenes de intensidad de las cámaras TOF predominan los detalles generados por la textura de los objetos.
- En las imágenes de rango los detalles y ruido asociados al fondo de la escena 3D.

Segmentar objetos usando técnicas que utilicen ambas imágenes.



- TOFCapture es una aplicación multiplataforma escrita en C++ que permite la captura, almacenaje y documentación de las imágenes obtenidas con la cámara SwissRanger SR4000.

Librerías utilizadas

- Librería de la empresa SwissRanger
- Framework Qt
- CFitsIO

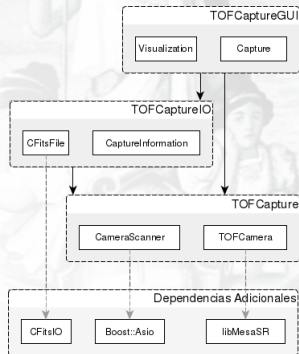


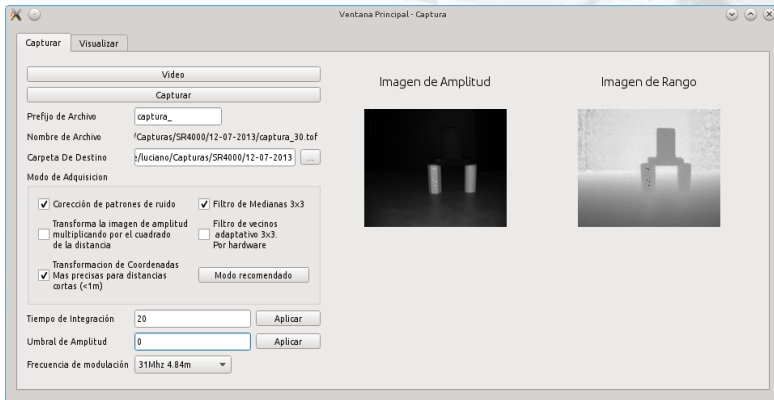
- La falta de portabilidad por parte de la librería provista por la empresa SwissRanger:
 - El descubrimiento de la cámara en la red.
 - La persistencia de las capturas.
- Carencia de funciones destinadas a la documentación de las imágenes.



TOFCapture se divide en tres componentes principales:

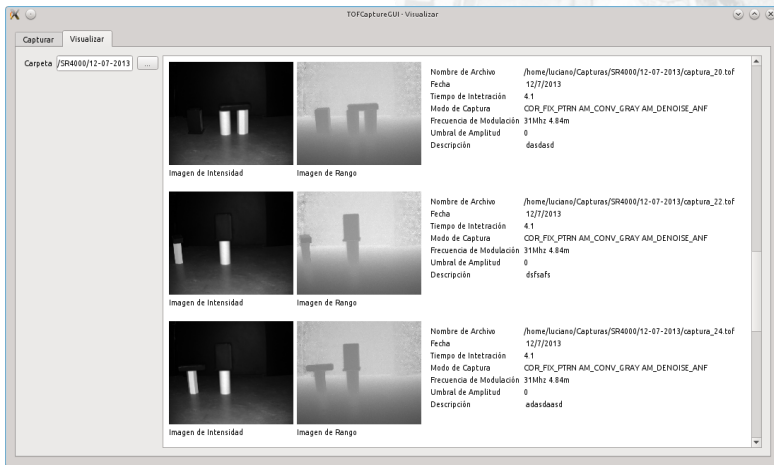
- Una librería para realizar capturas con la cámara llamada TOFCapture
- Una librería para la persistencia y documentación de las imágenes llamada TOFCaptureIO
- Un entorno gráfico, denominado TOFCaptureGUI, que provee una interfaz amigable que utiliza las funciones de las dos librerías anteriores







Una vez seleccionada una carpeta, el programa lista todas las imágenes junto con sus metadatos.

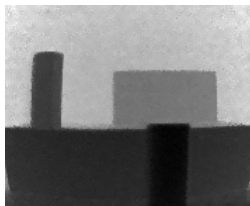




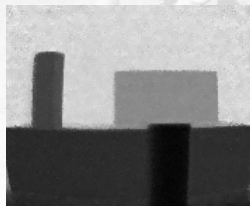
- Realiza una extracción de bordes mejorada combinando información útil de las dos imágenes.
 - Se utiliza para ésto una operación lógica definida por Danciu basada en la operación AND
- A partir de las regiones delimitadas por los bordes, analiza cuáles pertenecían al mismo objeto pero fueron segmentados como dos por una oclusión parcial.



- 1 Se aplica el método Canny a las imágenes $D(x, y)$ e $I(x, y)$



(a) $I(x, y)$



(b) $D(x, y)$



(c) Canny $I(x, y)$



(d) Canny $D(x, y)$



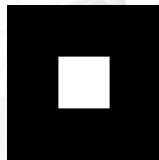
- 2 Por cada pixel (p, k) se inspecciona el entorno delimitado por un círculo de radio 2 en $B_i(x, y)$ y $B_d(x, y)$
- 3 Si al menos un pixel de contorno es encontrado en el entorno del pixel (p, k) en ambas imágenes. La imagen resultante tendrá un pixel de contorno.



(a) $I(x, y)$



(b) $I(x, y)$



(c) $I(x, y)$

Descripción del Método

Extracción de Bordes Inicial



- 4 Dada la imagen resultante R
- 5 $NR = \text{NOT}(R(x,y))$
- 6 Utilizando NR se localizan los componentes conectados W_i en la imagen de rango





- 1 Por cada componente se estima el plano que mejor describe a ese conjunto de puntos y se calcula SSE
- 2 Mientras $\sum_{i=1}^n SSE_{W_i} > umbral$
- 3 Se dilatan los bordes de R y se repite el proceso.



(a) Iteración 2



(b) Iteración 3



(c) Iteración 4



- 1 Se calcula para cada conjunto W_i
 - La media de intensidad.
 - La media de distancia.
 - El centroide de la nube de puntos.y se estima
 - El vector normal del plano que mejor que describe a la superficie
- 2 Se comparan los componentes conectados entre sí buscando la semejanza entre las características calculadas en el paso anterior. Si dos conjuntos resultan similares se unen.



Se segmentan los objetos correctamente y se encuentran unidos aquellos conjuntos de puntos que a priori pertenecían a conjuntos diferentes



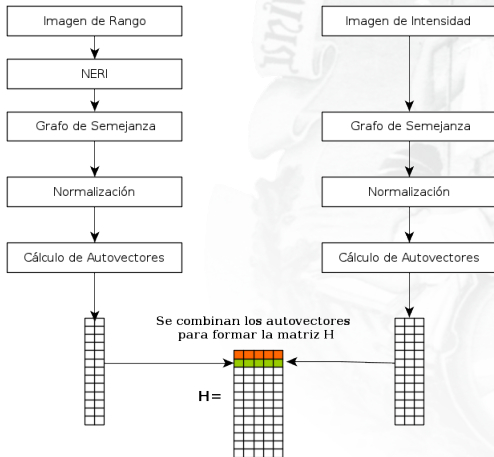


Clustering como un problema de particionado de grafos

- Utilizan los autovectores de las matrices derivadas de los datos
- Obtiene una representación de dimensión inferior que puede ser “clusterizada” fácilmente.

Método Propuesto

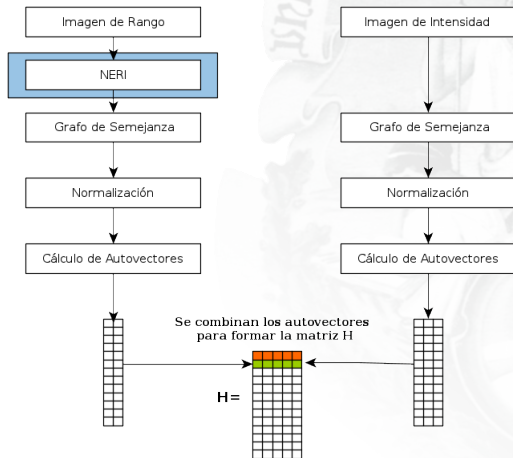
Descripción General



Se aplica un algoritmo de clustering sobre los patrones de H

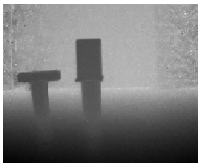
Método Propuesto

Descripción General

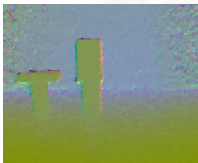




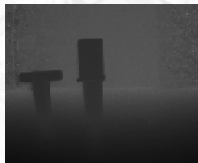
- Dada una imagen de rango de dimensión $m \times n$
- Por cada punto i de la imagen ubicado a una distancia d_i se estima el vector normal $n_i = [x_i \ y_i \ z_i]$
- Se construye una imagen RGB de dimensión $m \times n$ donde el pixel i tiene como componentes $[x_i \ y_i \ d]$
- Se convierte la imagen a escala de grises



(a) Rango



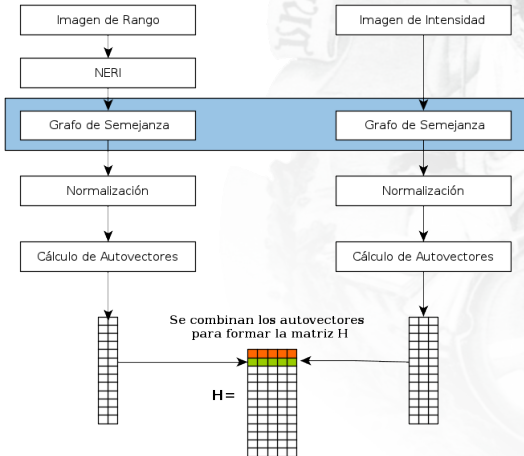
(b) Imagen RGB



(c) NERI

Método Propuesto

Grafo De Semejanza

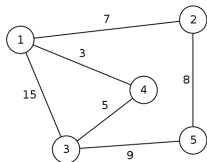


Grafo de Semejanza

Definición



- Dado un conjunto de patrones X
- Grafos pesados no dirigidos $G = (V, E)$
- $V = X$
- Los pesos de las aristas son proporcionales a la semejanza de los vértices que relacionan.
- Si dos vértices son completamente desemejantes no hay arista que los conecta.
- Representación Matricial



Matriz de pesos W

	1	2	3	4	5
1	0	7	15	3	0
2	7	0	0	0	8
3	15	0	0	5	9
4	3	0	5	0	9
5	0	8	9	0	0

$W(i,j) = w_{ij}$ es el peso de la arista que conecta i con j

Método Propuesto

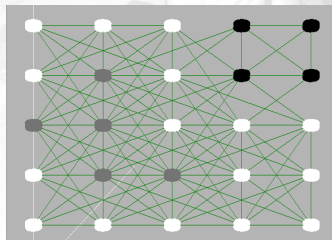
Grafos de semejanza de imágenes



- Cada pixel es un vértice
- Puede ser complicado elegir un peso entre aristas adecuado.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

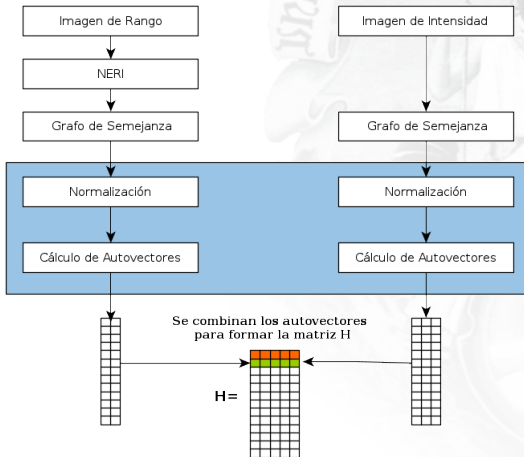
(a) Imagen de 5×5 píxeles



(b) Grafo generado a partir de la imagen

Método Propuesto

Clustering Espectral

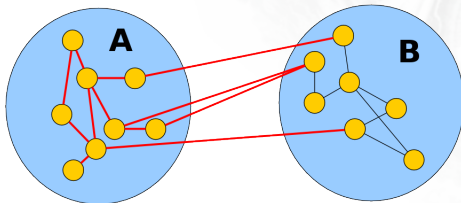




El grado de un vértice $d_i = \sum_{j \in V} w_{ij}$

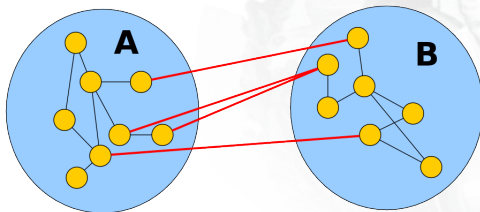


El volumen de un conjunto $\text{Vol}(A) = \sum_{i \in A} d_i$





El corte de un conjunto $\text{Cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$





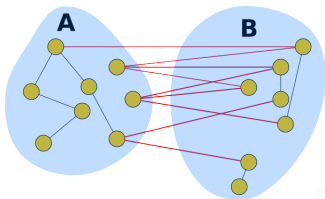
- El criterio de Corte Normalizado Multiclase es un criterio de partición para k clases.

$$N_k cut = \sum_{i=1}^k \frac{cut(A_i, \overline{A_i})}{Vol(A_i)}$$

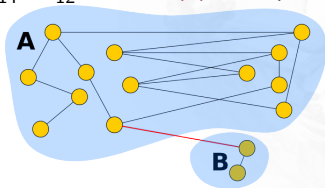
- El objetivo es encontrar los k conjuntos que minimicen el criterio.
- Maximiza la semejanza intra cluster y maximiza la desemejanza entre clusteres.

Clustering Espectral

Cortes Normalizado. Ejemplo



(a) $\text{Ncut}(A, B) = \frac{9}{14} + \frac{9}{12} = 0,69$ (b) $\text{Ncut}(A, B) = \frac{2}{9} + \frac{2}{10} = 0,42$

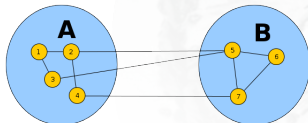


(c) $\text{Ncut}(A, B) = \frac{1}{14} + \frac{1}{2} = 0,57$



Podemos representar a un conjunto A utilizando un vector indicador escalado

$$x_A \in R^{|V|} \quad x_i = \begin{cases} \frac{1}{\text{Vol}(A)} & \text{Si } i \in A \\ 0 & \text{Si } i \notin A \end{cases}$$



$$x_A = \begin{bmatrix} \frac{1}{\text{Vol}(A)} & \frac{1}{\text{Vol}(A)} & \frac{1}{\text{Vol}(A)} & \frac{1}{\text{Vol}(A)} & 0 & 0 & 0 \end{bmatrix}$$
$$x_B = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{\text{Vol}(B)} & \frac{1}{\text{Vol}(B)} & \frac{1}{\text{Vol}(B)} \end{bmatrix}$$

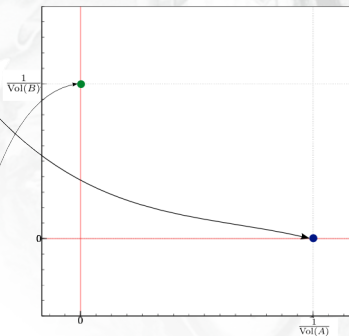
Clustering Espectral

Cortes Normalizados Discreto



$$[x_A \ x_B] =$$

$\frac{1}{\text{Vol}(A)}$	0
$\frac{1}{\text{Vol}(A)}$	0
$\frac{1}{\text{Vol}(A)}$	0
$\frac{1}{\text{Vol}(A)}$	0
0	$\frac{1}{\text{Vol}(B)}$
0	$\frac{1}{\text{Vol}(B)}$
0	$\frac{1}{\text{Vol}(B)}$





- Encontrar la matriz de vectores indicadores que minimicen N_{cut} es NP-Hard
- Aproximación:

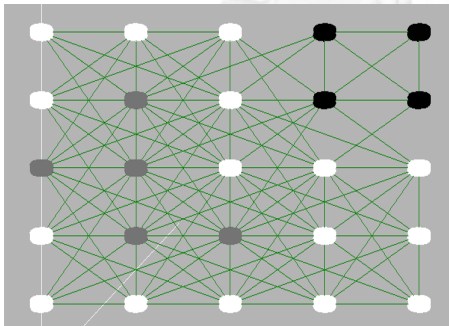
Calcular los autovectores correspondientes a los autovalores mas pequeños de

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

- W es la matriz del grafo de semejanza
 - D es una matriz diagonal con los grados de cada vertice
-
- Los vectores ya no son discretos



- Dado el grafo generado a partir de la imagen

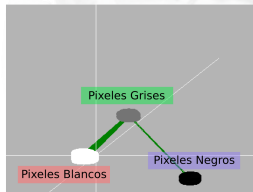


- Se construye la matriz $LN = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$



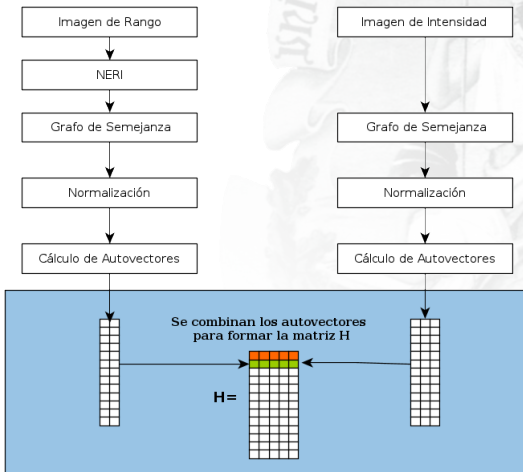
- Se calculan los autovectores correspondientes a los autovalores mas pequeños de LN

-0,6933	-0,3254
-0,6934	-0,3255
-0,6934	-0,3256
4,2683	-2,4846
4,2688	-2,4856
-0,6933	-0,3253
1,4920	3,4282
-0,6934	-0,3255
4,2678	-2,4835
4,2686	-2,4851
1,4916	3,4291
1,4919	3,4282
-0,6934	-0,3256
-0,6935	-0,3257
-0,6936	-0,3258
-0,6933	-0,3253
1,4918	3,4287
1,4919	3,4279
-0,6935	-0,3257
-0,6936	-0,3258
-0,6933	-0,3253
-0,6934	-0,3255
-0,6935	-0,3256
-0,6935	-0,3257
-0,6936	-0,3258



Método Propuesto

Descripción General





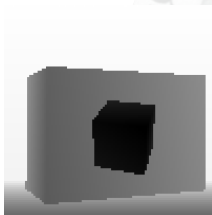
- Se genera la matriz H combinando autovectores de las dos imágenes
- Se aplica un algoritmo de clustering sobre los patrones utilizando los autovectores como características

Resultados Experimentales

Descripción General



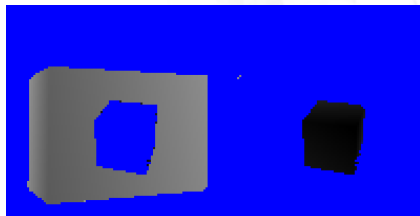
(a) Intensidad



(b) Rango



(c) NERI



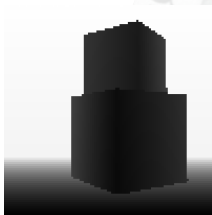
(d) Resultado de la Segmentación

Resultados Experimentales

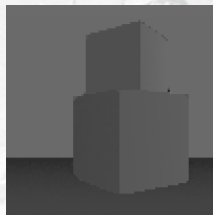
Descripción General



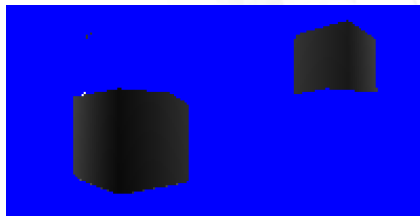
(a) Intensidad



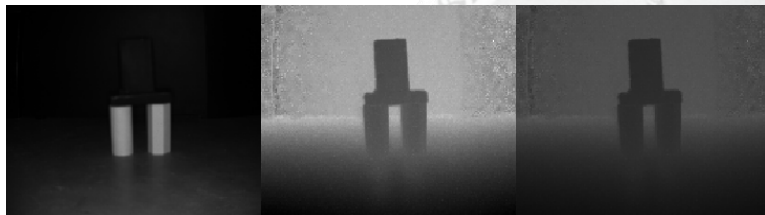
(b) Rango



(c) NERI



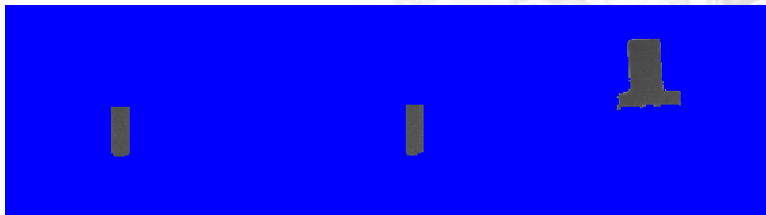
(d) Resultado de la Segmentación



(a) Intensidad

(b) Rango

(c) NERI



(d) Resultado de la Segmentación

El entorno SpectralGUI permite la aplicación de algoritmos espectrales de agrupamiento y la visualización de todas las etapas involucradas en el proceso. La aplicación consta de 5 librerías básicas y una interfaz de usuario.

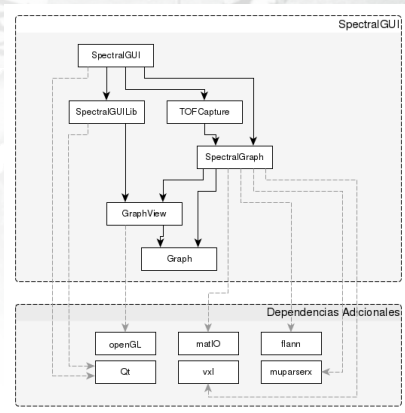


Figura: Diagrama de dependencias de SpectralGUI

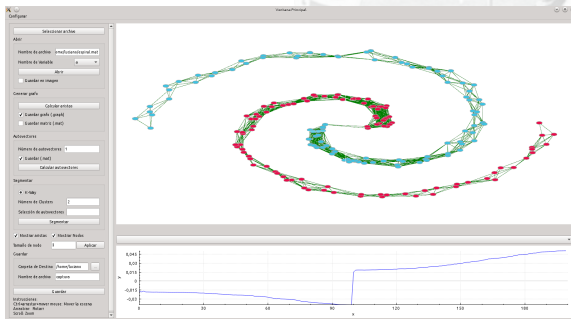


Figura: Captura de pantalla de SpectralGUI aplicado a un conjunto de datos de dimensión 2. El gráfico ubicado en la parte inferior representa a los valores del autovector correspondiente al segundo autovalor mas pequeño del grafo generado.

Esta ventana permite aplicar el algoritmo de clustering espectral propuesto sobre las imágenes de rango e intensidad.

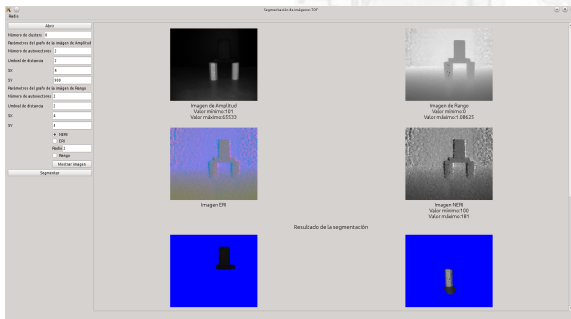


Figura: Captura de pantalla de SpectralGUI aplicado a una imagen de intensidad y de rango capturada con una cámara de tiempo de vuelo SwissRanger SR4000



- TOFCapture facilita la captura y documentación de imágenes
- El primer método propuesto mejora los resultados de la segmentación cerrando correctamente los contornos de los objetos y analizando oclusiones parciales
- El segundo método propuesto mejora la segmentación para objetos con niveles de intensidad o distancias similares
- SpectralGUI permite el análisis de los métodos estudiados



- Combinar la información de intensidad y profundidad en la función de pesos $w(i, j)$.
- Investigar la influencia de la función de semejanza elegida en los autovectores resultantes
- Comparar algoritmos de segmentación espectral
- Incorporar y comparar métodos de clustering alternativos