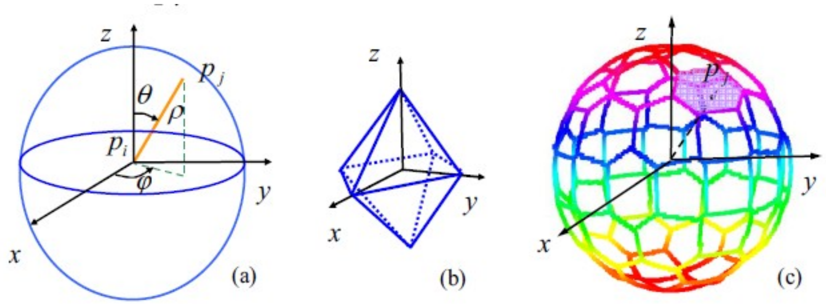
Intrinsic Shape Signatures

O cómo aprender a aprender



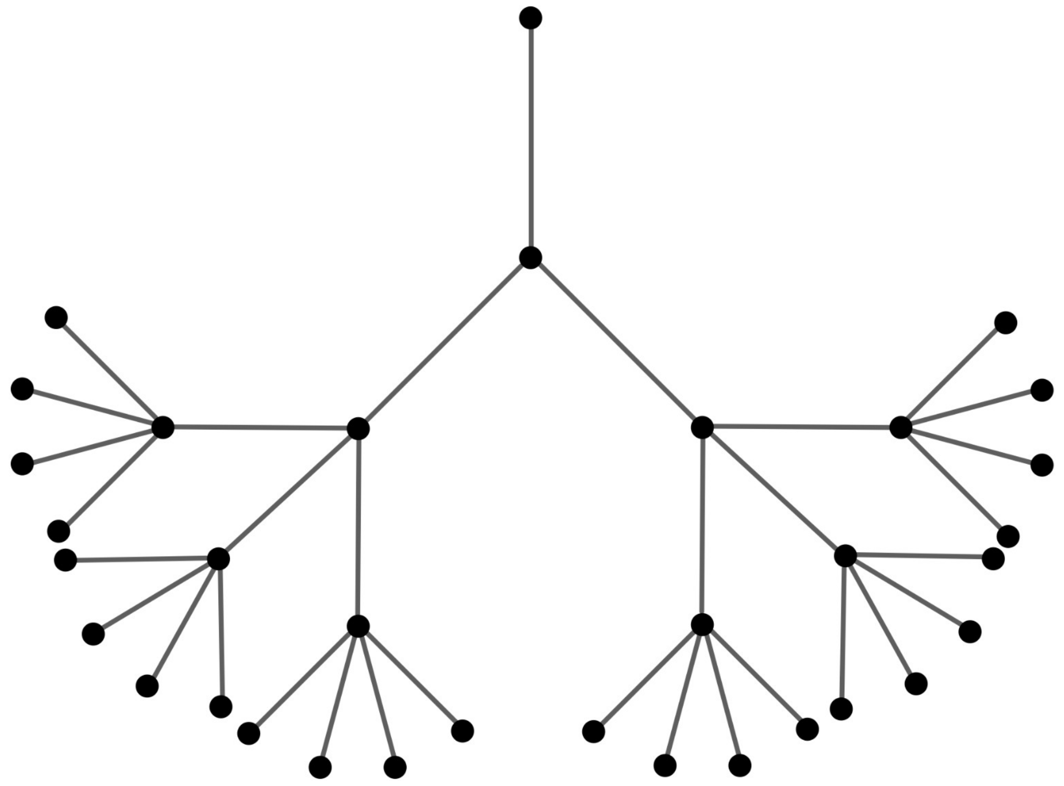
Enfoque top-down

¿Realmente me renta aprender esto?

¿Con qué profundidad?

Voy indagando de lo de más alto nivel a lo de más bajo nivel conforme lo voy necesitando.

- Rápido
- Óptimo



Paso número 1

Buscar información con criterio.

Cualquier mono con un portátil puede poner cosas en internet.

> cornerSubPix con el patrón de círculos?

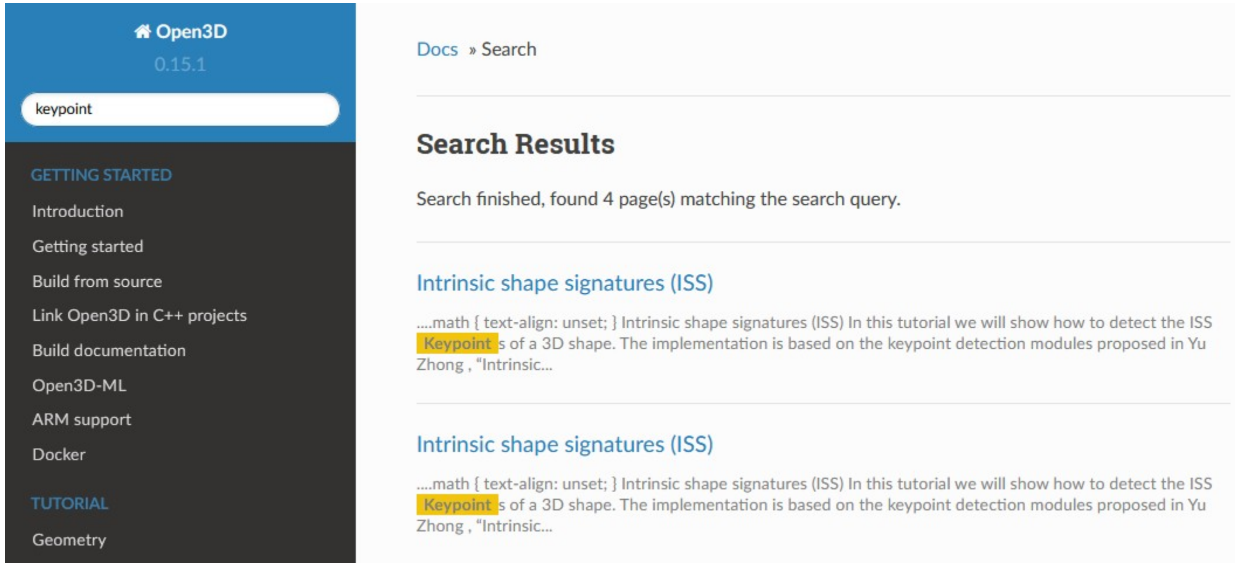
Pesar la fiabilidad en función de la fuente.

Paso número 1

¿Qué quiero hacer? ¿Buscar KPs?

Voy a la web de Open3D y busco“Keypoint”.

Sólo sale Intrinsic Shape Signatures.



Leer: ¿Qué me suena a chino?

Intrinsic Shape Signatures (ISS)

In this tutorial we will show how to detect the **ISS** Keypoints of a 3D shape. The implementation is based on the keypoint detection modules proposed in Yu Zhong , "Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition", 2009.

ISS Keypoints

ISS saliency measure is based on the Eigenvalue Decomposition (EVD) of the scatter matrix $\mathbf{\Sigma}(\mathbf{p})$ of the points belonging to the support of p, i.e.

$$\mathbf{\Sigma}(\mathbf{p}) = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}(\mathbf{q}-\mu_{\mathbf{p}})(\mathbf{q}-\mu_{\mathbf{p}})^T \quad \text{with}$$
$$\mu_{\mathbf{p}} = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}\mathbf{q}$$

Given $\mathbf{\Sigma}(\mathbf{p})$, its eigenvalues in decreasing magnitude order are denoted here as $\lambda_1, \lambda_2, \lambda_3$. During the pruning stage, points whose ratio between two successive eigenvalues is below a threshold are retained:

$$\frac{\lambda_2(\mathbf{p})}{\lambda_1(\mathbf{p})} < \gamma_{12} \wedge \frac{\lambda_3(\mathbf{p})}{\lambda_2(\mathbf{p})} < \gamma_{23}$$

subsequent description stage can hardly turn out effective. Among remaining points, the saliency is determined by the magnitude of the smallest eigenvalue

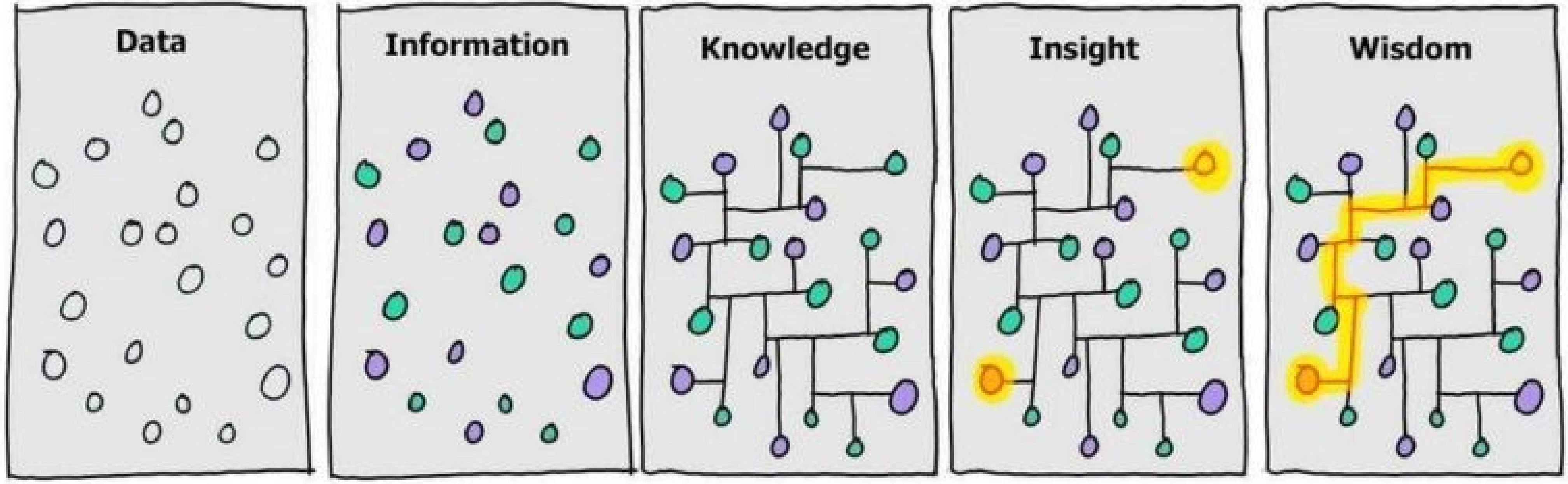
$$\rho(\mathbf{p}) \doteq \lambda_3(\mathbf{p})$$

So as to include only points with large variations along each principal direction.

After the detection step, a point will be considered a **keypoint** if it has the maxium saliency value on a given neighborhood.

NOTE: For more details please reffer to the original publication or to "Performance Evaluation of 3D Keypoint Detectors", by Tombari et.al.

Muchos de vosotros os habéis quedado aquí.



Todo el mundo sabe aprenderse de memoria la fórmula y la descripción como un loro…

Pero qué es? Qué significa? Cuál es la consecuencia? Qué efecto tienen los parámetros?

Leer: ¿Qué me suena a chino?

Intrinsic Shape Signatures (ISS)

In this tutorial we will show how to detect the **ISS** Keypoints of a 3D shape. The implementation is based on the keypoint detection modules proposed in Yu Zhong , "Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition", 2009.

ISS Keypoints

ISS saliency measure is based on the Eigenvalue Decomposition (EVD) of the scatter matrix $\mathbf{\Sigma}(\mathbf{p})$ of the points belonging to the support of p, i.e.

$$\mathbf{\Sigma}(\mathbf{p}) = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}(\mathbf{q}-\mu_\mathbf{p})(\mathbf{q}-\mu_\mathbf{p})^T \quad \text{with}$$
$$\mu_\mathbf{p} = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}\mathbf{q}$$

Given $\mathbf{\Sigma}(\mathbf{p})$, its eigenvalues in decreasing magnitude order are denoted here as λ_1, λ_2, λ_3. During the pruning stage, points whose ratio between two successive eigenvalues is below a threshold are retained:

$$\frac{\lambda_2(\mathbf{p})}{\lambda_1(\mathbf{p})} < \gamma_{12} \wedge \frac{\lambda_3(\mathbf{p})}{\lambda_2(\mathbf{p})} < \gamma_{23}$$

subsequent description stage can hardly turn out effective. Among remaining points, the saliency is determined by the magnitude of the smallest eigenvalue

$$\rho(\mathbf{p}) \doteq \lambda_3(\mathbf{p})$$

So as to include only points with large variations along each principal direction.

After the detection step, a point will be considered a **keypoint** if it has the maxium saliency value on a given neighborhood.

NOTE: For more details please reffer to the original publication or to "Performance Evaluation of 3D Keypoint Detectors", by Tombari et.al.

Leer: ¿Por dónde sigo?

Intrinsic Shape Signatures (ISS)

In this tutorial we will show how to detect the **ISS** Keypoints of a 3D shape. The implementation is based on the keypoint detection modules proposed in Yu Zhong , "Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition", 2009.

ISS Keypoints

ISS saliency measure is based on the Eigenvalue Decomposition (EVD) of the scatter matrix $\mathbf{\Sigma}(\mathbf{p})$ of the points belonging to the support of $\boldsymbol{p}$, i.e.

$$\mathbf{\Sigma}(\mathbf{p}) = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}(\mathbf{q}-\mu_{\mathbf{p}})(\mathbf{q}-\mu_{\mathbf{p}})^{T} \quad \text{with}$$
$$\mu_{\mathbf{p}} = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}\mathbf{q}$$

Given $\mathbf{\Sigma}(\mathbf{p})$, its eigenvalues in decreasing magnitude order are denoted here as λ_1, λ_2, λ_3. During the pruning stage, points whose ratio between two successive eigenvalues is below a threshold are retained:

$$\frac{\lambda_2(\mathbf{p})}{\lambda_1(\mathbf{p})} < \gamma_{12} \wedge \frac{\lambda_3(\mathbf{p})}{\lambda_2(\mathbf{p})} < \gamma_{23}$$

subsequent description stage can hardly turn out effective. Among remaining points, the saliency is determined by the magnitude of the smallest eigenvalue

$$\rho(\mathbf{p}) \doteq \lambda_3(\mathbf{p})$$

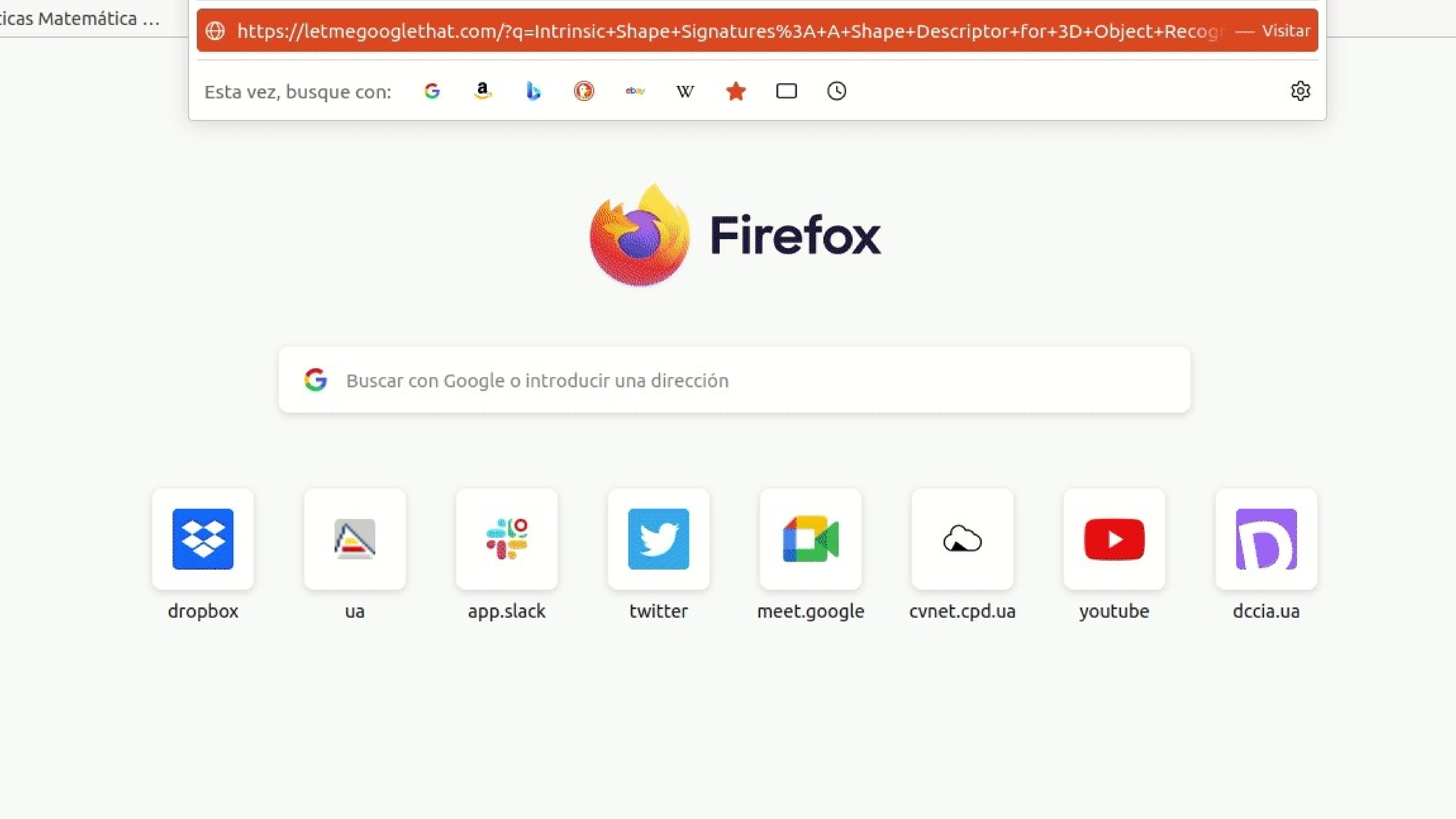
So as to include only points with large variations along each principal direction.

After the detection step, a point will be considered a **keypoint** if it has the maxium saliency value on a given neighborhood.

NOTE: For more details please reffer to the original publication or to "Performance Evaluation of 3D Keypoint Detectors", by Tombari et.al.

Más fuentes:

- Paperswithcode
- Google Scholar
- Arxiv
- Google



icas Matemática ...
https://letmegooglethat.com/?q=Intrinsic+Shape+Signatures%3A+A+Shape+Descriptor+for+3D+Object+Recog
— Visitar
Esta vez, busque con:
Firefox
Buscar con Google o introducir una dirección
dropbox
ua
app.slack
twitter
meet.google
cvnet.cpd.ua
youtube
dccia.ua

Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition

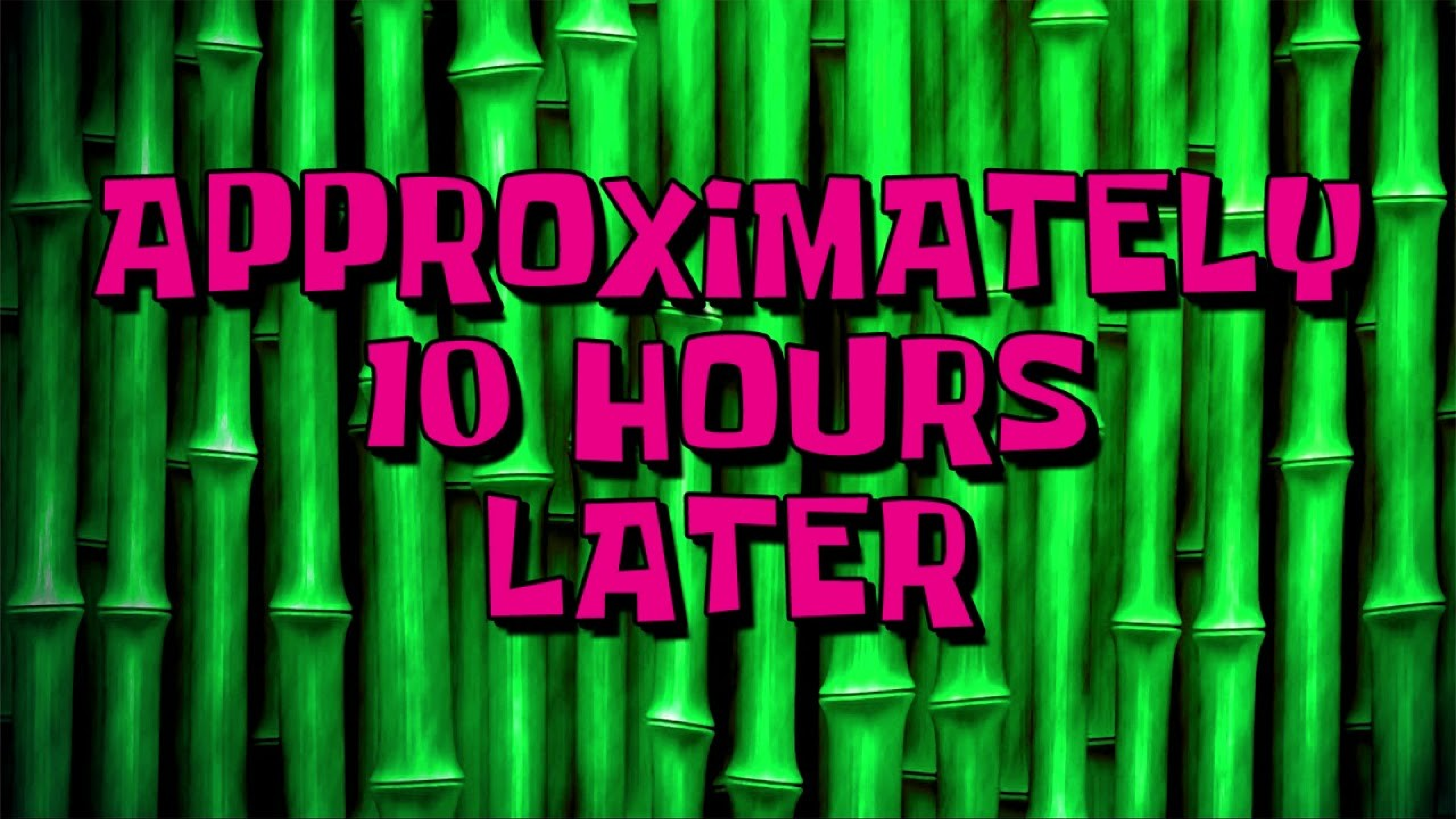
Yu Zhong
AIT, BAE Systems
6 New England Executive Park
Burlington, MA 01803-5012 USA
yu.zhong@baesystems.com

Abstract

This paper presents a new approach for recognition of 3D objects that are represented as 3D point clouds. We introduce a new 3D shape descriptor called Intrinsic Shape Signature (ISS) to characterize a local/semi-local region of a point cloud. An intrinsic shape signature uses a view-independent representation of the 3D shape to match shape patches from different views directly, and a view-dependent transform encoding the viewing geometry to facilitate fast pose estimation. In addition, we present a highly efficient indexing scheme for the high dimensional ISS shape descriptors, allowing for fast and accurate search of large model databases. We evaluate the performance of the proposed algorithm on a very challenging task of recognizing different vehicle types using a database of 72 models in the presence of sensor noise, obscuration and scene clutter.

1. Introduction

the idea of 2D shape contexts to 3D. A 3D shape context (3DSC) at an oriented basis point is a 3D occupational histogram of the data points in a surrounding support sphere, with its north pole aligned to the surface normal. However, given only the surface normal as a reference, there is a gauge of freedom in the rotation around the axes that needs to be eliminated in order to define the 3D histogram. This problem is worked around by uniformly sampling the reference rotation angle and computing one feature vector for each sample. This handling of the free rotation multiplies the computational and storage cost, and decreases the recognition performance due to the limited sampling of the rotation parameter. Mian et al. [15] also used feature descriptors maintaining 3D shape information to match surface meshes. They defined a 3D reference frame for a pair of oriented points (a vertex and its surface normal), and then computed a "tensor", which is a Cartesian partition of the cubic volume centered at the origin of the defined frame. The shape feature consists of the intersected object surface area in each bin. The drawback of this


APPROXIMATELY
10 HOURS
LATER

Comprender la información

Se define un sistema de referencia local (intrinsic) F_i a partir de un punto p_i con un radio de soporte r_{frame}. Usando el análisis Eigen de matriz de dispersión del punto tenemos:

1- Calcular un peso w_i para cada punto p_i inversamente proporcional a la densidad de los puntos dentro de la vecindad r_{frame}

2- Computar u … derada cov(p_i) usando todos … o r_{frame}

$$COV(p_i) = \sum_{|p_j - p_i| < r_{frame}} w_j (p_j - p_i)(p_j - p_i)^T \Big/ \sum_{|p_j - p_i| < r_{frame}} w_j$$

Comprender la información

3- Calcular los Eigenvalues λ^1_i , λ^2_i , λ^3_i y los Eigenvectors e^1_i , e^2_i , e^3_i en orden de magnitud decreciente

4- Usar p_i como el origen del sistema F_i y e^1_i , e^2_i , $(e^1_i \times e^2_i)$ como los ejes x, y, z.

Luego dice más cosas, pero de cómo aplicar esta idea a generar un descriptor, que no nos interesa.

¿PERO ES UN DESCRIPTOR O UN DETECTOR DE KPS AL FINAL?

Matriz de dispersión?

- Scatter matrix vs covariance matrix

La matriz de dispersión es una aproximación a la matriz de covarianza que se usa cuando calcular esta última es demasiado costoso.

Se usa indistintamente porque el resultado debe ser el mismo (o muy parecido).

Qué representa la matriz de dispersión? Cómo se interpreta?

> Me da igual

Eigenvectors y Eigenvalues?

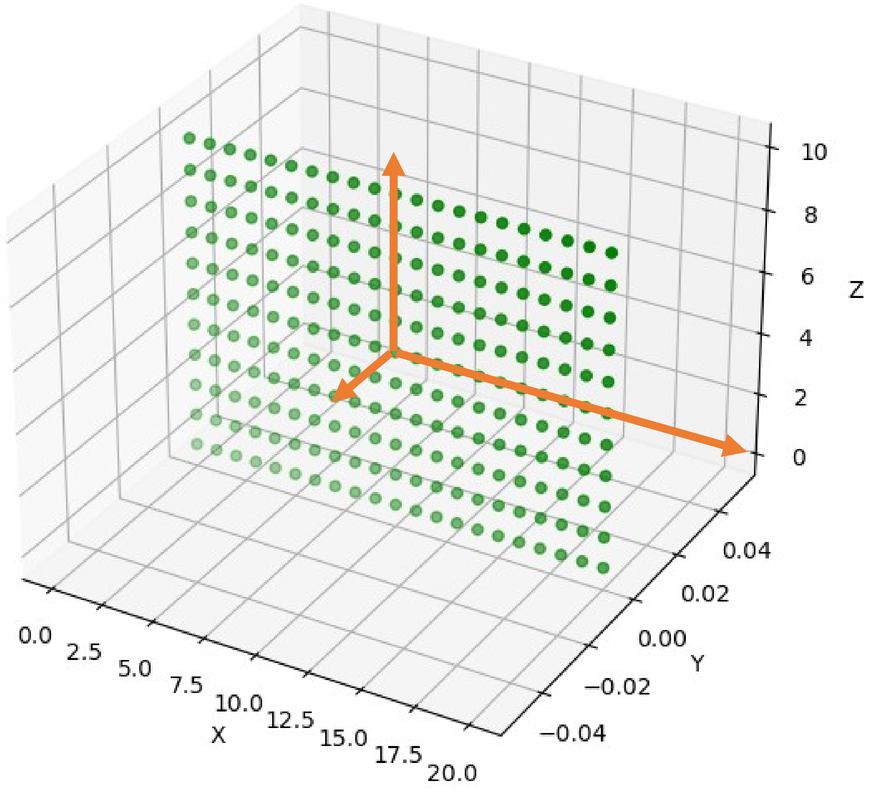
Cálculo de las direcciones principales que describen los datos.

En este caso tenemos 3 direcciones principales que, ordenadas de forma decreciente por su magnitud, corresponderían con los vectores:

(1,0,0) – 61.66
(0,0,1) – 10
(0,1,0) – 0

Este ejemplo lo he programado yo, para ver qué es lo que pasa, qué resultados tengo con diferentes superficies y si da lo que yo creo que da.

* Nota que el plano está alineado con los ejes Z y Y por esos lo vectores apuntan en esas direcciones.



Insight

Lo que hace el método es crear un sistema de referencia local a cada uno de los puntos de la nube. Para ello, se selecciona un radio de vecindad y se obtienen los vectores de las 3 direcciones principales (eigenvectors) y su magnitud (eigenvalues). Indica que las ordena en orden decreciente y que usa los dos de mayor magnitud como ejes x e y, y el tercero viene del producto vectorial entre esos dos, es decir, el vector normal al plano que forman. Se genera así un sistema de referencia, con sus 3 vectores perpendiculares entre sí.

Ya, pero cómo se si es un keypoint o no? Qué tiene que ver el tema del marco de referencia con lo que yo estoy haciendo?

> Volvemos a la web de Open3D

Alguien me está mintiendo

En Open3D

$$\boldsymbol{\Sigma}(\mathbf{p}) = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}(\mathbf{q}-\mu_{\mathbf{p}})(\mathbf{q}-\mu_{\mathbf{p}})^T \quad \text{with}$$

Usa la media de los puntos

$$\mu_{\mathbf{p}} = \frac{1}{N}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}\mathbf{q}$$

No hay ponderaciones

pero sí está normalizada con el número de vecinos

En el paper de ISS

$$COV(p_i) = \sum_{|p_j-p_i|<r_{frame}} w_j(p_j-p_i)(p_j-p_i)^T / \sum_{|p_j-p_i|<r_{frame}} w_j$$

Usa el punto actual

La matriz está ponderada

y normalizada usando los pesos

Alguien me está mintiendo

Son implementaciones diferentes!

Lo importante es la idea que subyace.

Cada implementación puede introducir variantes para ser más óptimo, más preciso, aplicarlo a un caso particular...

Sigo leyendo

Given $\mathbf{\Sigma}(\mathbf{p})$, its eigenvalues in decreasing magnitude order are denoted here as λ_1, λ_2, λ_3. During the pruning stage, points whose ratio between two successive eigenvalues is below a threshold are retained:

$$\frac{\lambda_2(\mathbf{p})}{\lambda_1(\mathbf{p})} < \gamma_{12} \wedge \frac{\lambda_3(\mathbf{p})}{\lambda_2(\mathbf{p})} < \gamma_{23}$$

The rationale is to avoid detecting keypoints at points exhibiting a similar spread along the principal directions, where a repeatable canonical reference frame cannot be established and, therefore, the subsequent description stage can hardly turn out effective. Among remaining points, the saliency is determined by the magnitude of the smallest eigenvalue

$$\rho(\mathbf{p}) \doteq \lambda_3(\mathbf{p})$$

So as to include only points with large variations along each principal direction.

After the detection step, a point will be considered a **keypoint** if it has the maxium saliency value on a given neighborhood.

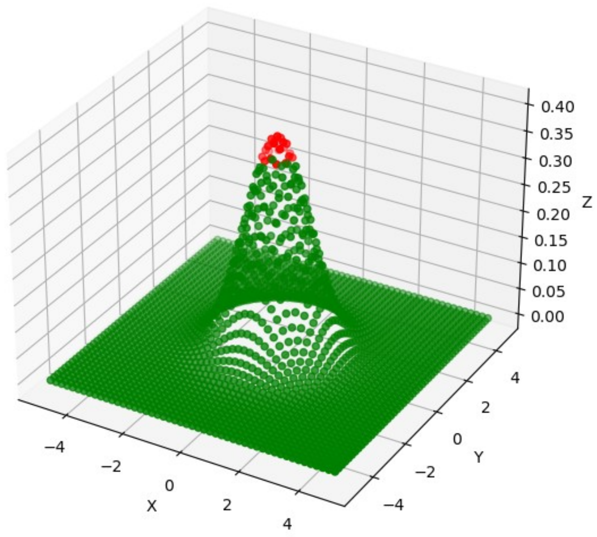
Se divide la magnitud del segundo con respecto del primer eigenvalue.
Y el tercero con respecto del segundo.

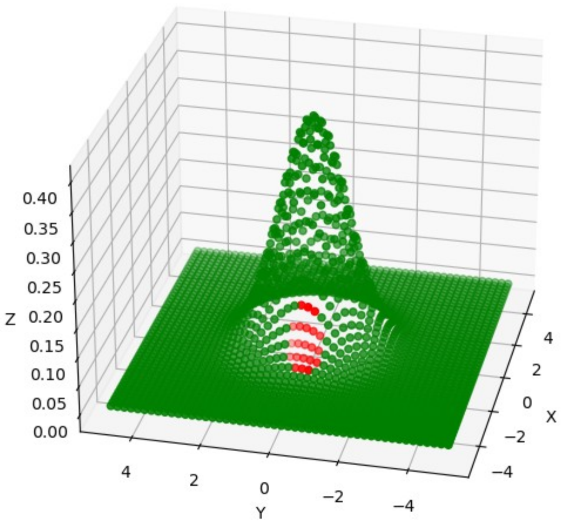
Como están ordenados de mayor a menor, el segundo nunca será mayor que el primero, como máximo será igual. Como resultado tenemos un ratio entre 0 y 1 que indica cómo de parecidos son las magnitudes de esos vectores. 1 es que son iguales, un número muy pequeño es que son muy diferentes. Lo mismo con el otro término.

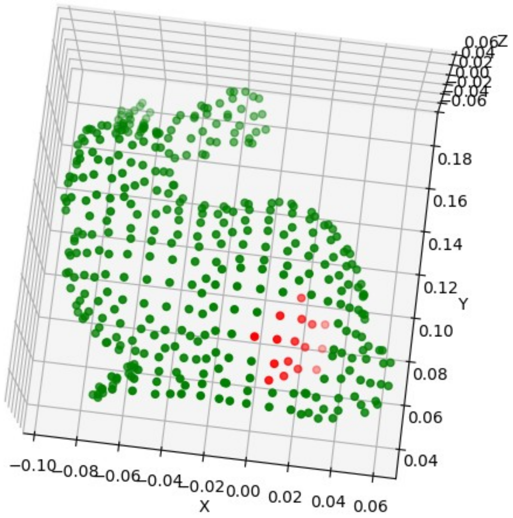
Hay dos parámetros γ_{12} y γ_{23} que controlan ese ratio. Lo que tú quieres es ajustar esos parámetros para que si ambos ratios están por debajo, eso indique que los 3 eigenvalues son diferentes. Esto es, que ese punto es un punto interesante. Que se encuentra en una zona no monótona, no plana.

Insight

Eigenvalues y ratios







Eigenvalues: 8.35, 8.33, 0.12
ratio1: 0.99, ratio2: 0.01

Eigenvalues: 11.95, 8.33, 0.005
ratio1: 0.69, ratio2: 0.0006

Eigenvalues: 0.006, 0.001, 0.0003
ratio1: 0.19, ratio2: 0.24

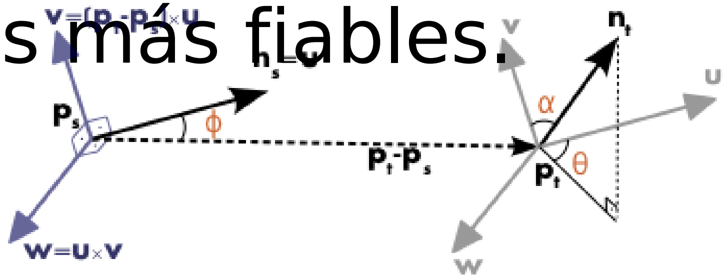
Cuál de estos 3 es mejor KP según ISS?

Insight

Lo que pretende ISS es detectar aquellos puntos cuyas magnitudes entre las direcciones principales sufran mucha diferencia.

Con la intención de generar sistemas de coordenadas únicos, que no tengan ambigüedad y que sean repetibles entre 2 nubes del mismo objeto.

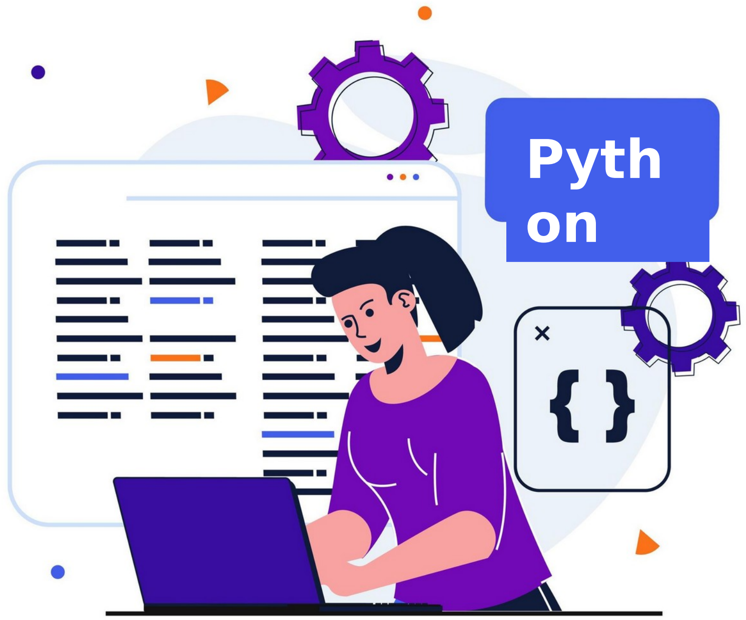
Esto lo hace porque los descriptores hacen uso del cálculo de sistemas de coordenadas locales. Cuanto más robustos, mejor para generar descriptores más fiables.



Ya lo domino totalmente?

Impleméntalo.

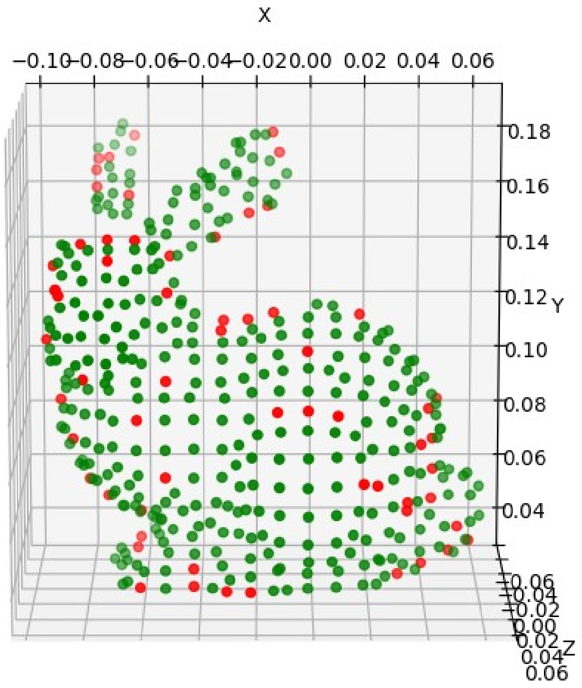
Si todavía tienes puntos ciegos,
así los descubrirás.



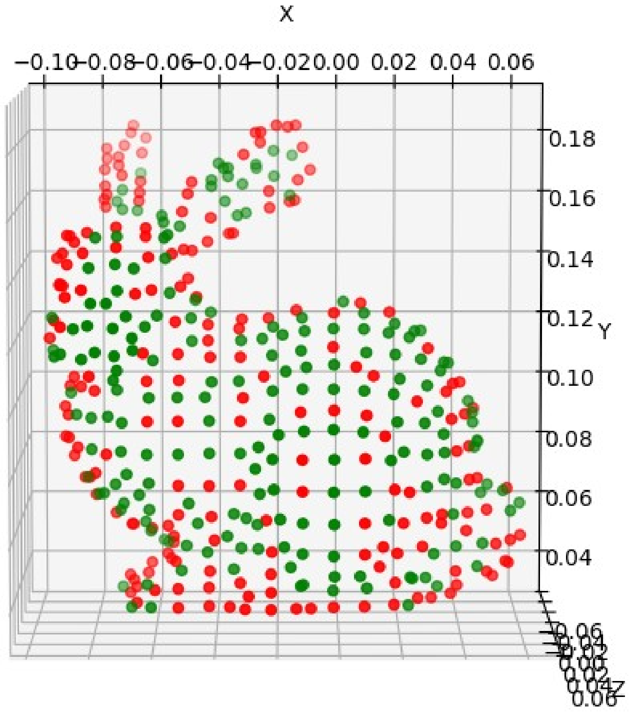
Implementación del método

```python
# step 1 - method implementation

gamma12 = 0.2
gamma23 = 0.5

kp_idx = []
saliencies = []
for m in range(len(pcd)):

    # get neighbors within a radius
    idxs = getNbyRadius(pcd[m], pcd, 0.01) # (pcd[66], pcd, 2.5) # cuspide 1275, falda 1215

    # if i cant reach 3 neighbors to compute the planes, skip
    if len(idxs) < 3:
        saliencies.append(-1)
        continue

    # compute covariance matrix
    mu = getMeanFromIndices(pcd, idxs)
    print("Centroid of the neighbors:", mu)

    neigbors_pcd = pcd[idxs]
    print("# of Neighbors:", len(neigbors_pcd))

    cov_mat = (neigbors_pcd-mu).T.dot((neigbors_pcd-mu))/float(len(neigbors_pcd))
    #print(cov_mat)

    u,s,vh = np.linalg.svd(cov_mat)
    print("Eigenvalues", s)
    print("Eigenvectors", vh)

    ratio1 = s[1]/s[0]
    ratio2 = s[2]/s[1]

    saliencies.append(s[2])

    print("Ratio 1", ratio1)
    print("Ratio 2", ratio2)

    if ratio1 < gamma12 and ratio2 < gamma23:
```

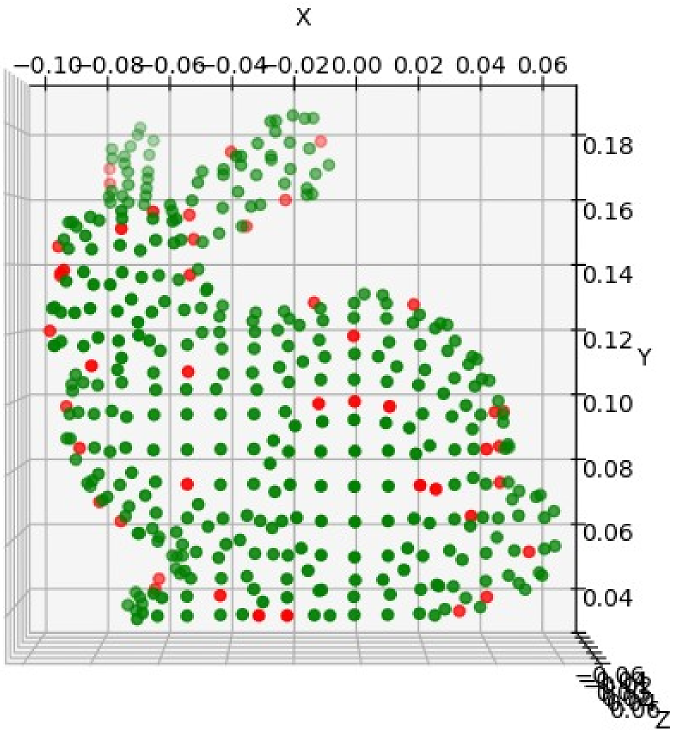
Resultados



$\gamma_{12} = 0.25$, $\gamma_{23} = 0.25$



$\gamma_{12} = 0.5$, $\gamma_{23} = 0.2$



$\gamma_{12} = 0.2$, $\gamma_{23} = 0.5$

Sigo leyendo

Among remaining points, the saliency is determined by the magnitude of the smallest eigenvalue

$$\rho(\mathbf{p}) \doteq \lambda_3(\mathbf{p})$$

So as to include only points with large variations along each principal direction.

After the detection step, a point will be considered a **keypoint** if it has the maxium saliency value on a given neighborhood.

Se aplica un proceso adicional de non maxima supression.

Se comprueba, para cada KP, si hay otros KPs en un radio determinado. Si los hay, se elige el que tenga el máximo saliency value.

Esto se hace para evitar que hayan muchos KPs en el mismo espacio 3D.

Interfaz de Open3D

```
keypoints =
o3d.geometry.keypoint.compute_iss_keypoints(
        pcd,
        salient_radius=0.005,
        non_max_radius=0.005,
        gamma_21=0.5,
        gamma_32=0.5)
```

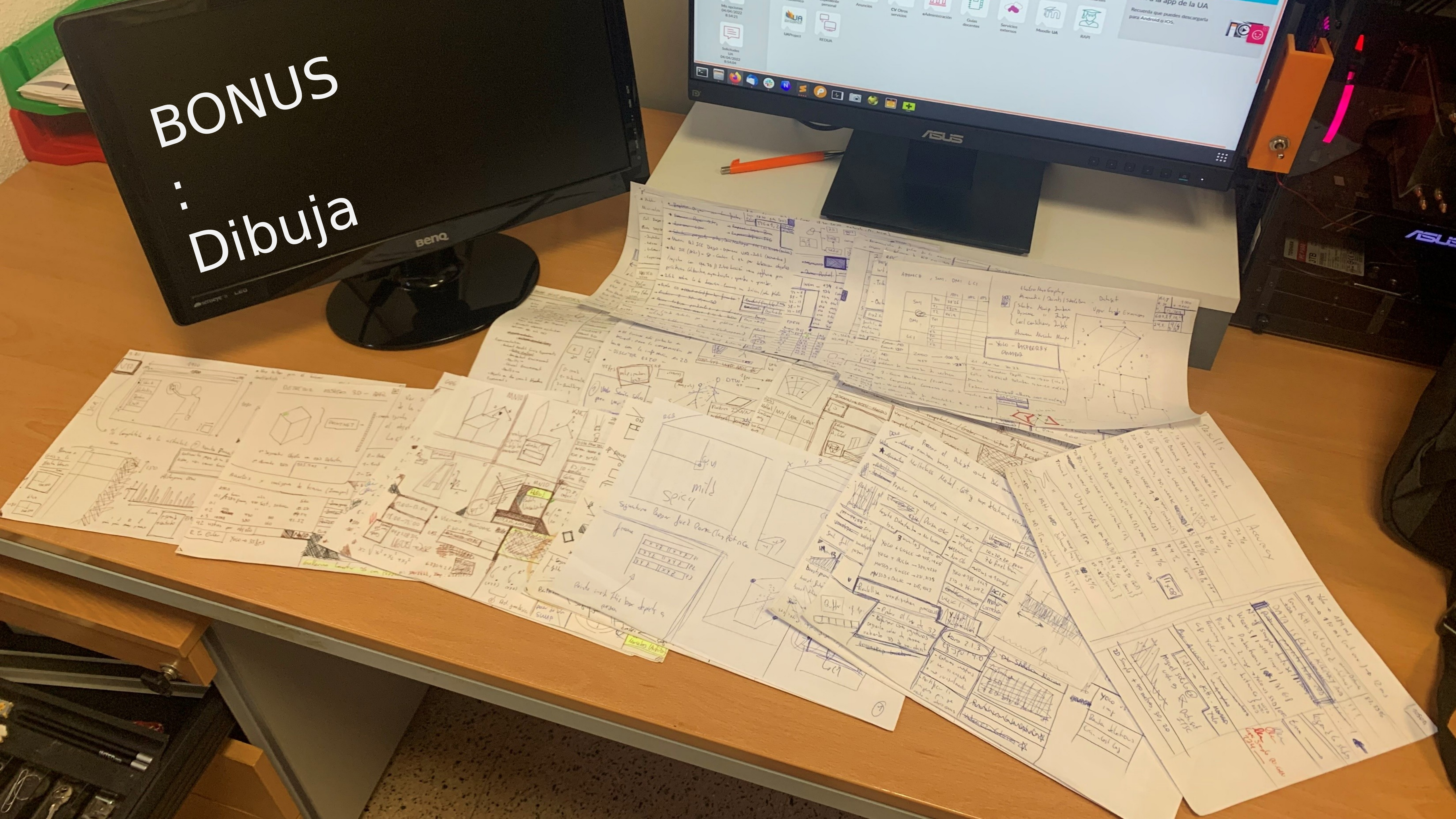
La nube de puntos de entrada

Radio para el calculo de F

Radio para el NMS

Umbral primer ratio, γ_{12}

Umbral segundo ratio, γ_{23}


BONUS
:
Dibuja

Deberes:

Ahora hazlo tú con FPFH

$$FPFH(\boldsymbol{p}_q) = SPFH(\boldsymbol{p}_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_i}\cdot SPFH(\boldsymbol{p}_i)$$