

Sistema de localización en espacios interiores para un robot móvil utilizando nubes de puntos y modelos ocultos de Markov

Para optar por el grado de:
Maestro en Ingeniería

Presenta:
Carlos Adrián Sarmiento Gutiérrez

Tutor principal:
Dr. Jesús Savage Carmona

Sistema de localización en espacios interiores para un robot móvil

- Introducción
- Objetivo
- Marco teórico
- Implementación del sistema
- Pruebas
- Resultados
- Conclusiones y trabajo futuro

Introducción

- En la última década los robots han comenzado a ser aplicados fuera del ramo industrial.
- La expansión de su campo de aplicación a dado lugar a una nueva rama de la robótica llamada ***robótica de servicio***.
- La federación internacional de robótica define un ***robot de servicio*** como:

*“Un robot que opera de manera **automática** o **semiautomática** para realizar servicios útiles al bienestar de los humanos o a su equipamiento, excluyendo las operaciones de fabricación.”*

- Se diseñan con capacidades de convivir con personas y de ejecutar tareas comunes que realizan las mismas.
- Son **autónomos**, es decir, que el robot pueda cumplir con sus objetivos sin intervención de un ser humano (*robot móvil autónomo*).



Problemas básicos

- Navegación
- Se define como la planeación de las acciones para lograr llegar a un lugar determinado en un ambiente.

● Localización

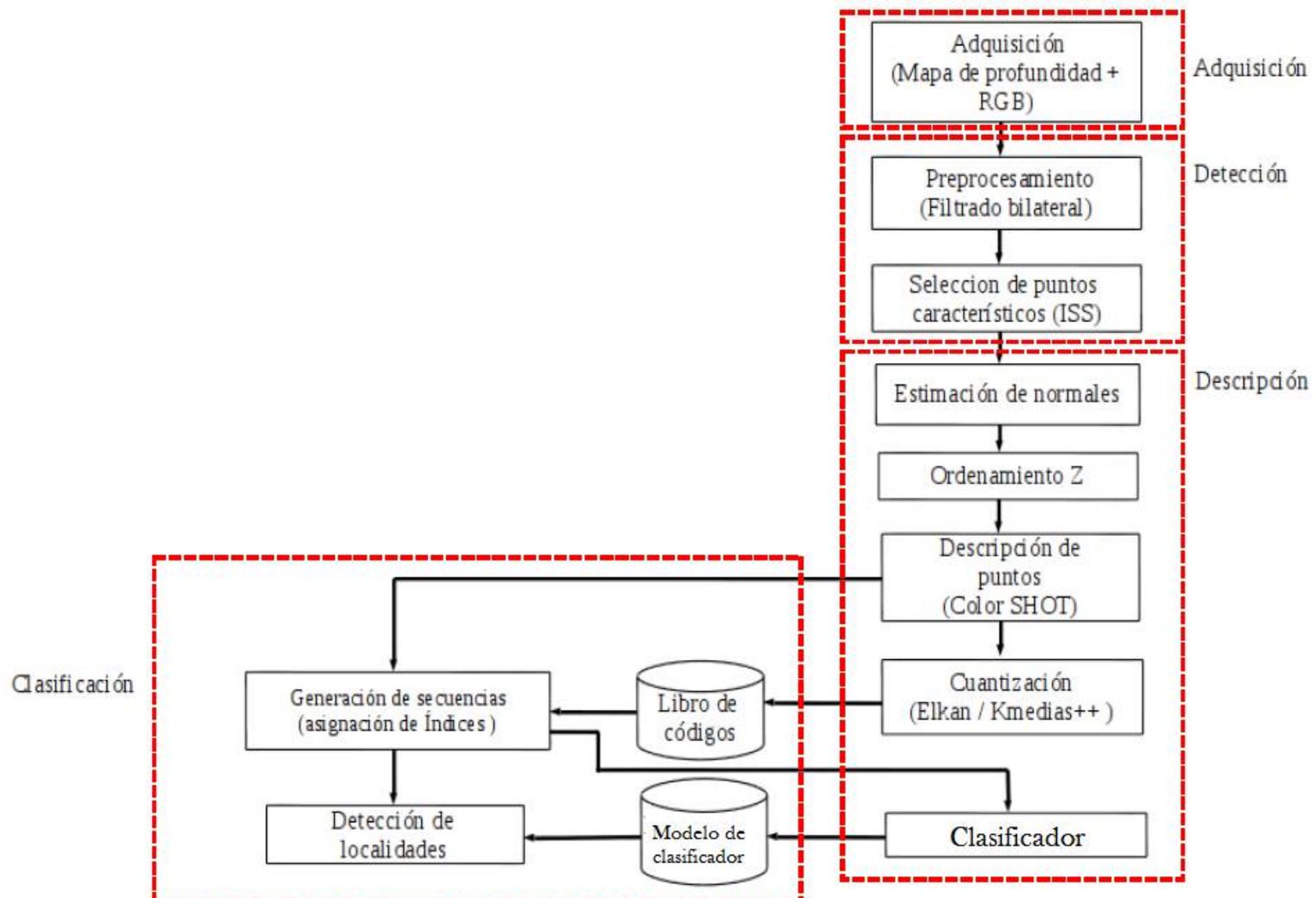
- Se define como la estimación de la posición de un robot móvil.
- Es la parte fundamental de los sistemas de navegación en un robot autónomo.

Objetivo

- Diseñar e implementar un sistema de localización para un robot móvil de servicio basado en información visual de color y de forma, utilizando modelos ocultos de Markov.

Marco teórico

- Sistema de localización propuesto



Etapa de adquisición

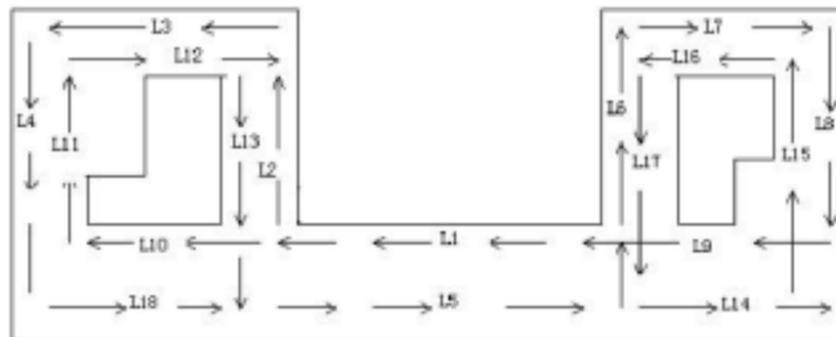


Sistemas de navegación basados en visión por computadora

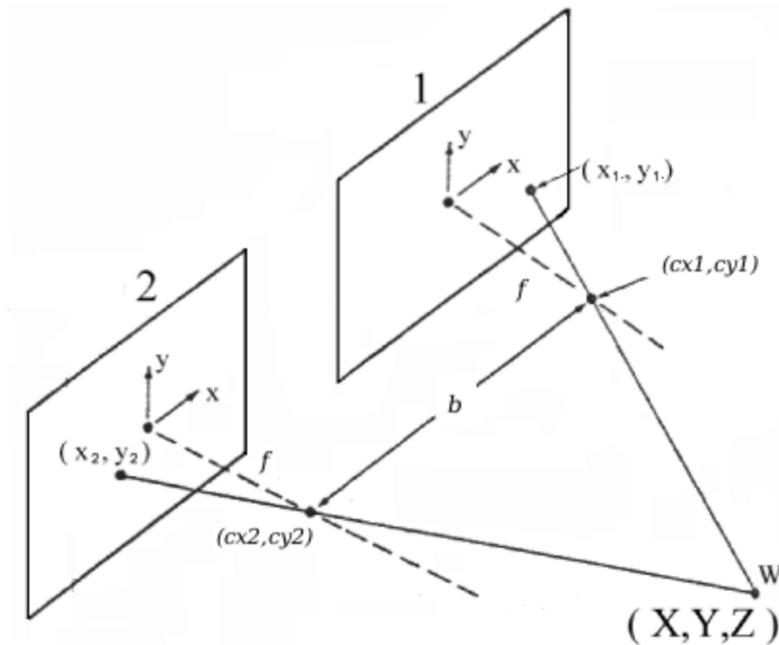
- La tecnología de visión por computadora es por mucho la selección óptima para sistemas de navegación en robots móviles, debido a la relación de la información proporcionada contra los costos del sistema.
- Los sistemas de visión resultan adecuados ya que la percepción visual ofrece mucha información útil del entorno y aumenta el alcance de las aplicaciones.
- Un robot puede ser capaz de navegar, detectar y manipular objetos, reconocer personas y lugares, con tan solo la información proveniente de su sistema de visión.
- Sistemas bioinspirados.

- Existen sistemas basados en radar, en ultrasonido o en láseres.
- Sistemas más complejos
 - Radar (interferencia electromagnética, guías de onda, mayor costo).
 - Láseres (sistemas ópticos muy susceptibles; espejos, rotores, etc).
 - Ultrasonido (mucho ruido en la adquisición, poco alcance).

- Ejemplos de sistemas de navegación basados en visión



Visión estéreo



$$m(x, y) = (x, y) - (x', y') = \frac{bf}{Z}$$

$$X = \frac{(u - c_x) Z}{f}$$

$$Y = \frac{(v - c_y) Z}{f}$$

$$Z = \frac{(u, v)}{K}$$

- Los dispositivos ópticos no son perfectos (los lentes tienen defectos).
- Los parámetros de la cámara se estiman a través del proceso de calibración y corrigen los defectos en la captura.

- Parámetros intrínsecos de la cámara:

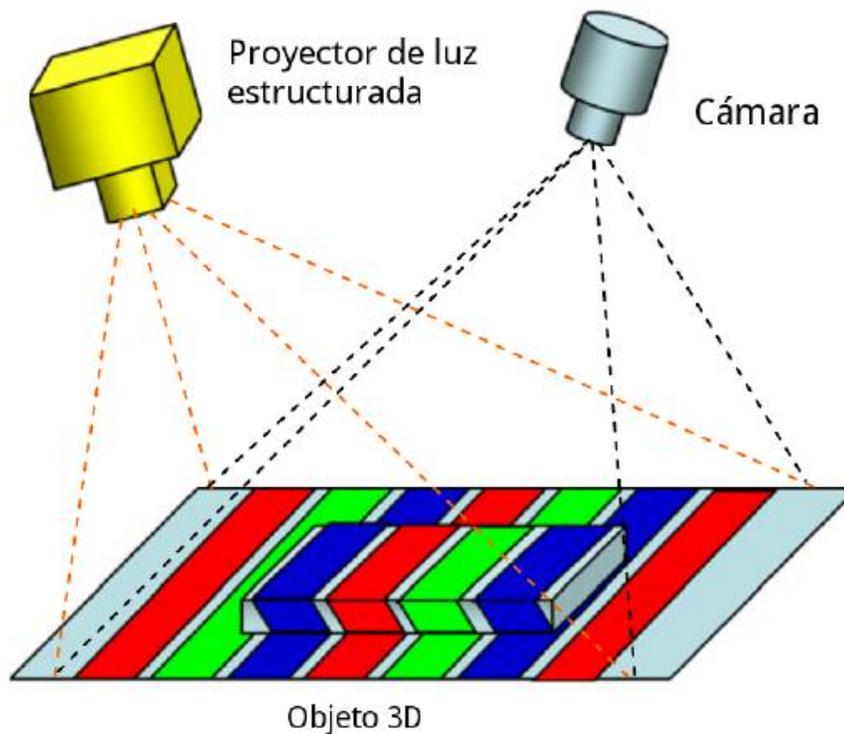
$$c_x, c_y, f=F(f_x, f_y), K$$

- Parámetros de distorsión:

$$d_0, d_1, d_2, d_3, d_4$$

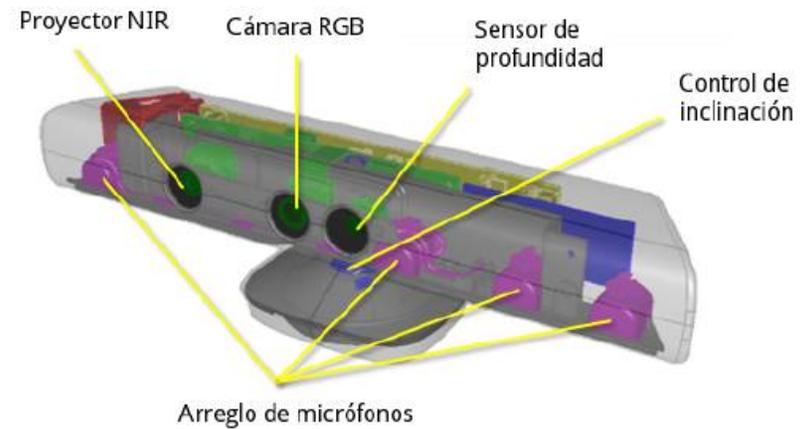
Visión estéreo con luz estructurada

- Es posible saber de manera predeterminada la disparidad de un región específica, dado un patrón predefinido.



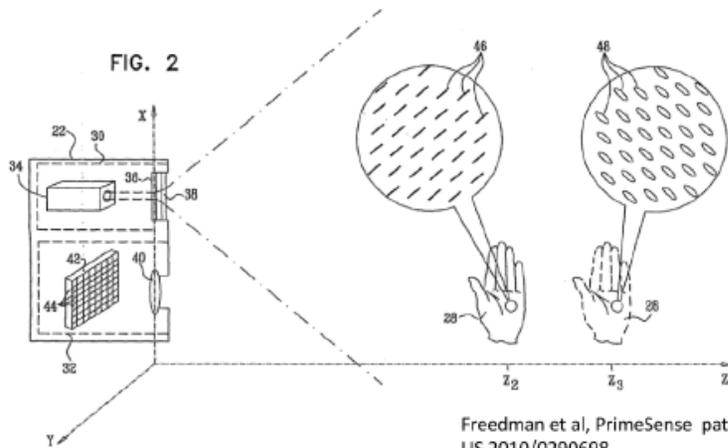
$$m(x, y) = (x, y) - (x', y') = \frac{bf}{Z}$$

Sensor Kinect



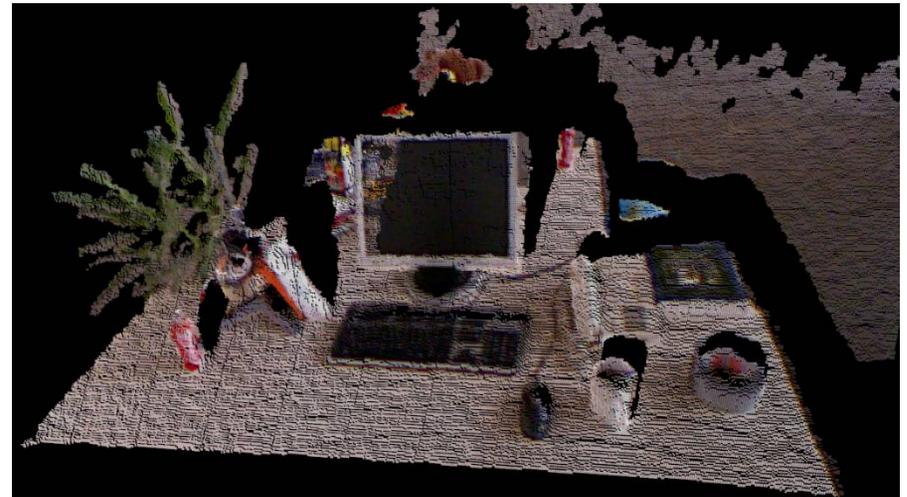
Cámara	Resolución (píxeles)	Tasa de transferencia (fps)
RGB	1280x1024	10
	640x480	30
Profundidad	640x480	30

Cámara	Resolución (píxeles)	Tasa de transferencia (fps)
RGB	1280x1024	10
	640x480	30
Profundidad	640x480	30

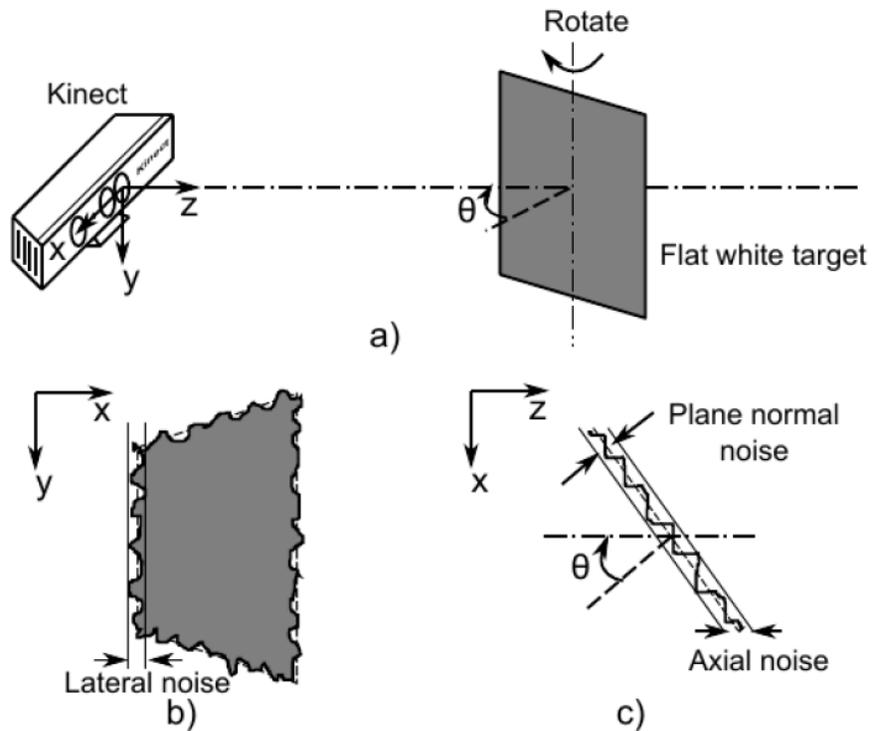


Freedman et al, PrimeSense patent application
US 2010/0290698





Modelo de ruido



$$\sigma_L(\theta)_{pixels} = 0.8 + 0.035 \frac{\theta}{\frac{\pi}{2} - \theta}$$

$$\sigma_L(\theta)_{metros} = \sigma_L(\theta)_{pixels} Z \frac{p_x}{f_x}$$

$$\sigma_Z(Z, \theta) = 0.0012 + 0.0019 (Z - 0.4)^2 + 0.0001 \frac{\theta^2}{\sqrt{Z} (\frac{\pi}{2} - \theta)^2}$$

Etapa de detección



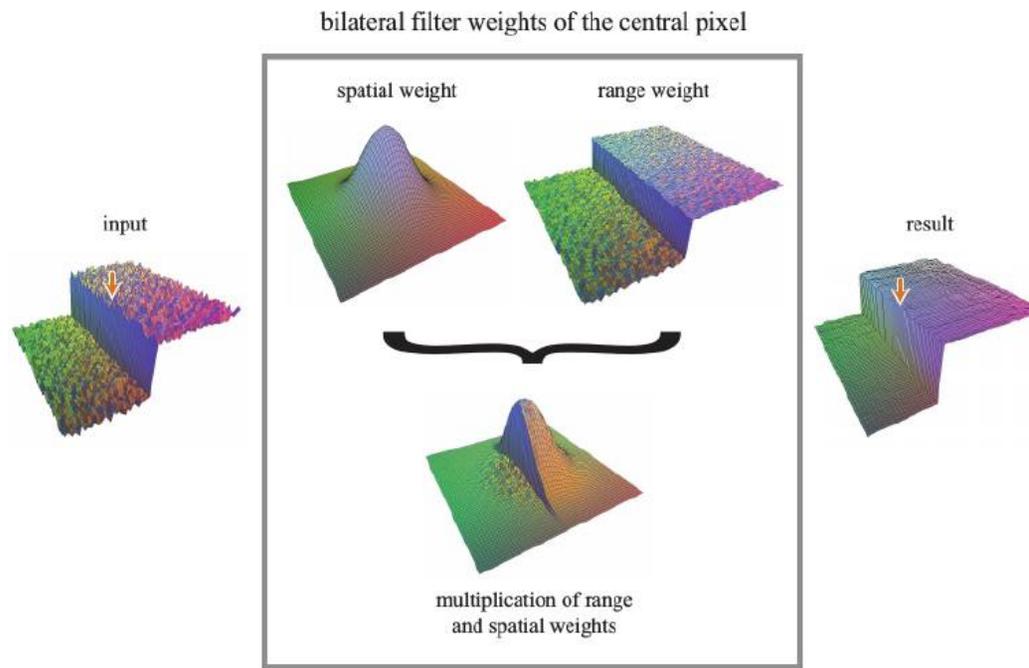
Filtrado Bilateral

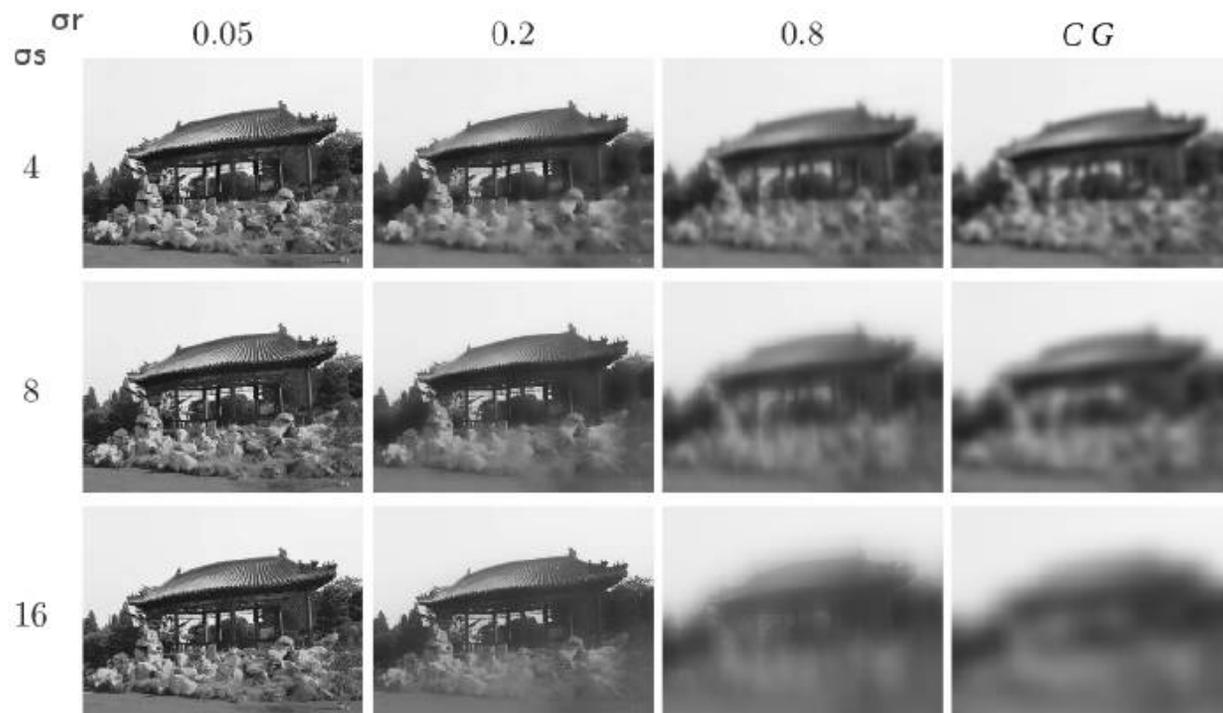
$$\hat{I}(u) = \frac{\sum_{p \in N(u)} W_s(\|p - u\|) W_r(|I(u) - I(p)|) I(p)}{\sum_{p \in N(u)} W_s(\|p - u\|) W_r(|I(u) - I(p)|)}$$

$$N(u) = \{p_i : \|u - p_i\| < \rho = 2\sigma_s\}$$

$$W_s(x) = e^{\frac{-x^2}{2\sigma_s^2}}$$

$$W_r(x) = e^{\frac{-x^2}{2\sigma_r^2}}$$





Filtrado Bilateral en nubes de puntos

$$\mathbf{n}_i = \frac{\sum_{p_j \in N(p_i)} W_s(\|p_j - p_i\|) W_r(d_{ij})}{\sum_{p_j \in N(p_i)} W_s(\|p_j - p_i\|) W_r(d_{ij})} \hat{\mathbf{n}}_j$$

$$d_{ij} = \hat{\mathbf{n}}_i \cdot (\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_j)$$

$$\Delta \mathbf{p}_i = \frac{\sum_{p_j \in N(p_i)} W_s(\|p_j - p_i\|) W_r(d_{ij})}{\sum_{p_j \in N(p_i)} W_s(\|p_j - p_i\|) W_r(d_{ij})} \mathbf{x}_j$$

$$\mathbf{x}_j = (((\mathbf{n}_i \cdot \mathbf{n}_j) \mathbf{n}_j) \cdot \mathbf{n}_i) \mathbf{n}_i$$

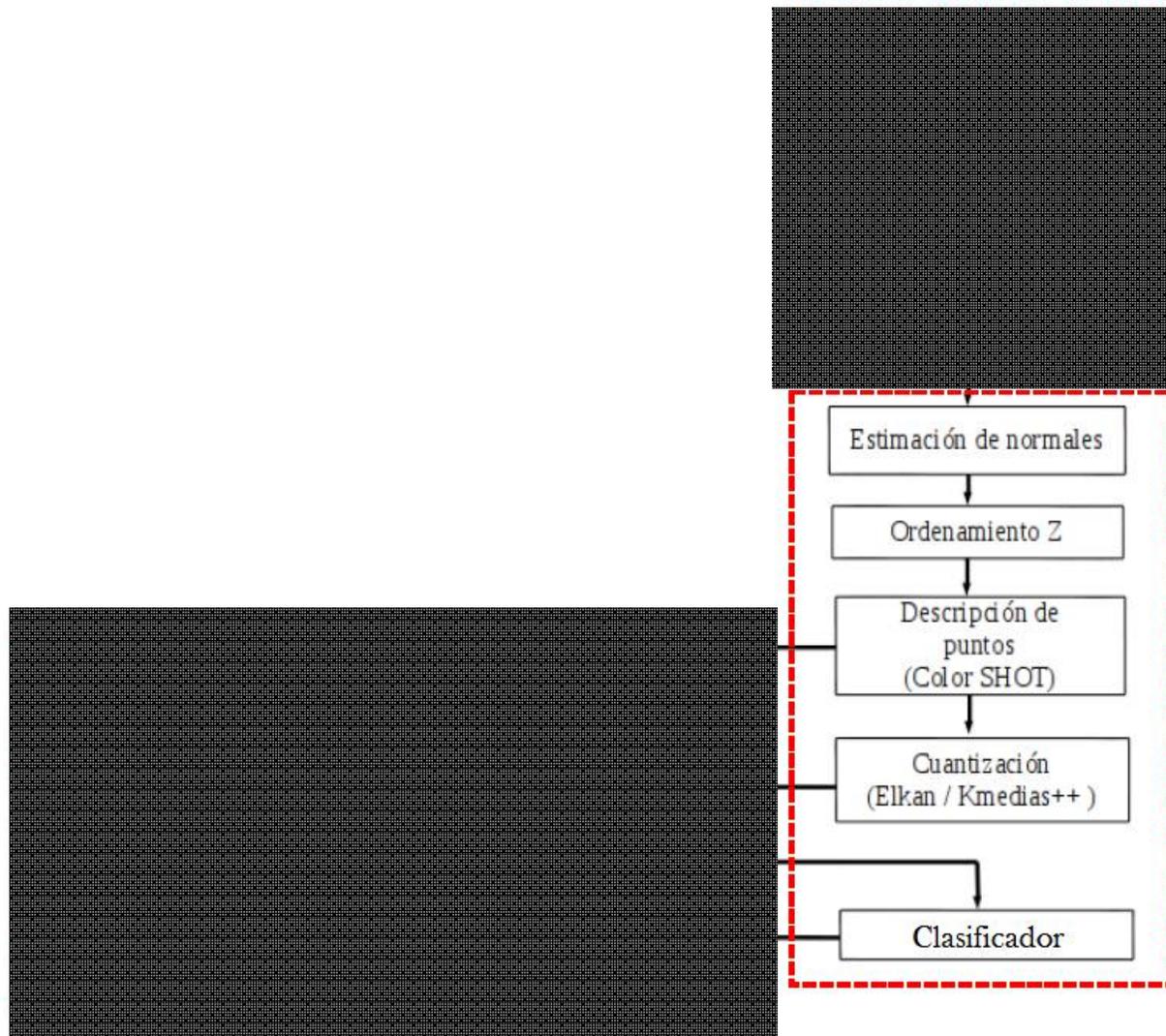
$$\mathbf{p}_i = \mathbf{p}_i + \Delta \mathbf{p}_i$$

Detector de puntos característicos ISS

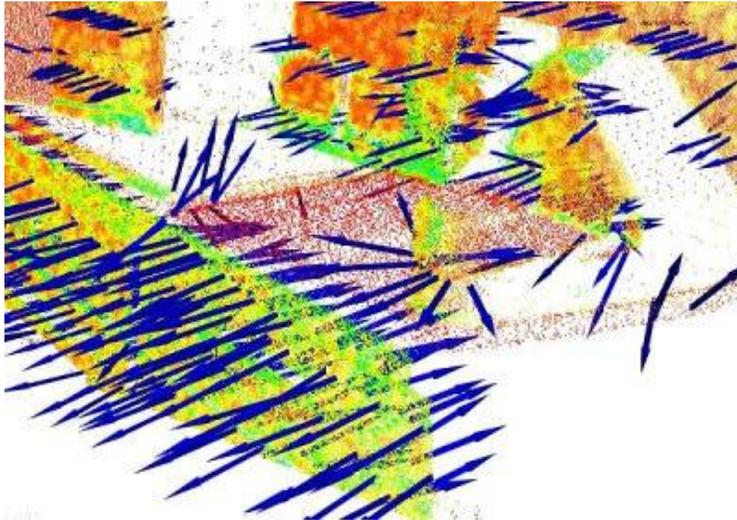
$$w_i = \frac{1}{|\{p_j : |p_j - p_i| < r_{density}\}|}$$
$$C(p_i) = \frac{\sum_{|p_j - p_i| < r_{frame}} w_j (p_j - p_i)(p_j - p_i)^T}{\sum_{|p_j - p_i| < r_{frame}} w_j}$$

- Para seleccionar los puntos candidatos para crear su vector de características de forma, se usan dos umbrales: el primer umbral γ_{12} , es para comparar la razón entre el primer y segundo valor propio; el segundo umbral γ_{23} , para comparar la razón entre el segundo y tercer valor propio.

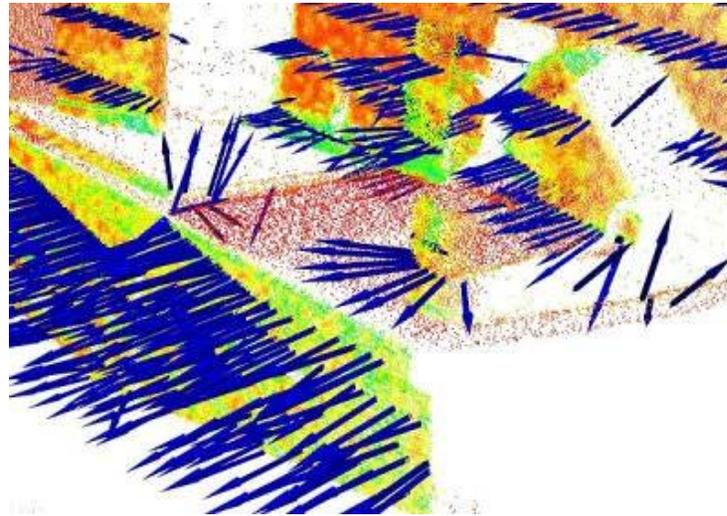
Etapa de descripción



Estimación de normales



$$C = \frac{1}{k} \sum_{i=1}^k \cdot (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, C \cdot v_j = \lambda_j \cdot v_j, j \in \{0, 1, 2\}$$

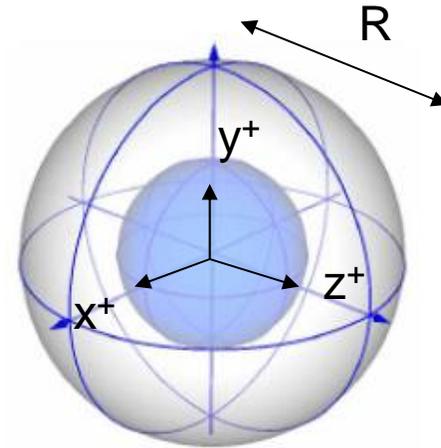


$$\vec{n}_i \cdot (v_p - p_i) > 0$$

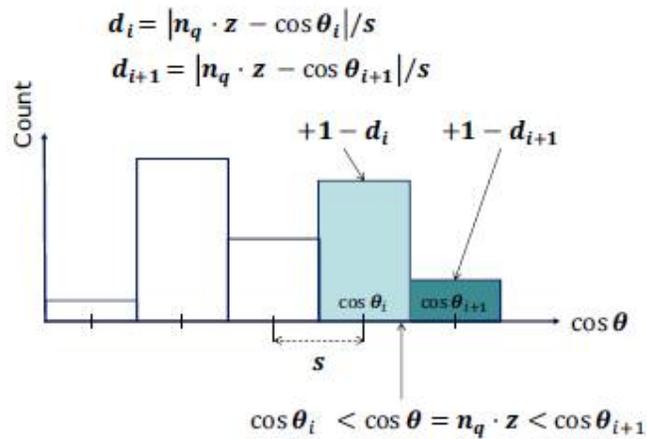
Descriptor de puntos característicos SHOT

$$M = \frac{1}{\sum_{i:d_i \leq R} (R - d_i)} \sum_{i:d_i \leq R} (R - d_i) (\mathbf{p}_i - \mathbf{p}) (\mathbf{p}_i - \mathbf{p})^T$$

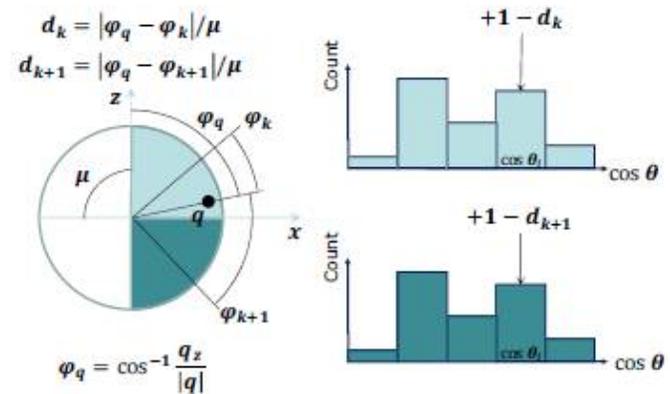
con $d_i = \|\mathbf{p}_i - \mathbf{p}\|$



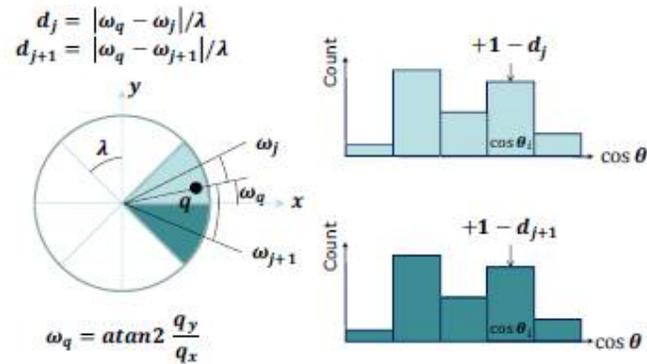
- Se calcula $\cos(\Theta_q) = \mathbf{z}_k \cdot \mathbf{n}_q$, el azimut, la elevación y la distancia radial.
- 8 regiones de azimuth, 2 regiones de elevación, 2 regiones radiales.
- 11 bins por cada región.
- $8 \times 2 \times 2 \times 11 =$ vector de 352 características.



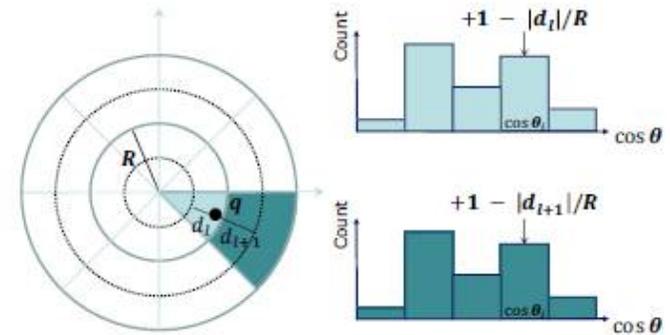
(a) Interpolation on normal cosines



(c) Interpolation on elevation

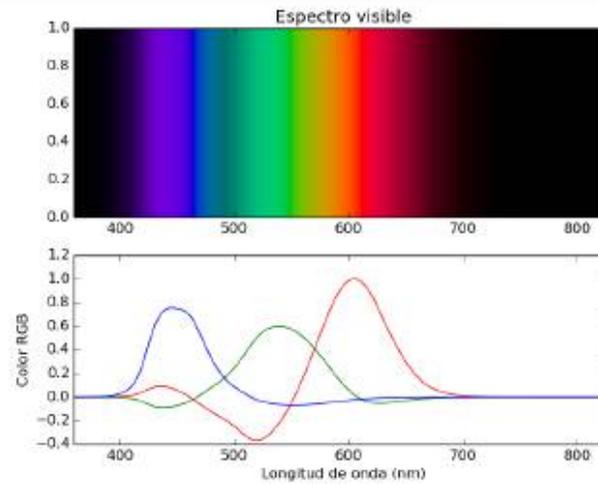


(b) Interpolation on azimuth

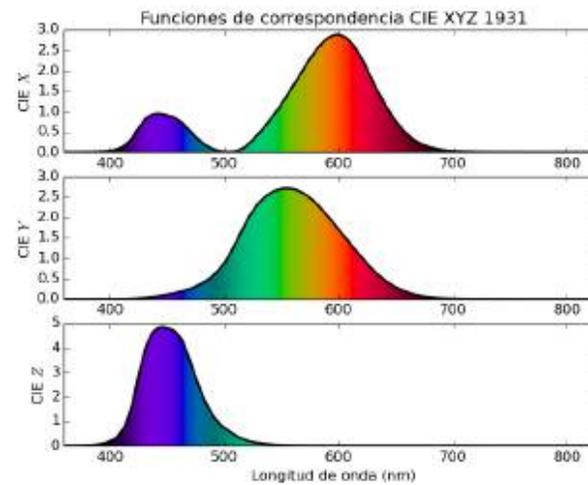


(d) Interpolation on distance

Espacio de color CIE XYZ

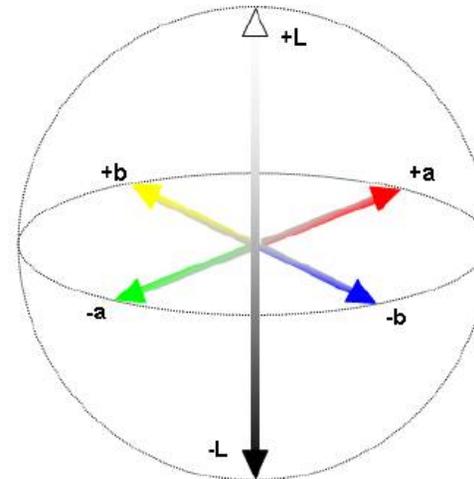
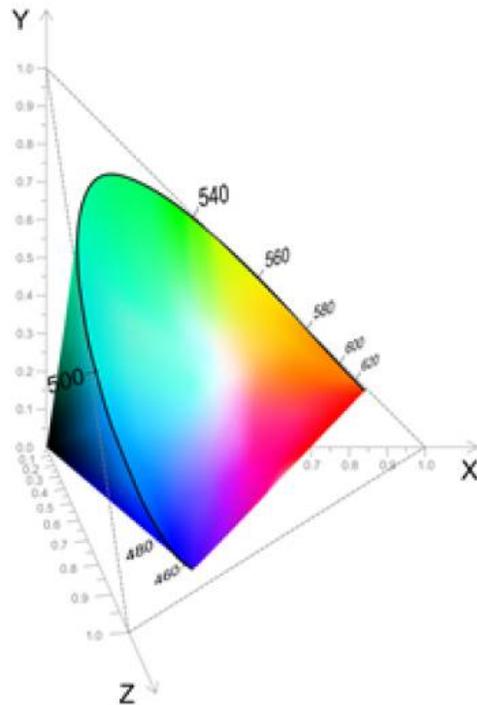


(a)



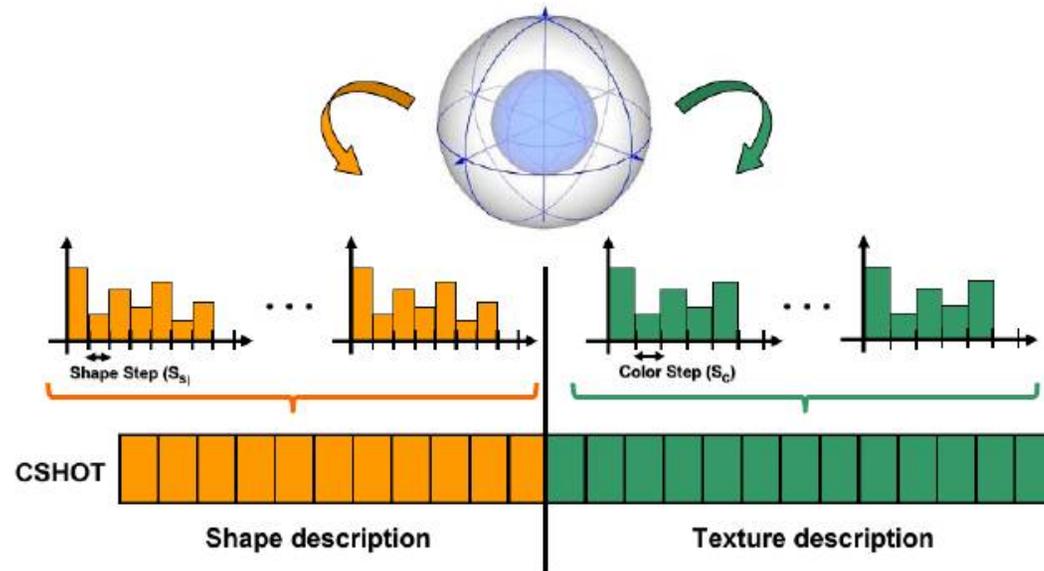
(b)

CIE Lab



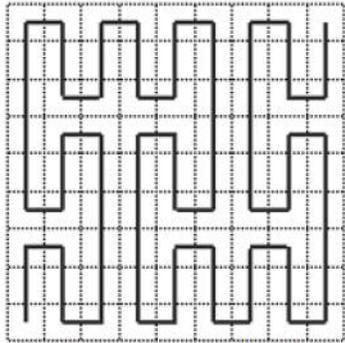
Color SHOT

$$\gamma = \| Lab_{\mathbf{q}} - Lab_{\mathbf{p}} \|_1$$

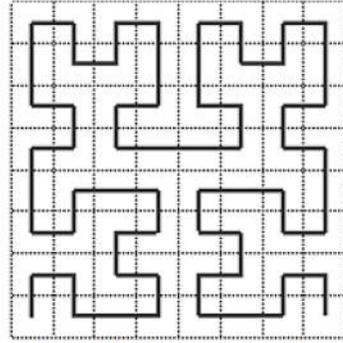


- Se calcula $\cos(\Theta_q) = \mathbf{z}_k \cdot \mathbf{n}_q$, el azimut, la elevación y la distancia radial.
- 8 regiones de azimuth, 2 regiones de elevación, 2 regiones radiales.
- 11 bins por cada región.
- Se agregan histogramas de color CIE Lab con 31 bins por región.
- $8 \times 2 \times 2 \times 11 + 8 \times 2 \times 2 \times 31 = \text{vector de } 352 + 992 = 1344 \text{ características.}$

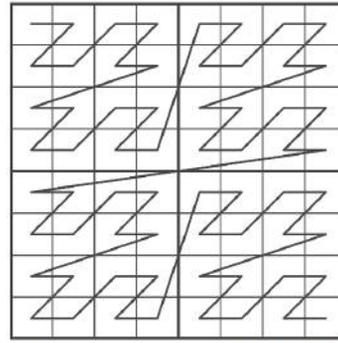
Curvas de llenado de espacio



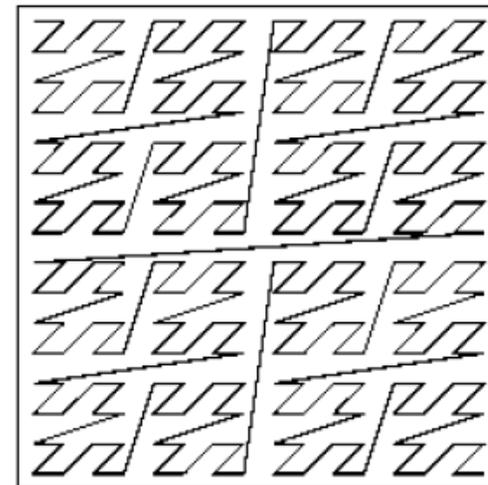
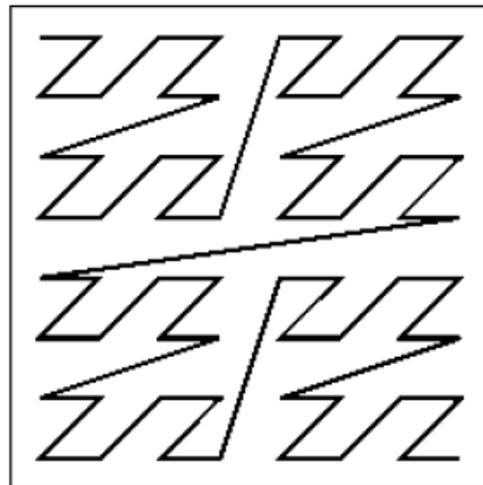
(a) Ejemplo de curva de Peano.



(b) Ejemplo de curva de Hilbert.

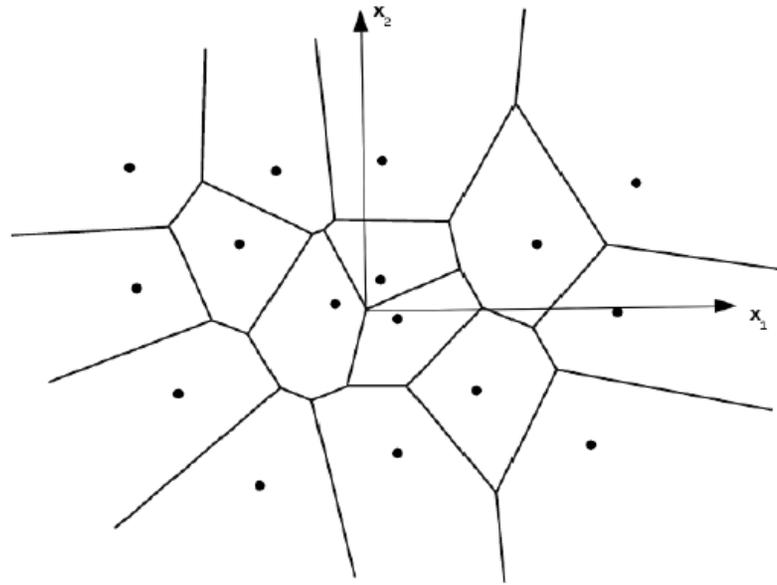


(c) Ejemplo de curva de Morton.



	x:	0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
y: 0	000	000000	000001	000100	000101	010000	010001	010100	010101
1	001	000010	000011	000110	000111	010010	010011	010110	010111
2	010	001000	001001	001100	001101	011000	011001	011100	011101
3	011	001010	001011	001110	001111	011010	011011	011110	011111
4	100	100000	100001	100100	100101	110000	110001	110100	110101
5	101	100010	100011	100110	100111	110010	110011	110110	110111
6	110	101000	101001	101100	101101	111000	111001	111100	111101
7	111	101010	101011	101110	101111	111010	111011	111110	111111

El método de la K medias



$$E(C) = \frac{1}{N} \sum_{i=1}^N \|x_i - c_i\|^2$$

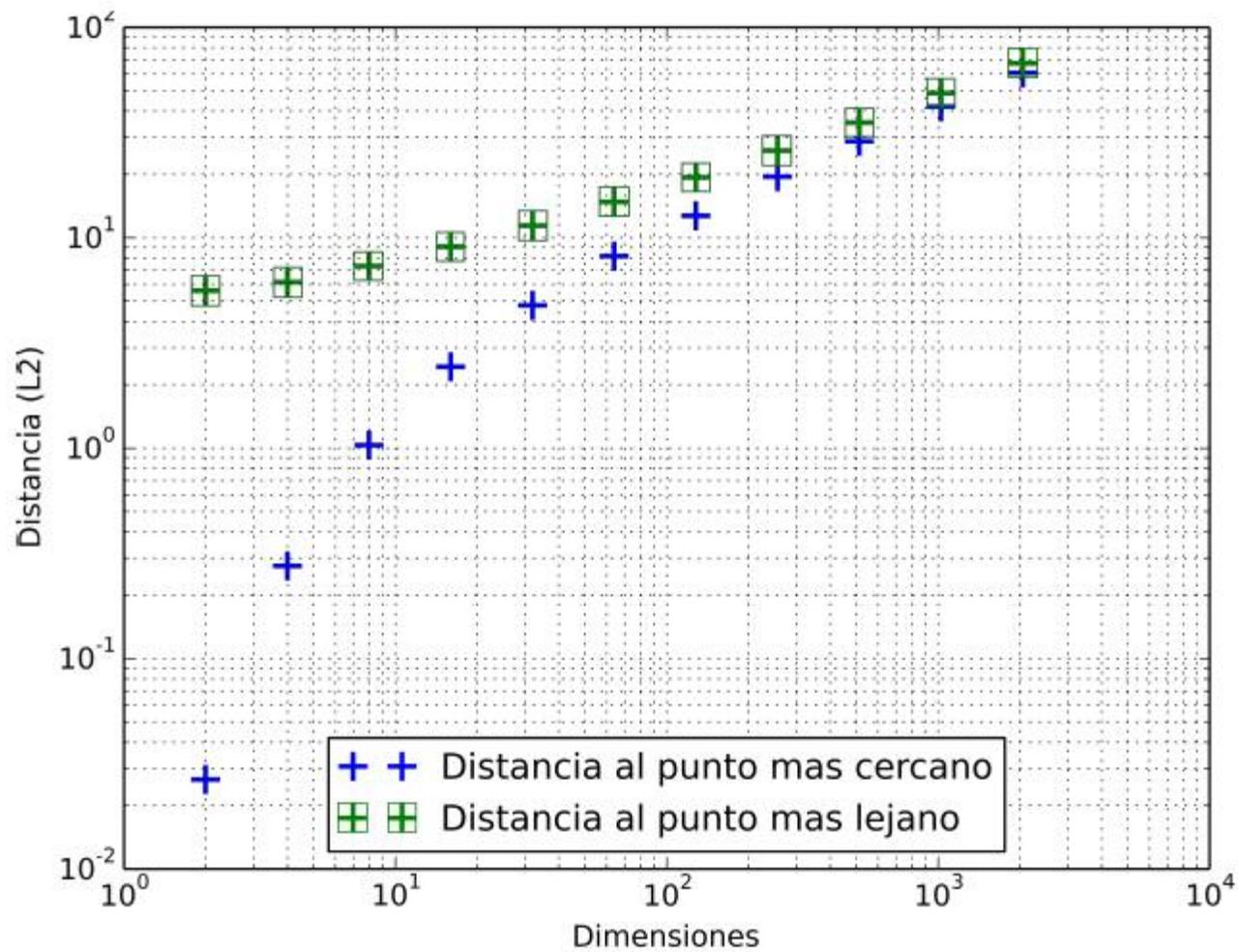
$$c_i = \text{Arg Min}_{j \in \{1, \dots, k\}} \|x_i - c_j\|^2$$

Problemas de dimensionalidad

$$d_{max} - d_{min}$$

$$L_p = \|x - y\|_p = \sqrt[p]{\sum_{i=1}^N |x_i - y_i|^p}, \text{ con } d \geq 1$$

- Para la norma L_1 la distancia relativa se incrementa según aumenta la dimension del espacio.
- Para norma L_2 la distancia relativa permanece aproximadamente constante y:
- Para la norma L_d para $d \geq 3$, esta tiende a cero conforme aumenta la dimensionalidad, lo que significa que los puntos en dichos espacios tienden a estar alejados una distancia igual a cero.



Solución

$$L_p = \|x - y\|_p = \sqrt[p]{\sum_{i=1}^N |x_i - y_i|^p} \quad , \text{con } p \geq 1$$



$$L_p = \|x - y\|_p = \sum_{i=1}^N |x_i - y_i|^p \quad 0 < p < 1 \quad 0 < p = \frac{a}{b} < 1$$

$$L_p = \sum_{i=1}^N \sqrt[b]{|x_i - y_i|^a} \quad a < b$$

Mejoras al desempeño

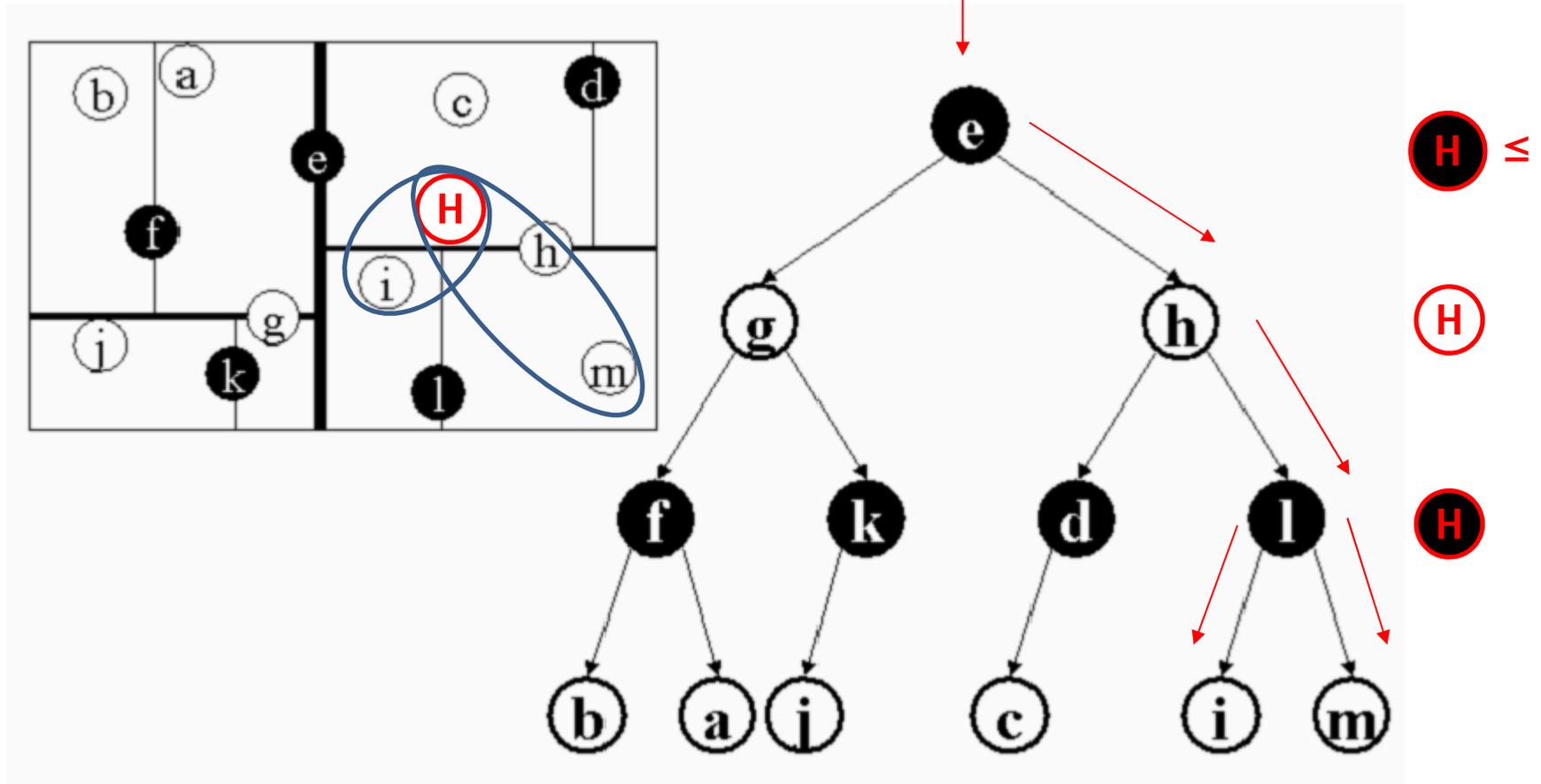
- K medias++

$$P(x_i) = \frac{D(x_i)^2}{\sum_{x \in X} D(x)^2} \quad D(x_i) = \text{Arg Min}_{c_j \in C} \|x_i - c_j\|_p$$

- Algoritmo de Elkan

$$\|x_i - c\|_p - \|c - c'\|_p \leq \|x_i - c'\|_p \leq \|x_i - c'\|_p + \|c + c'\|_p$$

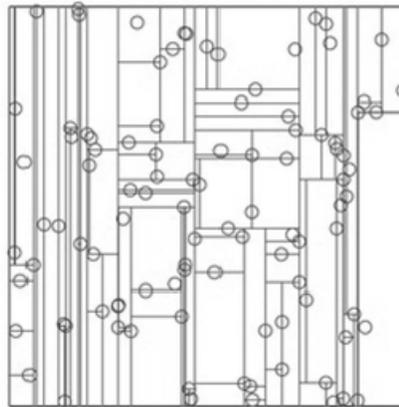
- Árboles KD



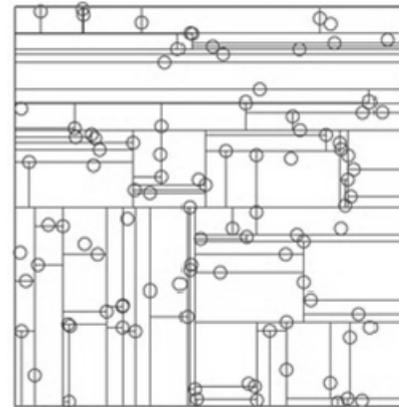
Solución desempeño + dimensionalidad

- Árboles KD aleatorios

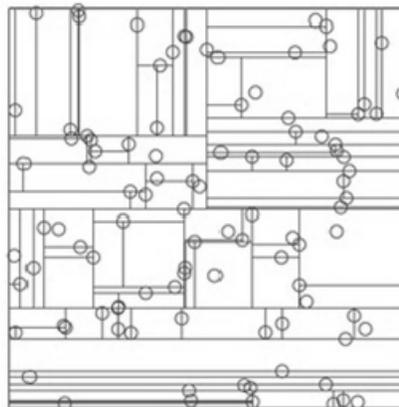
árbol KD 1



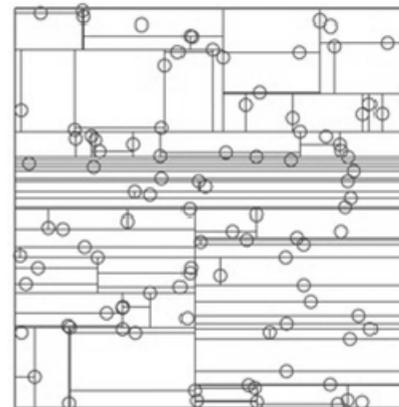
árbol KD 2



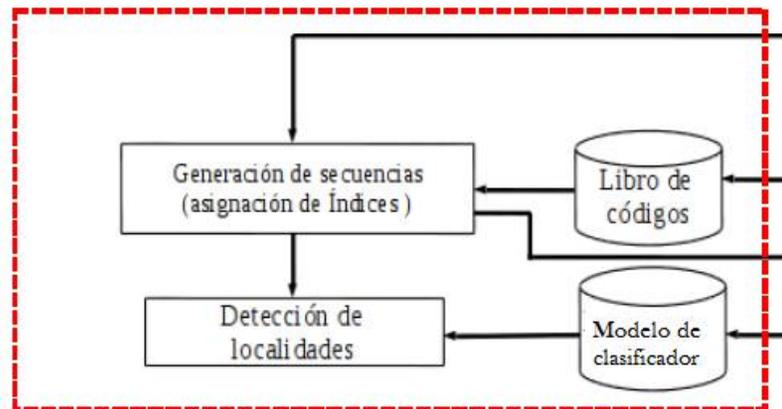
árbol KD 3



árbol KD 4



Etapa de clasificación



Métodos de clasificación

● Modelos ocultos de Markov

$$P[X(t+1) = j \mid X(0) = k_0, X(1) = k_1, \dots, X(t) = i] = P[X(t+1) = j \mid X(t) = i] = P_{ij}$$

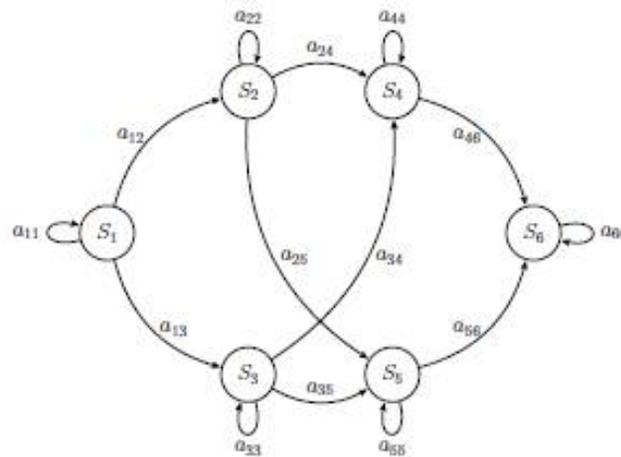
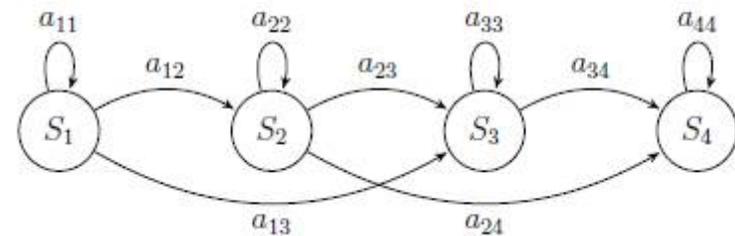
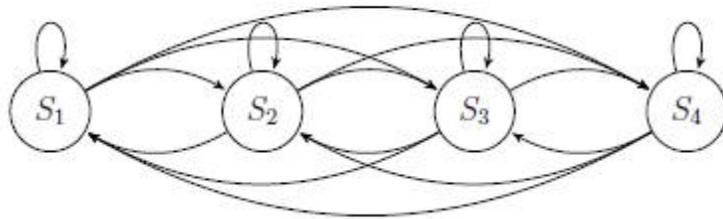
- $V = \{v_1, v_2, \dots, v_M\}$ es un alfabeto finito con M símbolos.
- $S = \{S_1, S_2, \dots, S_N\}$ es un conjunto finito de N estados, con el estado al tiempo t denotado como q_t .
- $A = [a_{ij}]_{N \times N}$ es una matriz de probabilidades de transición donde el elemento a_{ij} es la probabilidad de transición del estado i al estado j .

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i) \text{ para todo } 1 \leq i, j \leq N.$$

- $B = [b_j(k)]_{N \times M}$ es la matriz de probabilidad de emisiones, esto es que $b_j(k) = P(V_k \text{ en } t \mid q_t = S_j)$.
- $\Pi = [\pi_i]_{1 \times N}$ es la matriz de probabilidades iniciales donde $\pi_i = P(q_1 = S_i)$ con $1 \leq i \leq N$.

$$\lambda = (V, S, A, B, \Pi)$$

Modelos ocultos de Markov



Los tres problemas fundamentales

1. Calcular la probabilidad de una secuencia observada.

Dada una secuencia de observaciones $O = \{O_1, O_2, \dots, O_T\}$ y un modelo $\lambda = (A, B, \Pi)$, calcular $P(O | \lambda)$.

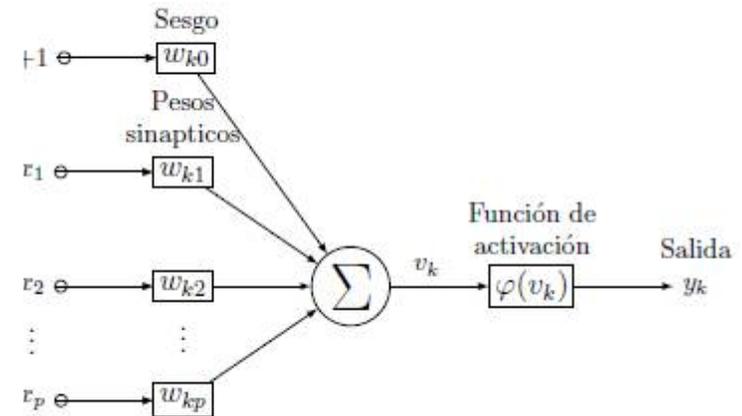
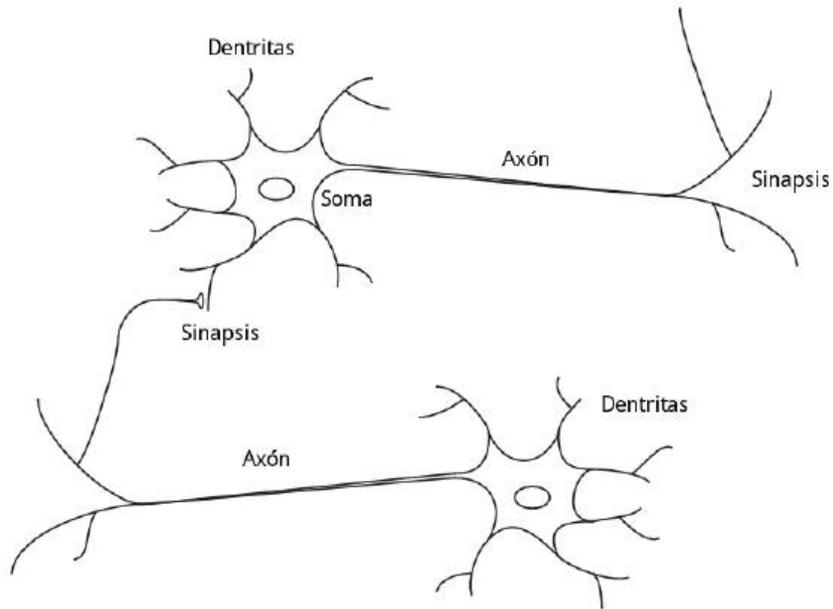
2. Calcular la secuencia óptima de estados que genera las observaciones.

Dada una secuencia de observaciones $O = \{O_1, O_2, \dots, O_T\}$ y un modelo $\lambda = (A, B, \Pi)$, encontrar $Q = \{q_1, q_2, \dots, q_T\}$ óptima para generar O .

3. El aprendizaje del modelo.

Ajustar los parámetros A , B y Π para maximizar $P(O | \lambda)$.

Redes neuronales artificiales

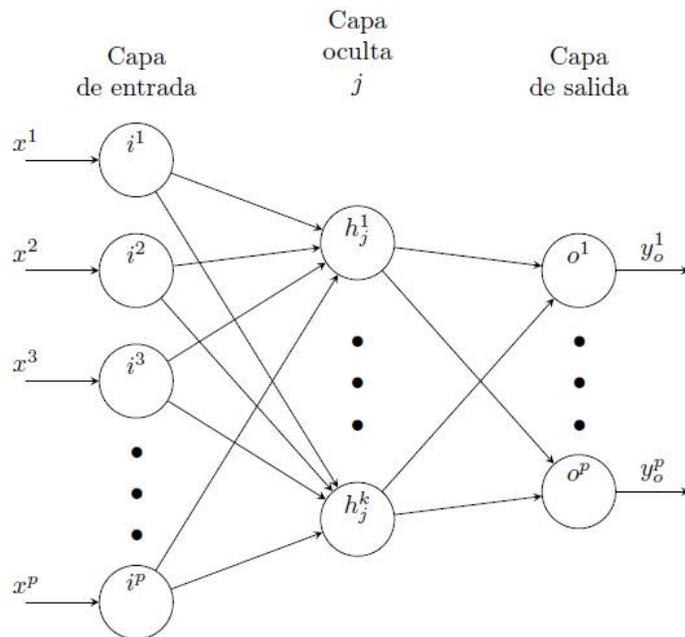


$$v_k = \sum_{j=0}^p w_{kj} x_j$$

$$v_k = \begin{bmatrix} w_{k0} & w_{k1} & \dots & w_{kp} \end{bmatrix} \begin{bmatrix} x_0 = +1 \\ x_1 \\ \vdots \\ x_p \end{bmatrix} = \mathbf{w}_k^T \mathbf{x}$$

$$y_k = \varphi(v_k)$$

Redes neuronales de propagación hacia adelante

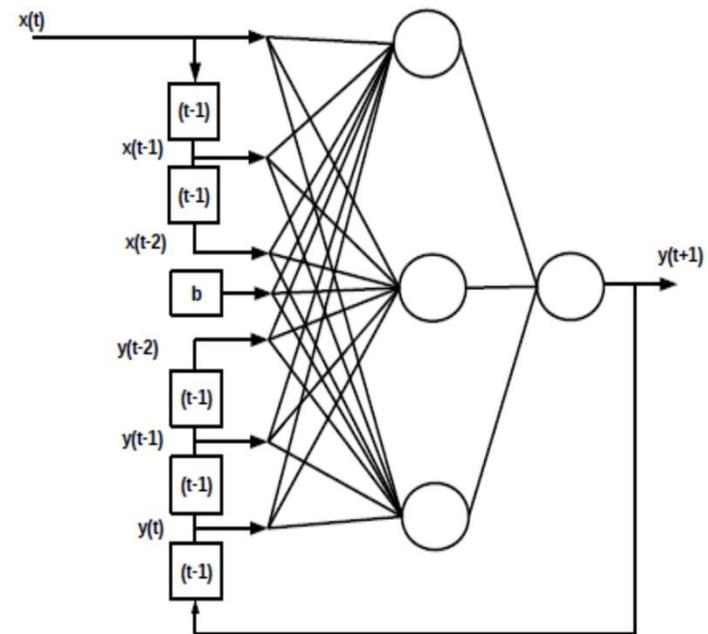
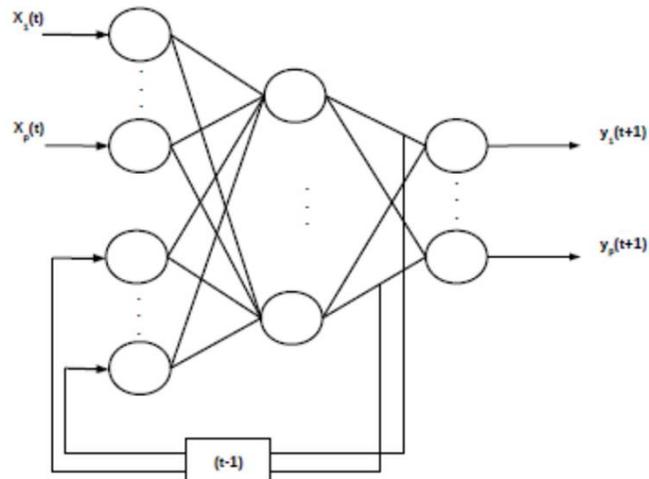
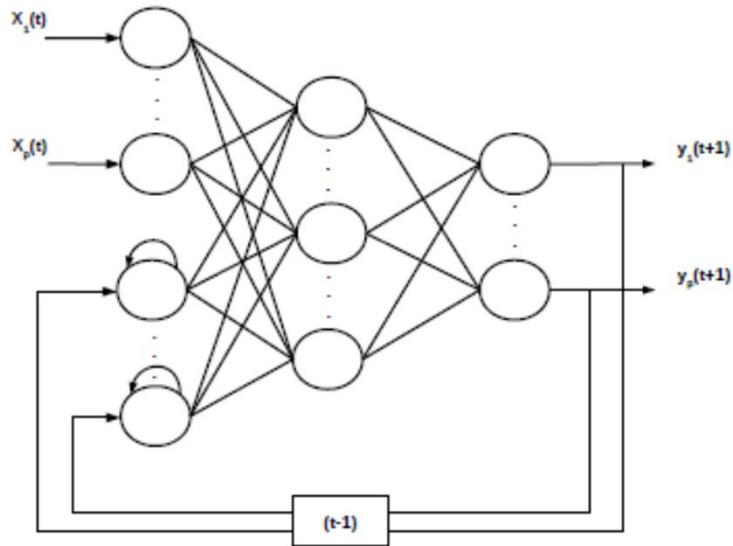


$$\frac{\partial E^p}{\partial y_h^p} = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} \frac{\partial s_o^p}{\partial y_h^p} = \frac{\partial E^p}{\partial s_o^p} \frac{\partial}{\partial y_h^p} \sum_{j=1}^{N_h} w_{ko} y_j^p = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} w_{ho} = - \sum_{o=1}^{N_o} \delta_o^p w_{ho}$$

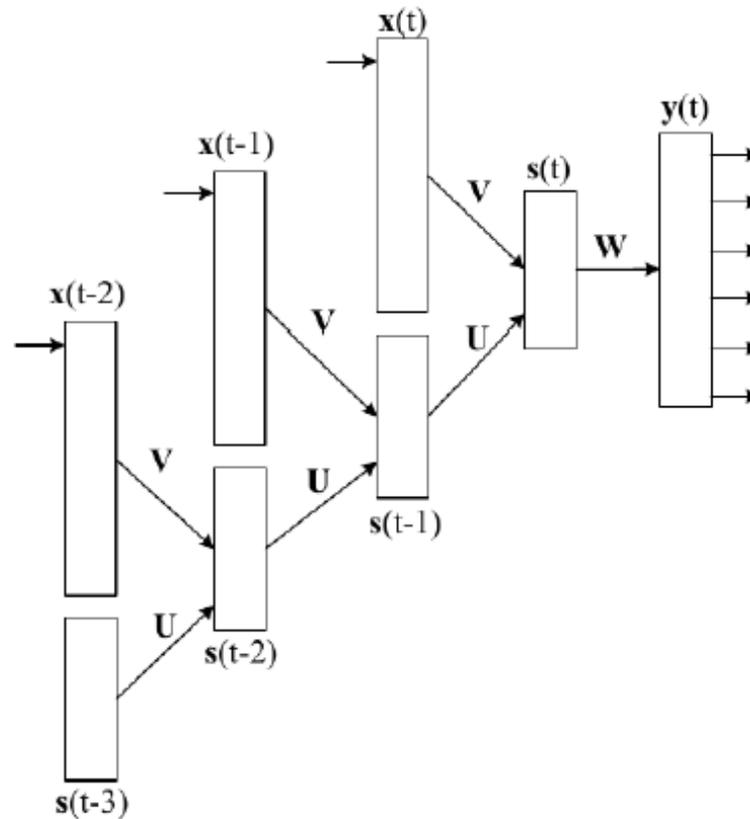
$$\delta_h^p = F'(s_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho}$$

- Son estimadores universales. Pueden aproximar cualquier función (Teorema de Kolmogórov).
- Además pueden aproximar cualquier función y su derivada (Teorema de Cybenko) usando funciones derivables y acotadas.

Redes recurrentes

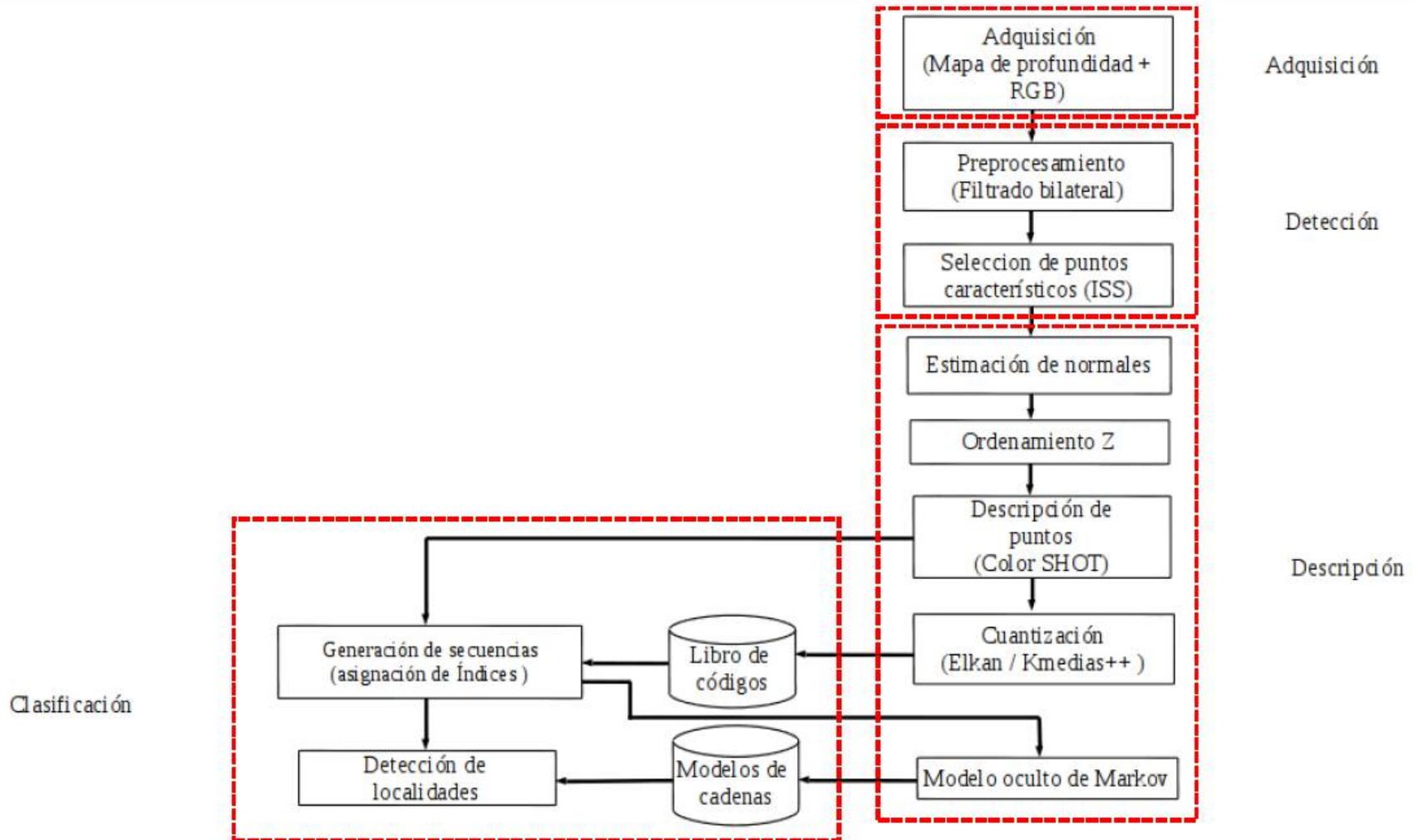


Retropropagación del error en el tiempo



$$\delta_{pj}(t-1) = \sum_h^m \delta_{ph}(t) u_{hj} F'(y_{pj}(t-1))$$

Implementación del sistema

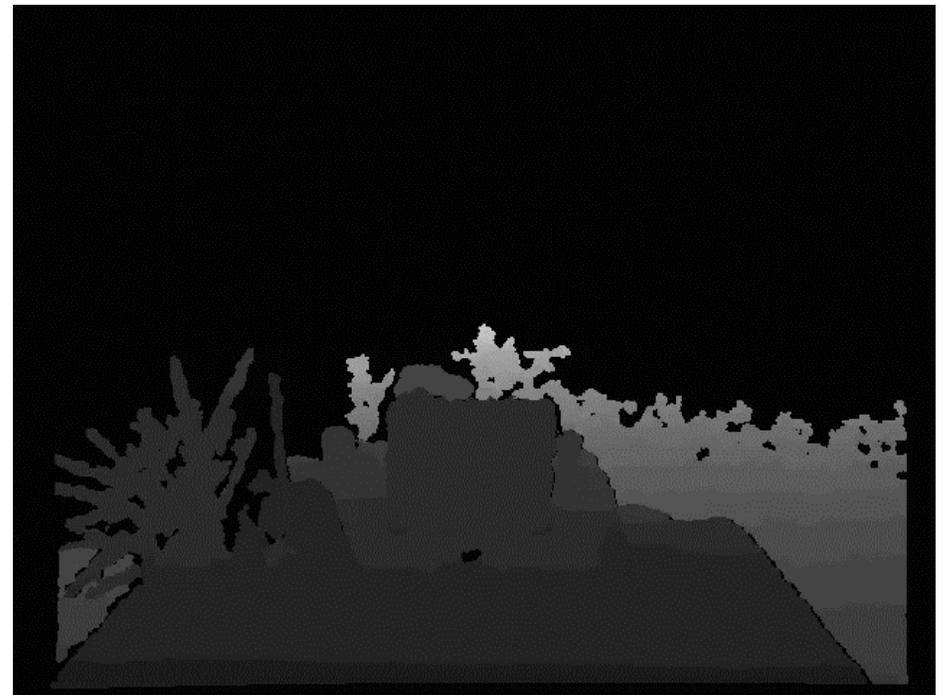


Pruebas

- Conjunto de datos

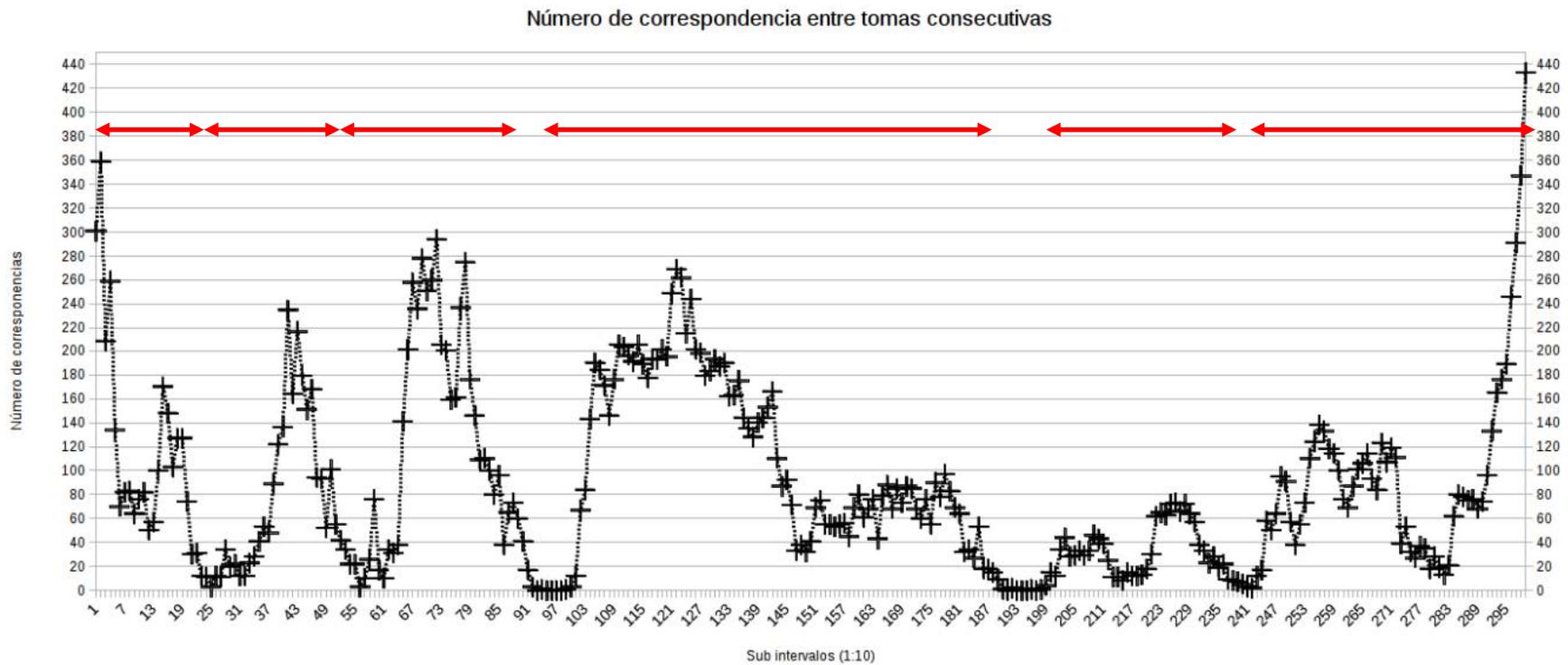


Camera	fx	fy	cx	cy	d0	d1	d2	d3	d4
(ROS default)	525.0	525.0	319.5	239.5	0.0	0.0	0.0	0.0	0.0
 Freiburg 2 RGB	520.9	521.0	325.1	249.7	0.2312	-0.7849	-0.0033	-0.0001	0.9172



Camera	fx	fy	cx	cy	d0	d1	d2	d3	d4
 Freiburg 2 IR	580.8	581.8	308.8	253.0	-0.2297	1.4766	0.0005	-0.0075	-3.4194

- Selección de las clases (localidades)



6 tipos de clases (3 ejemplos por clase)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)



(j)



(k)



(l)



(m)



(n)



(ñ)



(o)

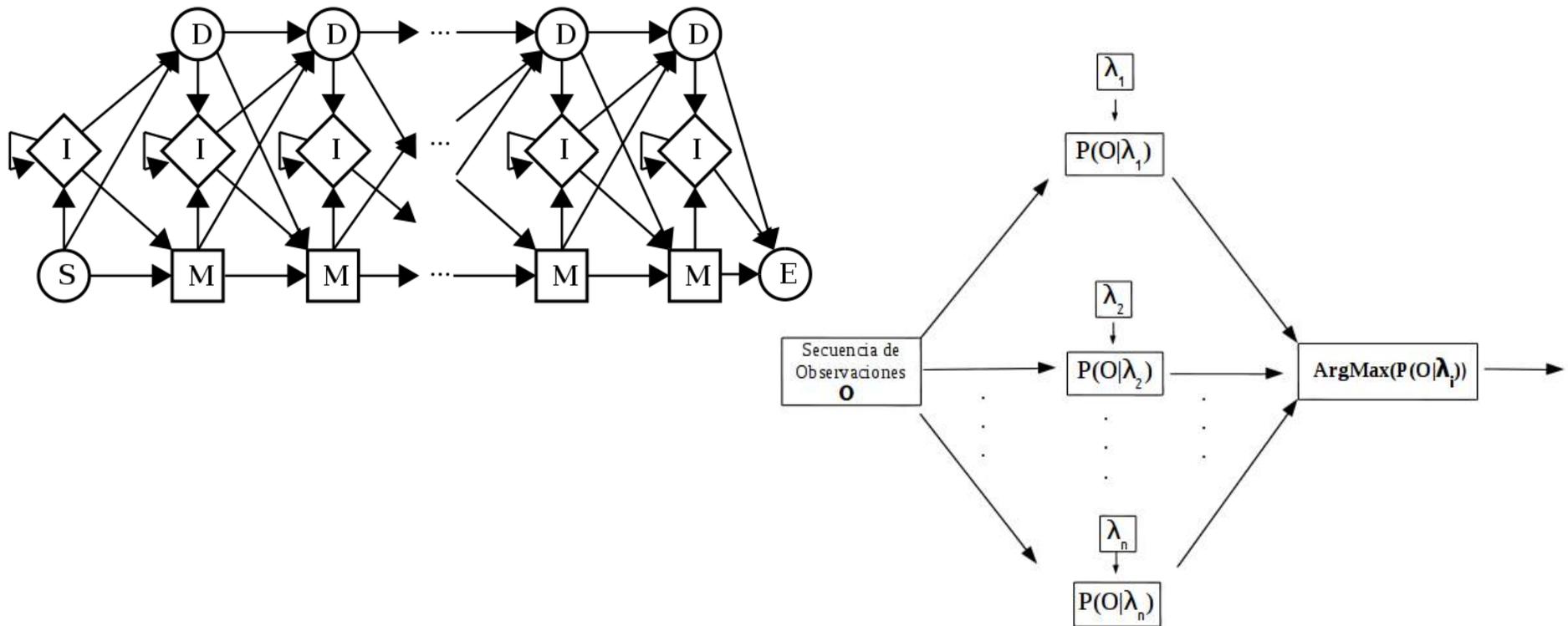


(p)



(q)

Clasificador HMM



- Se elimina aleatoriamente el 10% de símbolos en todas las secuencias.
- Se usa un perfil de modelo oculto con 15 bloques M-I-D.
- Se utiliza validación aleatoria con una razón 50% entrenamiento, 50% pruebas.
- Se hacen 10 validaciones.

Resultados

		Valor real						R	NR	% R	% NR
		L_1	L_2	L_3	L_4	L_5	L_6				
Valor actual	L_1	100	20	0	0	0	0	100	20	83.3	16.7
	L_2	0	140	10	0	0	0	140	10	93.3	6.7
	L_3	10	40	90	10	0	20	90	80	52.9	47.1
	L_4	30	80	60	250	20	10	250	200	55.6	44.4
	L_5	0	0	40	0	170	0	170	40	81	19
	L_6	0	110	60	10	30	70	70	210	25	75
		Total						$\frac{820}{1380}$	$\frac{560}{1380}$	59.4	40.6

Tabla de confusión para 16 símbolos. L_1 =Localidad 1. L_2 =Localidad 2. L_3 =Localidad 3. L_4 =Localidad 4. L_5 =Localidad 5. L_6 =Localidad 6. R=Secuencias reconocidas. NR=No reconocidas.

		Valor real						R	NR	% R	% NR
		L_1	L_2	L_3	L_4	L_5	L_6				
Valor actual	L_1	100	20	0	0	0	0	100	20	83.3	16.7
	L_2	0	140	0	10	0	0	140	10	93.3	0.7
	L_3	0	80	80	0	10	0	80	90	47	53
	L_4	40	70	0	330	10	0	330	120	73.3	26.7
	L_5	0	0	30	0	160	20	160	50	76.2	23.8
	L_6	10	100	0	0	30	140	140	140	50	50
	Total							$\frac{950}{1380}$	$\frac{430}{1380}$	68.8	31.2

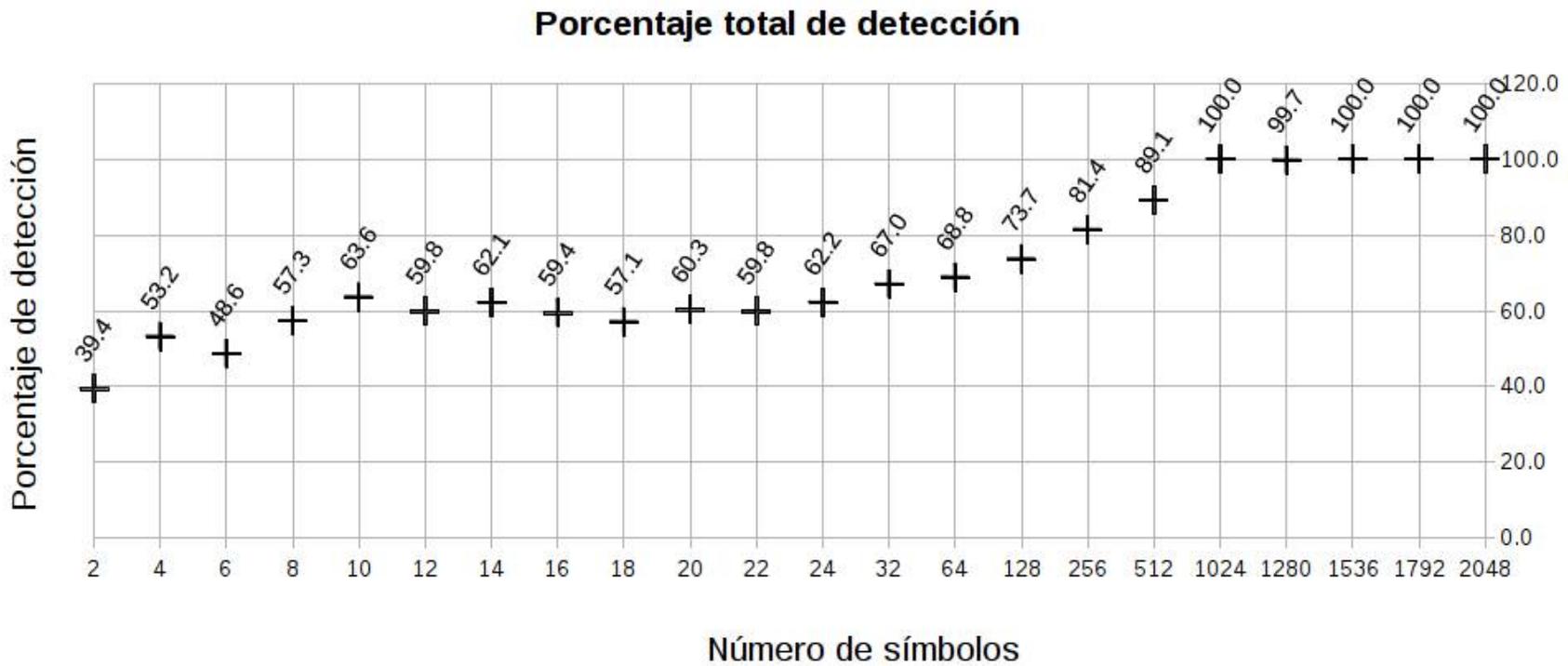
Tabla de confusión para 64 símbolos. L_1 =Localidad 1. L_2 =Localidad 2. L_3 =Localidad 3. L_4 =Localidad 4. L_5 =Localidad 5. L_6 =Localidad 6. R=Secuencias reconocidas. NR=No reconocidas.

		Valor real						R	NR	% R	% NR
		L_1	L_2	L_3	L_4	L_5	L_6				
Valor actual	L_1	110	10	0	0	0	0	110	10	91.7	8.3
	L_2	0	140	0	10	0	0	140	10	93.3	6.7
	L_3	0	20	150	0	0	0	150	20	88.2	11.8
	L_4	0	0	10	440	0	0	440	10	97.8	2.2
	L_5	0	0	0	0	210	0	210	0	100	0
	L_6	0	70	0	10	20	180	180	100	64.3	35.7
	Total							<u>1230</u>	<u>150</u>	89.1	10.9
							1380	1380			

Tabla de confusión para 512 símbolos. L_1 =Localidad 1. L_2 =Localidad 2. L_3 =Localidad 3. L_4 =Localidad 4. L_5 =Localidad 5. L_6 =Localidad 6. R=Secuencias reconocidas. NR=No reconocidas.

		Valor real						R	NR	% R	% NR
		L_1	L_2	L_3	L_4	L_5	L_6				
Valor actual	L_1	120	0	0	0	0	0	120	0	100	0
	L_2	0	150	0	0	0	0	150	0	100	0
	L_3	0	0	170	0	0	0	170	0	100	0
	L_4	0	0	0	450	0	0	450	0	100	0
	L_5	0	0	0	0	210	0	210	0	100	0
	L_6	0	0	0	0	0	280	280	0	100	0
	Total							1380	0	1380	100

Tabla de confusión para 1024 símbolos. L_1 =Localidad 1. L_2 =Localidad 2
 L_3 =Localidad 3. L_4 =Localidad 4. L_5 =Localidad 5. L_6 =Localidad 6. R=Secuencias reconocidas
 NR=No reconocidas.



Red neuronal de propagación hacia adelante

		Valor real						R	NR	% R	%NR
		L_1	L_2	L_3	L_4	L_5	L_6				
Valor actual	L_1	120	0	0	0	0	0	120	0	100	0
	L_2	15	135	0	0	0	0	135	15	90	10
	L_3	0	0	165	5	0	0	165	5	97	3
	L_4	5	0	5	430	5	5	430	20	96	4
	L_5	0	0	0	5	200	5	200	10	95.2	4.8
	L_6	5	0	0	0	5	270	270	10	96.4	3.6
	Total							1320	60	95.7	4.3
							1380	1380			

Tabla de confusión para la red neuronal de propagación hacia adelante. L_1 =Localidad 1. L_2 =Localidad 2. L_3 =Localidad 3. L_4 =Localidad 4. L_5 =Localidad 5. L_6 =Localidad 6. R=Secuencias reconocidas. NR=No reconocidas.

Etapa del sistema	Tiempo de procesamiento (s)
Eliminación del fondo	3.70×10^{-3}
Filtrado bilateral	4.94
Estimación de normales	609×10^{-3}
Detector ISS	1.40
Orden en Z	5.23×10^{-6}
Descriptor Color SHOT	87.6×10^{-3}

Número de símbolos	Tiempo de cuantización (ms)
2	4913
4	9049
6	14733
8	14548
10	18556
12	23869
14	24227
16	28307
18	36505
20	48819
22	30137
24	63450
32	40101
64	82690
128	124773
256	203260
512	366798
1024	656792
1280	738534
1536	896729
1792	1094362
2048	1230155

Comparación

Porcentaje total de detección		
	No. de símbolos	%
Perfil de modelo oculto	16	59.4
	64	68.8
	512	89.1
	1024	100
Red neuronal de propagación hacia adelante	---	95.7

Comparación de tiempos promedio		
	Entrenamiento (s)	Detección (ms)
Perfil de modelo oculto (todas las localidades, 1024 símbolos)	42.77 + 657	109.72
Red neuronal de propagación hacia adelante	1477	578
Red tipo NARX	115226	2294

Conclusiones

- La principal aportación de esta investigación es que usando los perfiles de modelos de ocultos de Markov (PHMM) es posible obtener resultados ligeramente mejores en la clasificación que una red neuronal con una sola capa oculta, ya que los modelos ocultos toman en cuenta la relación espacial y temporal de las observaciones.
- Una primera aproximación para el manejo de la oclusión usando secuencias de símbolos mediante los perfiles de modelos ocultos.
- Los mejores tiempos de entrenamiento y detección igualmente se logran usando modelos ocultos de Markov con 699.77 y 107.72×10^{-3} segundos respectivamente.
- La desventaja de este enfoque es la selección empírica para los parámetros del modelo.

Trabajo futuro

- Paralelización de la implementación de la biblioteca de modelos ocultos de Markov, además de los detectores y descriptores usados (PCL 1.8 incluye fallos en su implementación en paralelo). La etapa de filtrado bilateral y detección de puntos característicos son las más demandantes en tiempo de cómputo.
- Adecuación del sistema para aplicaciones computacionalmente más ligeras (uso de descriptores menos robustos pero más veloces).
- Implantación del sistema completo en un robot de servicio añadiendo la etapa de navegación.
- Experimentación con diferentes arquitecturas y modelos, como son modelos no homogéneos en el tiempo o mixtos.

Preguntas