

# Sistemas Inteligentes

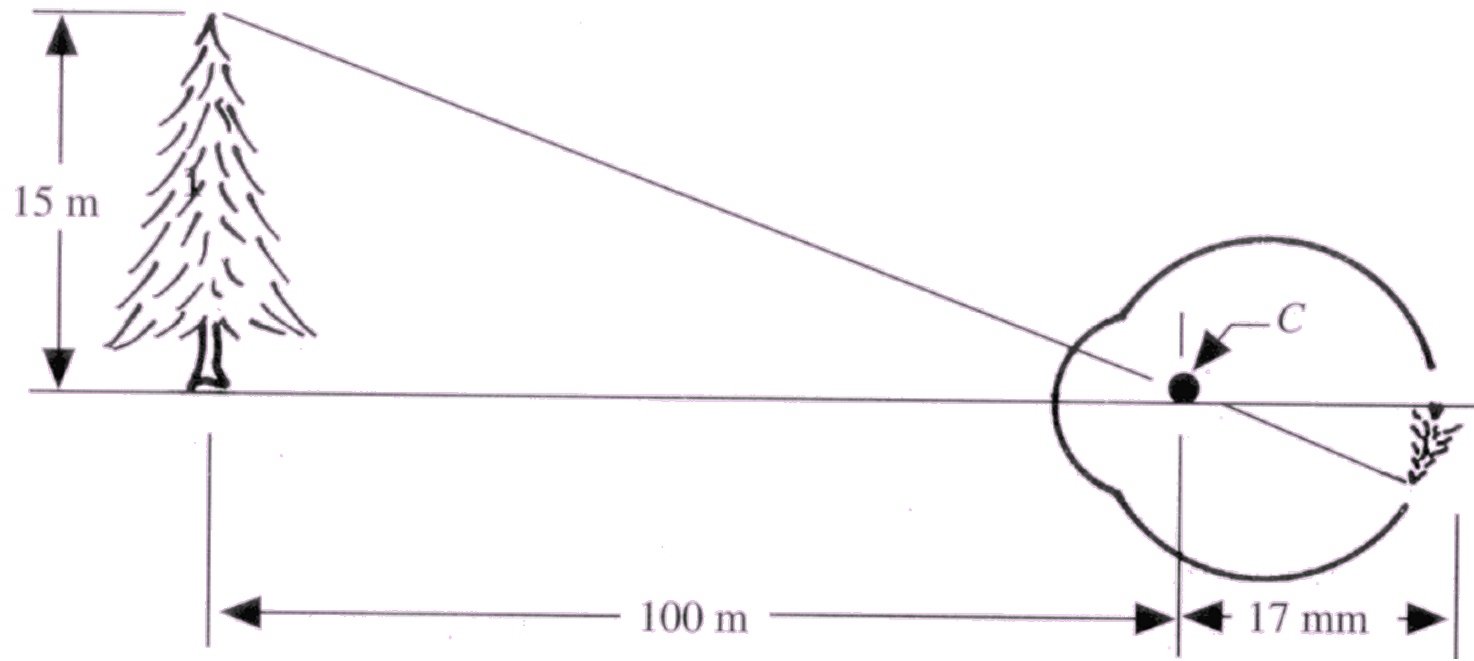
## Tema 8: Visión Artificial.

Curso 2024-25

# Índice

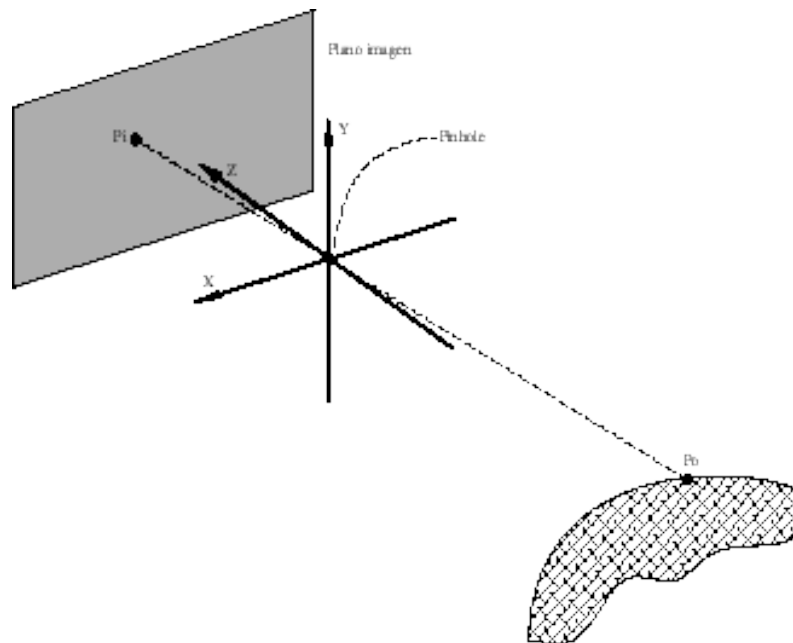
- Introducción a la visión
- Definiciones
- Convolución
- Filtrado
- Histograma de la imagen
- Aristas: algoritmo de Canny
- Detector de Nitzberg-Harris

# Símil biológico: cómo vemos



# Captura de imagen

- Sistemas CCD o CMOS (y otros)
- Sistema de *pin-hole*: se pierde la “perspectiva”
- Se produce una discretización



# ¿Qué es una imagen para el ordenador?

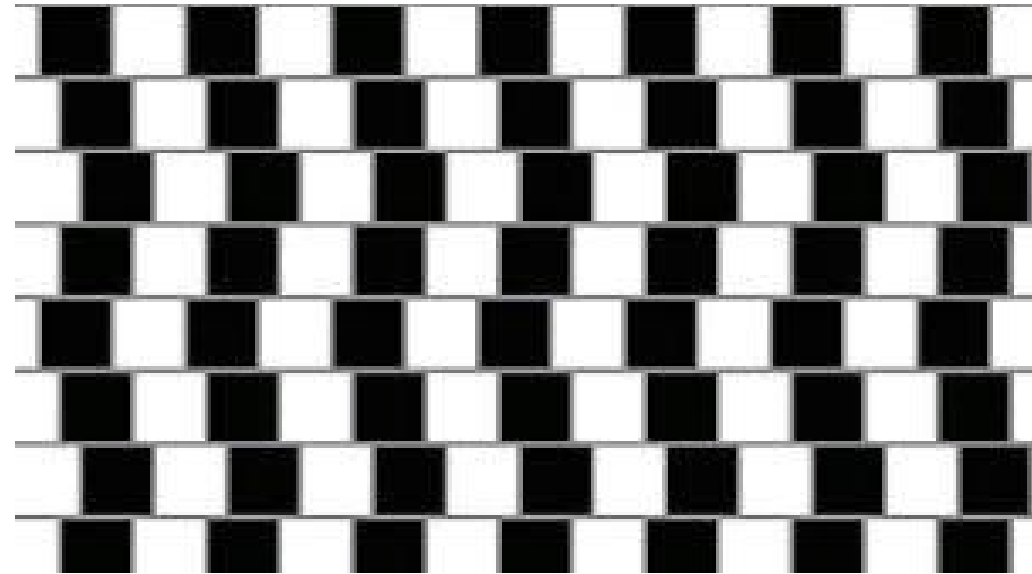
201	196	189	184	184	189	196	201	210	198	231	175	108	095	069	086	082	111	161	167	072	140	168	165	150	161	157	163	145	153	146	153
209	205	199	194	191	192	195	197	202	214	240	143	057	067	050	039	058	074	128	162	100	159	167	155	157	162	158	170	166	184	189	201
216	214	211	207	203	199	197	195	193	216	231	114	022	055	059	031	046	063	116	146	091	133	146	156	160	158	155	173	184	208	220	234
215	217	218	218	215	210	204	200	196	201	213	122	023	026	033	019	025	043	089	114	090	140	164	188	149	142	142	162	178	191	198	206
207	212	218	222	223	218	212	208	206	198	225	188	082	012	000	011	018	035	076	114	143	193	178	165	124	113	119	137	151	145	144	147
198	204	213	220	223	221	217	213	212	211	241	236	151	059	033	051	016	049	105	148	193	204	140	098	094	081	091	106	122	110	117	127
194	200	208	215	219	218	215	212	234	221	179	129	081	041	015	000	006	033	083	128	185	176	103	071	072	054	065	080	107	109	143	172
194	198	205	212	215	214	211	209	208	191	101	036	078	159	180	143	026	007	018	062	151	153	084	058	060	039	050	066	105	123	182	229
181	169	181	185	196	202	184	189	202	197	034	165	236	255	250	251	199	044	001	024	091	146	059	017	027	036	039	044	077	140	204	240
167	163	171	157	159	172	166	180	217	207	120	226	255	255	255	255	255	088	010	008	025	055	035	027	021	019	010	013	059	134	191	213
144	150	160	138	135	154	164	196	224	205	190	239	254	238	249	245	255	094	007	016	007	008	033	013	014	000	000	028	120	206	229	209
189	162	140	119	132	155	171	216	194	183	191	168	241	253	255	255	254	102	007	015	000	009	090	052	012	002	013	070	158	223	232	211
159	140	147	166	188	175	156	196	104	129	133	051	203	255	254	255	252	106	016	007	000	018	134	089	036	049	073	099	120	144	179	208
134	156	200	214	186	120	103	179	058	115	116	007	183	254	240	253	236	070	002	002	005	029	131	088	102	118	137	145	145	157	193	228
210	211	209	172	119	064	082	193	081	130	138	043	192	255	255	255	255	063	063	084	067	039	122	134	171	179	183	189	209	233	239	229
198	170	148	141	154	136	129	195	071	193	110	046	169	248	255	253	189	095	215	255	179	055	111	182	203	214	208	193	208	236	226	191
152	125	149	147	147	155	146	186	069	110	131	042	103	251	250	255	223	228	247	231	255	141	117	209	190	214	176	178	220	207	189	197
153	137	148	111	083	091	101	152	224	224	205	120	153	246	236	249	237	221	168	224	237	167	157	198	177	173	105	093	150	173	180	194
192	186	191	131	070	044	022	046	245	255	255	226	239	255	248	255	255	165	175	204	218	134	163	172	193	170	087	059	118	166	186	197
185	180	191	161	120	089	053	058	194	226	251	255	254	244	239	240	200	147	224	220	149	156	133	151	122	118	075	062	117	173	193	199
166	149	153	146	132	126	130	163	175	196	204	212	214	216	215	186	080	152	182	135	98	132	103	097	076	098	104	102	131	166	173	172
167	150	147	137	116	108	135	184	179	183	188	195	197	193	147	065	109	175	081	062	110	049	078	082	116	135	157	150	149	162	156	152
154	151	155	151	133	119	127	151	173	163	173	161	140	136	096	030	191	167	071	142	106	080	093	140	135	128	144	139	139	157	153	151
169	165	162	161	158	148	132	120	110	088	095	066	043	098	158	185	151	055	160	179	088	134	147	155	167	133	131	123	125	145	139	132
174	153	174	170	139	122	111	119	158	128	081	037	013	014	031	046	016	096	200	165	107	125	129	141	128	122	117	115	114	111	102	095
204	194	190	141	113	138	134	113	064	055	041	028	019	012	007	004	042	123	219	166	094	104	104	111	110	093	075	068	071	071	063	053
185	180	182	128	073	063	040	018	029	028	032	039	045	041	028	017	055	141	229	155	066	072	069	071	061	055	053	061	078	093	101	102
148	148	206	217	152	078	043	061	073	064	058	064	077	085	083	076	048	138	223	138	041	046	044	044	071	083	098	110	119	129	142	152

# ¿Qué es una imagen para el ordenador?

201	196	189	184	184	189	196	201	210	198	231	175	108	95	69	86	82	111	161	167	72	140	168	165	150	161	157	163	145	153	146	153
209	205	199	194	191	192	195	197	202	214	240	143	57	67	50	39	58	74	128	162	100	159	167	155	157	162	158	170	166	184	189	201
216	214	211	207	203	199	197	195	193	216	231	114	22	55	59	31	46	63	116	146	91	133	146	156	160	158	155	173	184	208	220	234
215	217	218	218	215	210	204	200	196	201	213	122	23	26	33	19	25	43	89	114	90	140	164	188	149	142	142	162	178	191	198	206
207	212	218	222	223	218	212	208	206	198	225	188	82	12	0	11	18	35	76	114	143	193	178	165	124	113	119	137	151	145	144	147
198	204	213	220	223	221	217	213	212	211	241	236	151	59	33	51	16	49	105	148	193	204	140	98	94	81	91	106	122	110	117	127
194	200	208	215	219	218	215	212	234	221	179	129	81	41	15	0	6	33	83	128	185	176	103	71	72	54	65	80	107	109	143	172
194	198	205	212	215	214	211	209	208	191	101	36	78	159	180	143	26	7	18	62	151	153	84	58	60	39	50	66	105	123	182	229
181	169	181	185	196	202	184	189	202	197	34	165	236	255	250	251	199	44	1	24	91	146	59	17	27	36	39	44	77	140	204	240
167	163	171	157	159	172	166	180	217	207	120	226	255	255	255	255	255	88	10	8	25	55	35	37	21	19	10	13	59	134	191	213
144	150	160	138	135	154	164	196	224	205	190	239	254	238	249	245	255	94	7	10	7	33	13	14	0	0	28	120	206	229	209	
189	162	140	119	132	155	171	216	194	183	191	168	241	253	255	255	254	102	7	15	10	9	90	52	12	8	13	70	158	223	232	211
159	140	147	166	188	175	156	196	104	129	133	51	203	255	254	255	252	106	16	7	10	18	134	89	36	49	73	99	120	144	179	208
134	156	200	214	186	120	103	179	58	115	116	7	183	254	240	253	236	70	2	3	10	29	131	88	102	118	137	145	145	157	193	228
210	211	209	172	119	64	82	193	81	130	138	43	192	255	255	255	255	63	63	84	67	39	122	134	171	179	183	189	209	233	239	229
198	170	148	141	154	136	129	195	71	193	110	46	169	248	255	253	189	95	215	255	179	55	111	182	203	214	208	193	208	236	226	191
152	125	149	147	147	155	146	186	69	110	131	42	103	251	250	255	223	228	247	231	255	141	117	209	190	214	176	178	220	207	189	197
153	137	148	111	83	91	101	152	224	224	205	120	153	246	236	249	237	221	168	224	237	167	157	198	177	173	105	93	150	173	180	194
192	186	191	131	70	44	22	46	245	255	255	226	239	255	248	255	255	165	175	204	218	134	163	172	193	170	87	59	118	166	186	197
185	180	191	161	120	89	53	58	194	226	251	255	254	244	239	240	200	147	224	220	149	156	133	151	122	118	75	62	117	173	193	199
166	149	153	146	132	126	130	163	175	196	204	212	214	216	215	186	80	152	182	135	98	132	103	97	76	98	104	102	131	166	173	172
167	150	147	137	116	108	135	184	179	183	188	195	197	193	147	65	109	175	81	62	110	49	78	82	116	135	157	150	149	162	156	152
154	151	155	151	133	119	127	151	173	163	173	161	140	136	96	30	191	167	71	142	106	80	93	140	135	128	144	139	139	157	153	151
169	165	162	161	158	148	132	120	110	88	95	66	43	98	158	185	151	55	160	179	88	134	147	155	167	133	131	123	125	145	139	132
174	153	174	170	139	122	111	119	158	128	81	37	13	10	31	46	16	96	200	165	107	125	129	141	128	122	117	115	114	111	102	95
204	194	190	141	113	138	134	113	64	55	41	28	19	12	10	4	42	123	219	166	94	104	104	111	110	93	75	68	71	71	63	53
185	180	182	128	73	63	40	18	29	28	32	39	45	41	28	17	55	141	229	155	66	72	69	71	61	55	53	61	78	93	101	102
148	148	206	217	152	78	43	61	73	64	58	64	77	85	83	76	48	138	223	138	41	46	44	44	71	83	98	110	119	129	142	152

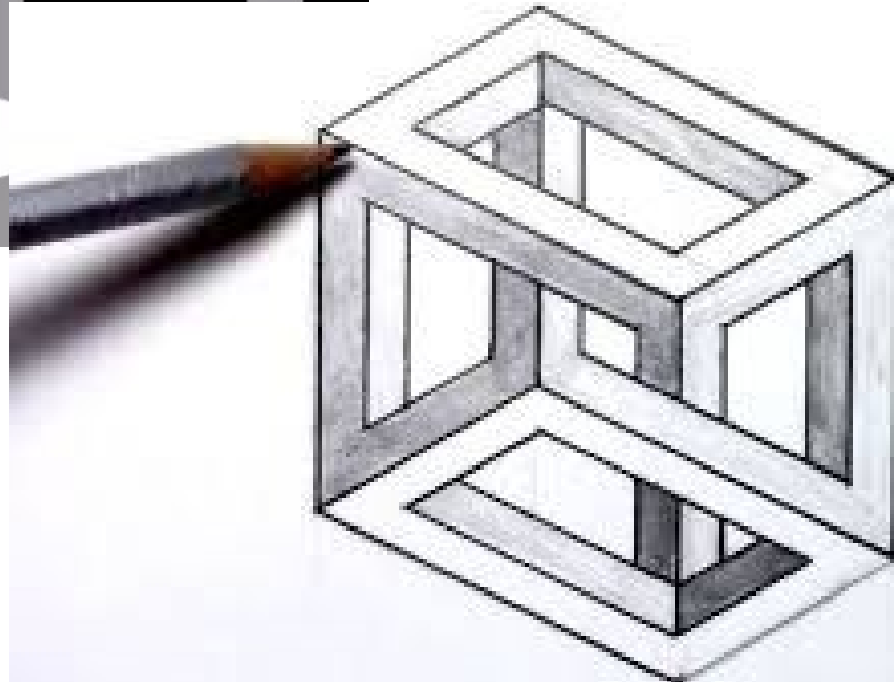
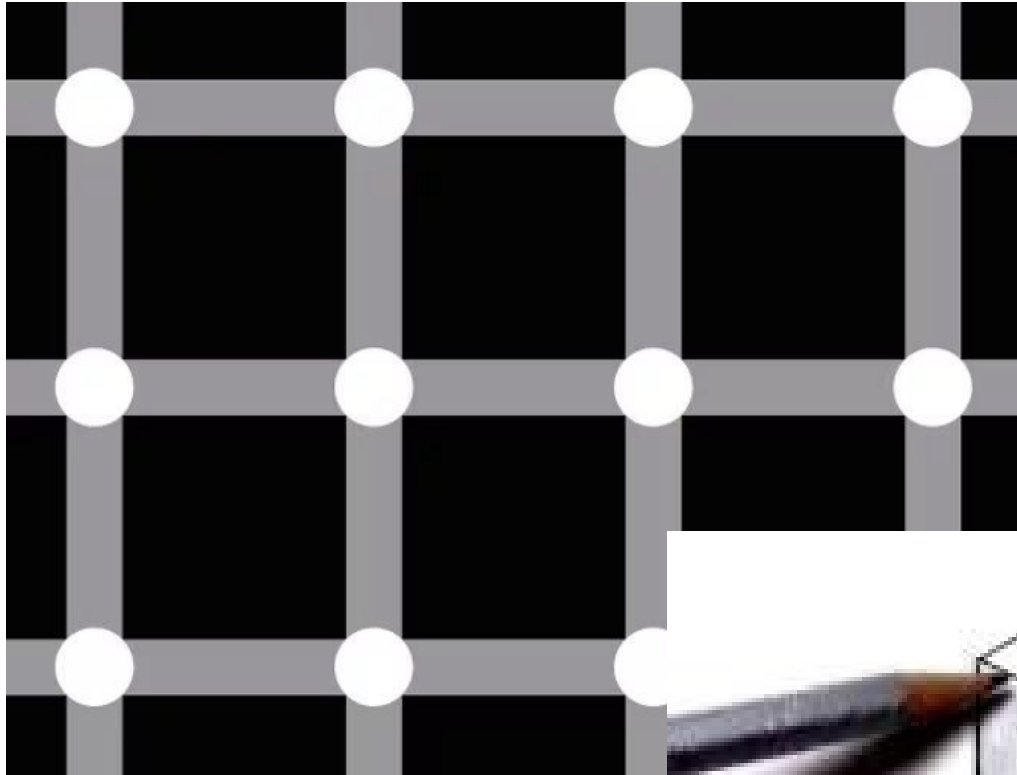
# Y ahora, ¿qué?

- Tenemos los datos:  
hay que procesarlos.
- Los métodos de visión artificial intentan procesar dichos datos
- Ojo: el mundo nos engaña



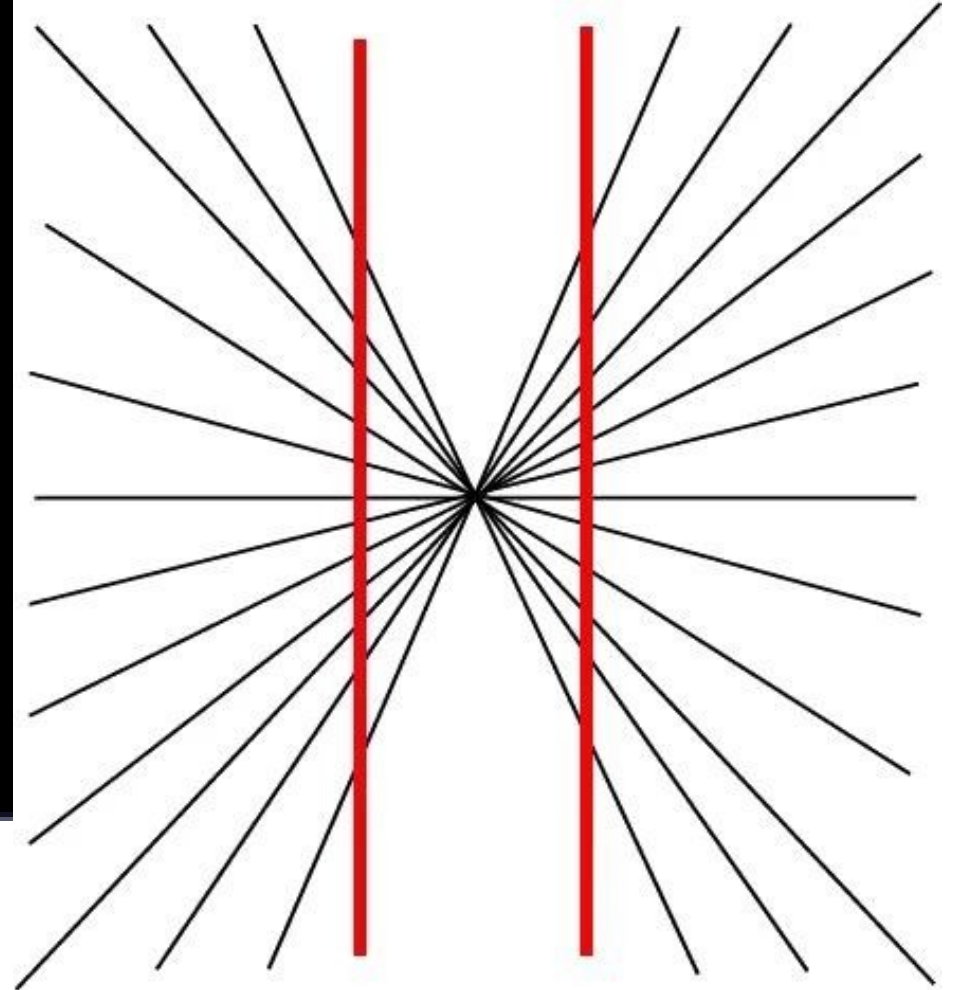
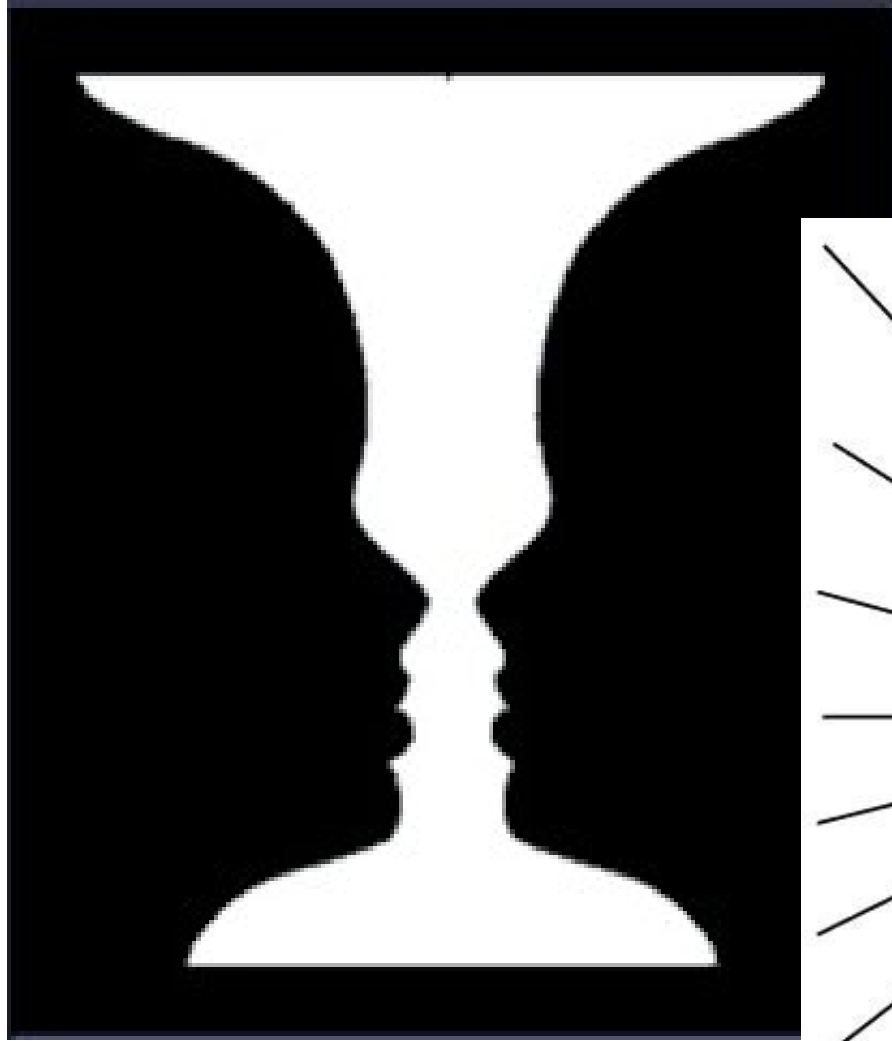


# Y ahora, ¿qué?





# Y ahora, ¿qué?



# Procesar una imagen

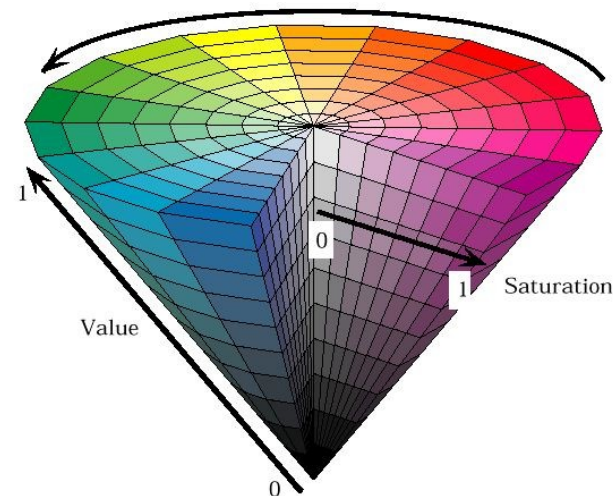
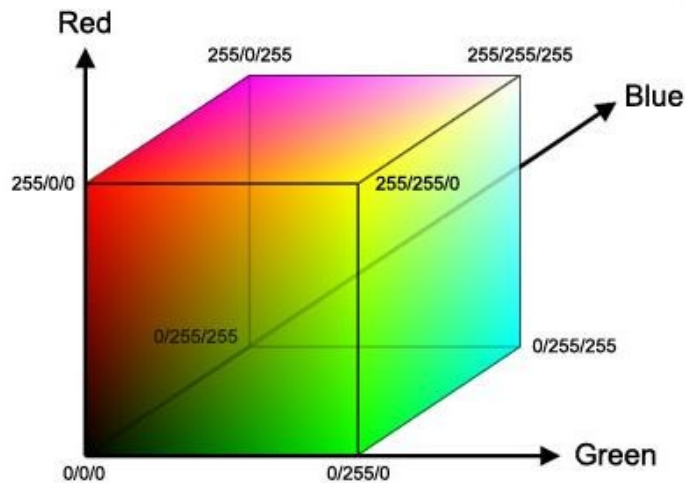
- La mayoría de métodos que veremos, usan los píxeles de la imagen para realizar algún tipo de operación matemática sobre ellos
- Haremos referencia al valor de un píxel con coordenadas  $x, y$  como  $I(x,y)$
- La idea es procesar una imagen  $I(x,y)$  y conseguir otra imagen  $G(x,y)$ , resultado de aplicar algún método o algoritmo. También es posible que el resultado sea otro tipo de datos: un entero, un objeto, una zona de la imagen, etc.

# Lecturas sobre visión recomendadas

- Computer vision homepage:  
<http://www.cs.cmu.edu/~cil/vision.html>
- OpenCV: <http://opencv.willowgarage.com/wiki/>
- CVOnline: <http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- Libro: Computer vision: a modern approach  
<http://luthuli.cs.uiuc.edu/~daf/book/book.html>

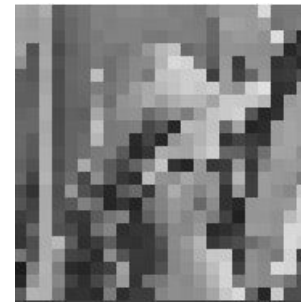
# Modelos de color

- Dependiendo de los datos de la matriz de la imagen, tendremos distintos tipos de imagen
  - Binaria: valores 0 y 1
  - Gris: valores 0 a 255 (usualmente)
  - Color RGB: tres valores entre 0 y 255
- En cuanto al color, existen otros modelos (HSV)



# Resolución de una imagen

- Una imagen tendrá un tamaño  $M \times N$
- Podemos ampliar o reducir la imagen.
  - Ampliar: se repiten píxeles
  - Reducir: al perder píxeles en la reducción, podemos: escoger un píxel o promediar entre los píxeles eliminados



# Definiciones

- Cambio de brillo: aumenta o reduce el valor de cada píxel



- Ruido: errores que pueden degradar la imagen
  - Se produce: adquisición, transmisión o el procesado de la imagen
  - Diremos que un método de procesamiento es robusto ante el ruido, cuando genera los mismos resultados con y sin ruido

# Convolución

- Operación matemática aplicada a dos funciones,  $f(x)$  y  $g(x)$  que tiene como resultado otra función  $h(x)$
- Está definida, en el caso continuo, como:

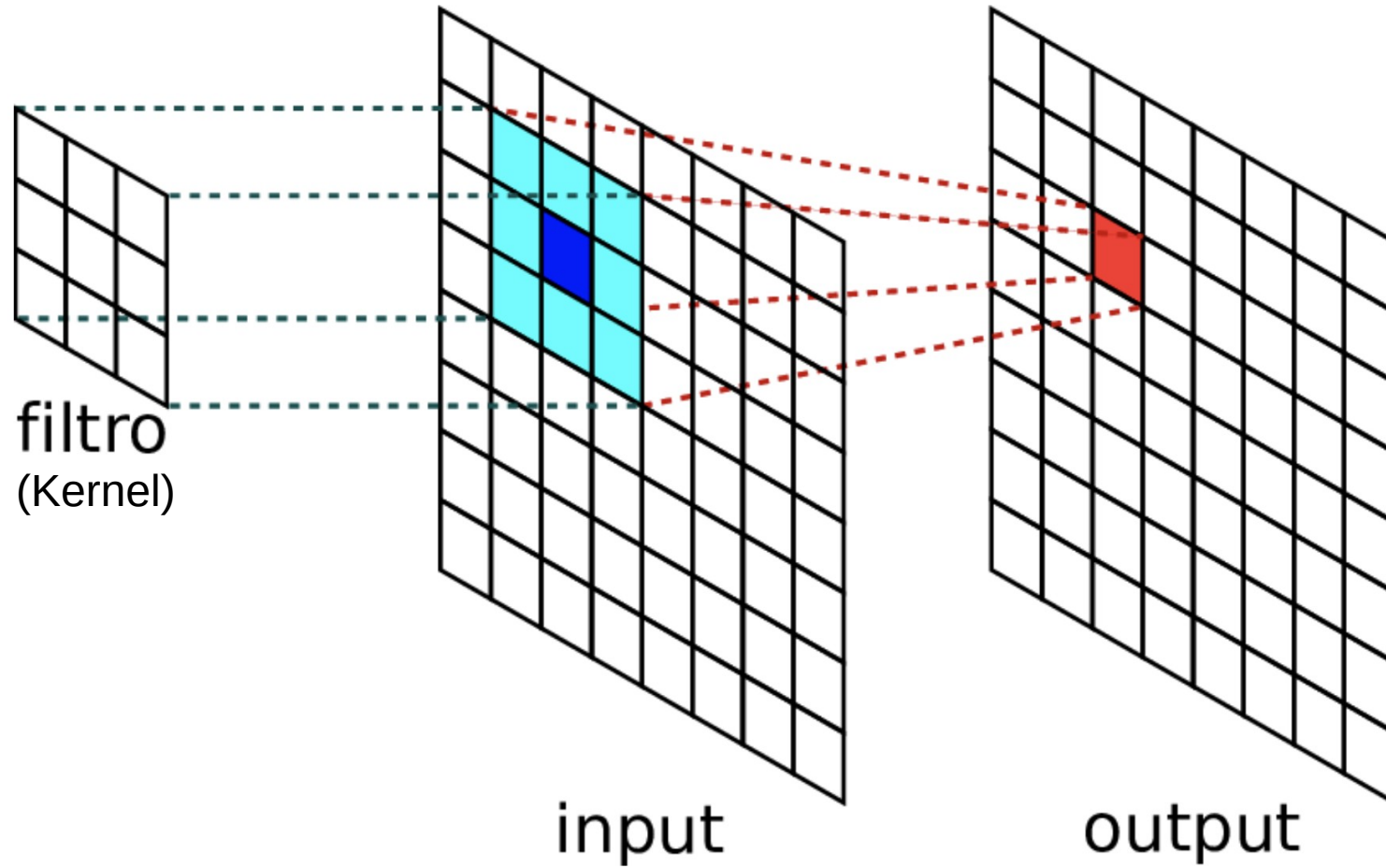
$$f(x) * g(x) = h(x) = \int_{-\infty}^{\infty} f(z)g(x - z)dz$$

- Y en el caso discreto y en dos dimensiones:

$$f(x, y) * g(x, y) = h(x, y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} f(m, n)g(x - m, y - n)$$



# Convolución



# Aplicación de la convolución

- Para cada píxel de la imagen, se “superpone” la máscara (kernel) sobre la imagen
- El nuevo valor del píxel es el resultado de sumar la multiplicación de los valores de la máscara por los valores de los píxeles
- ¿Tamaño del resultado?
  - Sin padding se reduce
  - Con padding (relleno de 0's al rededor de la imagen) mismo tamaño

Imagen de entrada (4x4)

1	2	1	3
0	3	2	1
2	1	0	2
1	2	3	1

Kernel (3x3)

1	0	1
0	1	0
1	0	1

Resultado (2x2)

# Aplicación de la convolución

- Para cada píxel de la imagen, se “superpone” la máscara (kernel) sobre la imagen
- El nuevo valor del píxel es el resultado de sumar la multiplicación de los valores de la máscara por los valores de los píxeles

Imagen de entrada (4x4)

1	2	1	3
0	3	2	1
2	1	0	2
1	2	3	1

Kernel (3x3)

1	0	1
0	1	0
1	0	1



Resultado (2x2)

7	10
7	7

Notas importantes:

1. La matriz resultado es más pequeña (2x2) debido al tamaño del kernel (3x3)
2. Ejemplo de cálculo para el valor superior izquierdo (7):  

$$1 \times 1 + 2 \times 0 + 1 \times 1 + 0 \times 0 + 3 \times 1 + 2 \times 0 + 2 \times 1 + 1 \times 0 + 0 \times 1 = 7$$
3. El kernel se desliza sobre la imagen, calculando cada valor de salida

# Filtros lineales

- Filtro de media

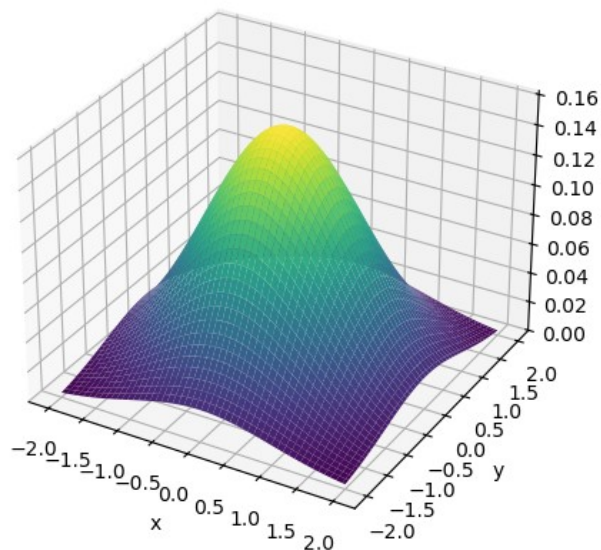
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

- Filtrado Gaussiano

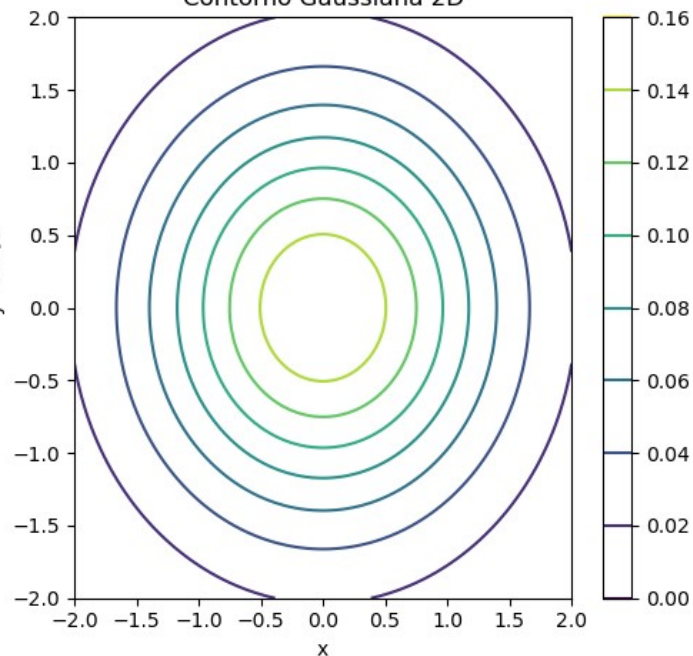
Para sigma=1

$$g(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x^2 + y^2)}{2\sigma^2}}$$

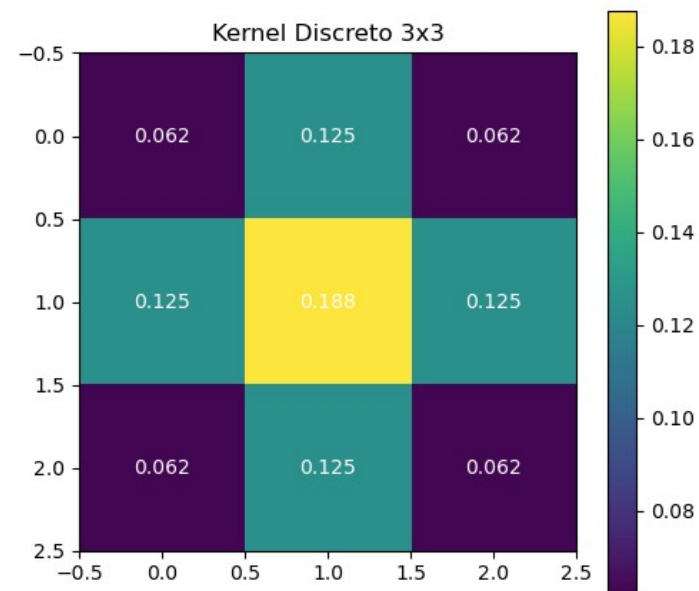
Gaussiana 2D Continua



Contorno Gaussiana 2D



Kernel Discreto 3x3



# Filtros lineales

- Filtro de media

1/9

1	1	1
1	1	1
1	1	1

- Filtrado Gaussiano

Crea un filtro de convolución con una función gaussiana de media 0 y varianza  $\sigma$

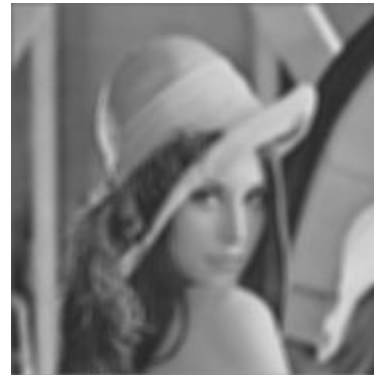
$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x^2 + y^2)}{2\sigma^2}}$$

1/16

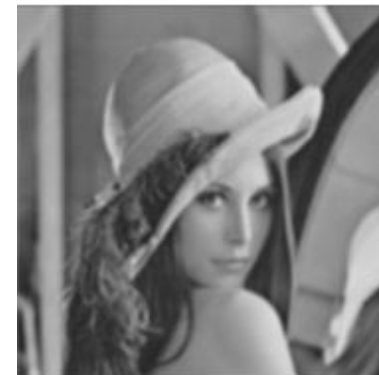
1	2	1
2	3	2
1	2	1



Original



Media



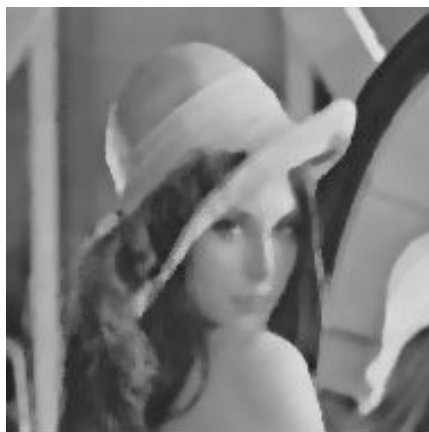
Gaussiano

# Filtro de mediana y resultados

Filtro no lineal de mediana: calcula la mediana en una vecindad



Original



Mediana



Media

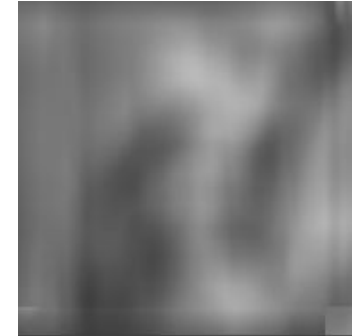
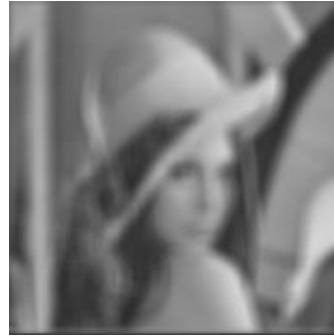
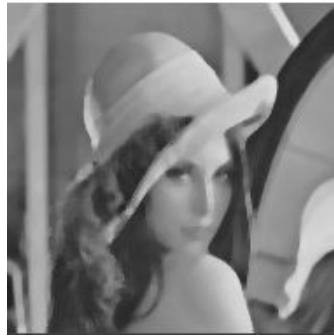


Gaussiana



# Efecto del tamaño de la máscara

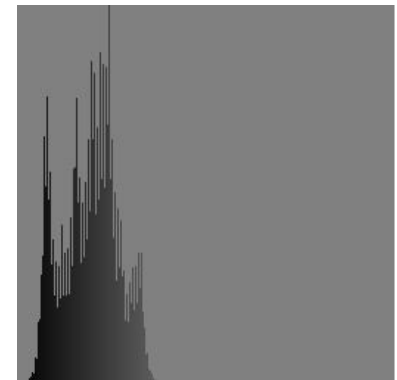
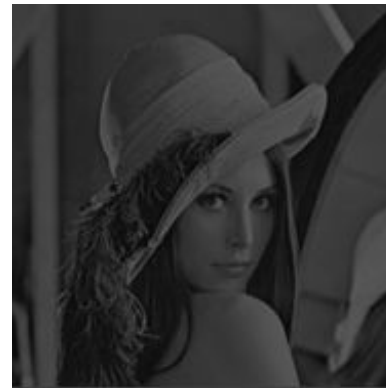
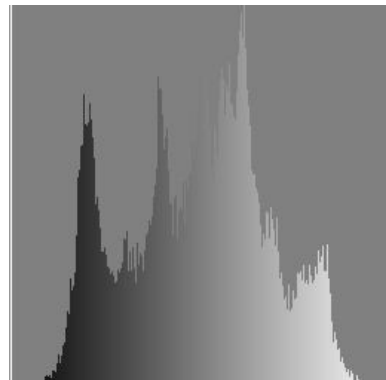
- Hay que tener en cuenta que el tamaño de la máscara usada o de la vecindad escogida (3x3, 5x5, 10x10) puede tener un efecto no deseado en el resultado, aunque hay veces que es lo que se pretende





# Histograma de la imagen

- Un histograma es una función definida en  $K$  niveles (los valores de intensidad de la imagen) e indica el número de píxeles en la imagen que tienen ese valor
- Si está normalizado (la suma de todos los valores es 1.0) nos indica la probabilidad de que un píxel tenga un determinado valor
- ¿Qué ocurre en el histograma si aumentamos el brillo?



# Operaciones con el histograma

- Binarización: indicar uno o dos umbrales y poner a 1 los píxeles que estén por debajo o por encima del umbral o entre dos umbrales, y a cero el resto
- También se puede hacer con imágenes de color, aplicando umbrales a cada banda de color



Binarización con  
umbrales 128 y 255

# Ecualización del histograma

- CDF (Cumulative Distribution Function) o Función de Distribución Acumulativa se calcula:
  - Histograma:
    - Se calcula el histograma  $h(i)$  que cuenta la frecuencia de cada nivel de intensidad
    - Para una imagen de 8 bits,  $i$  va de 0 a 255
  - Calculo acumulado:

$$CDF(k) = h_a = \sum_{i=0}^k h(i)$$

Histograma  $h(i)$ :

$h(0) = 10$

$h(1) = 20$

$h(2) = 15$

$h(3) = 5$

CDF(k):

$cdf(0) = 10$

$cdf(1) = 10 + 20 = 30$

$cdf(2) = 30 + 15 = 45$

$cdf(3) = 45 + 5 = 50$

# Ecualización del histograma

- Cuando tenemos una imagen con poco contraste podemos “expandir” el histograma
- El algoritmo es el siguiente:
  - Calculamos el histograma de la imagen
  - Calculamos el histograma acumulado
  - Se produce un mapeo con la siguiente fórmula:

$$I'(x, y) = \frac{h_a(I(x, y)) - \min_v}{totalpix - \min_v} \cdot (L - 1)$$

donde  $L$  es el número de niveles de intensidad y  $\min_v$  es el valor de  $h_a$  correspondiente al menor valor de gris.

# Ecualización del histograma

		Calculos Ecualización					
	j	h(j)	ha(j)	I'	I		
1	0	790	790	0	0	minv	790
2	1	1023	1813	2,16606171	2	totalpix	4096
3	2	850	2663	3,96581972	4	L	8
4	3	656	3319	5,35480944	5		
5	4	329	3648	6,05142166	6		
6	5	245	3893	6,57017544	7		
7	6	122	4015	6,82849365	7		
8	7	81	4096	7	7		

$$I'(x, y) = \frac{h_a(I(x, y)) - \min_v}{\text{totalpix} - \min_v} \cdot (L - 1)$$

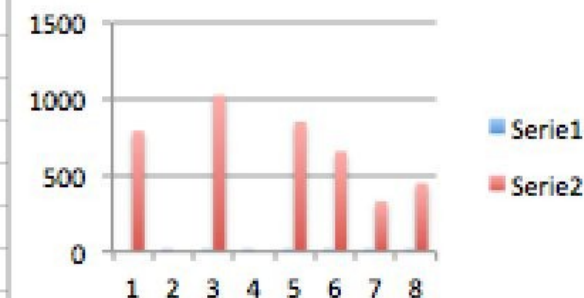
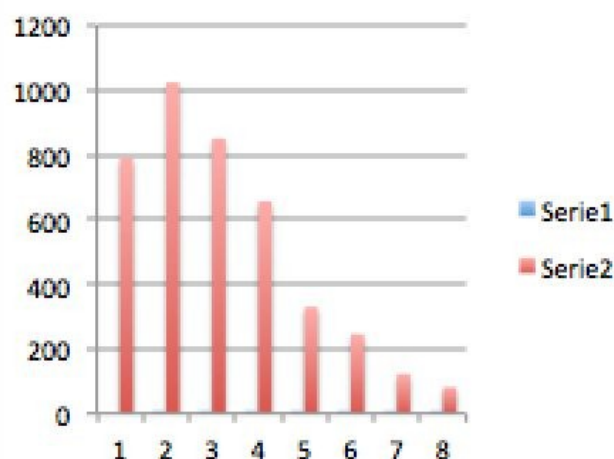
$$(1813 - 790) / (4096 - 790) * (8 - 1)$$

Original

1	0	790
2	1	1023
3	2	850
4	3	656
5	4	329
6	5	245
7	6	122
8	7	81

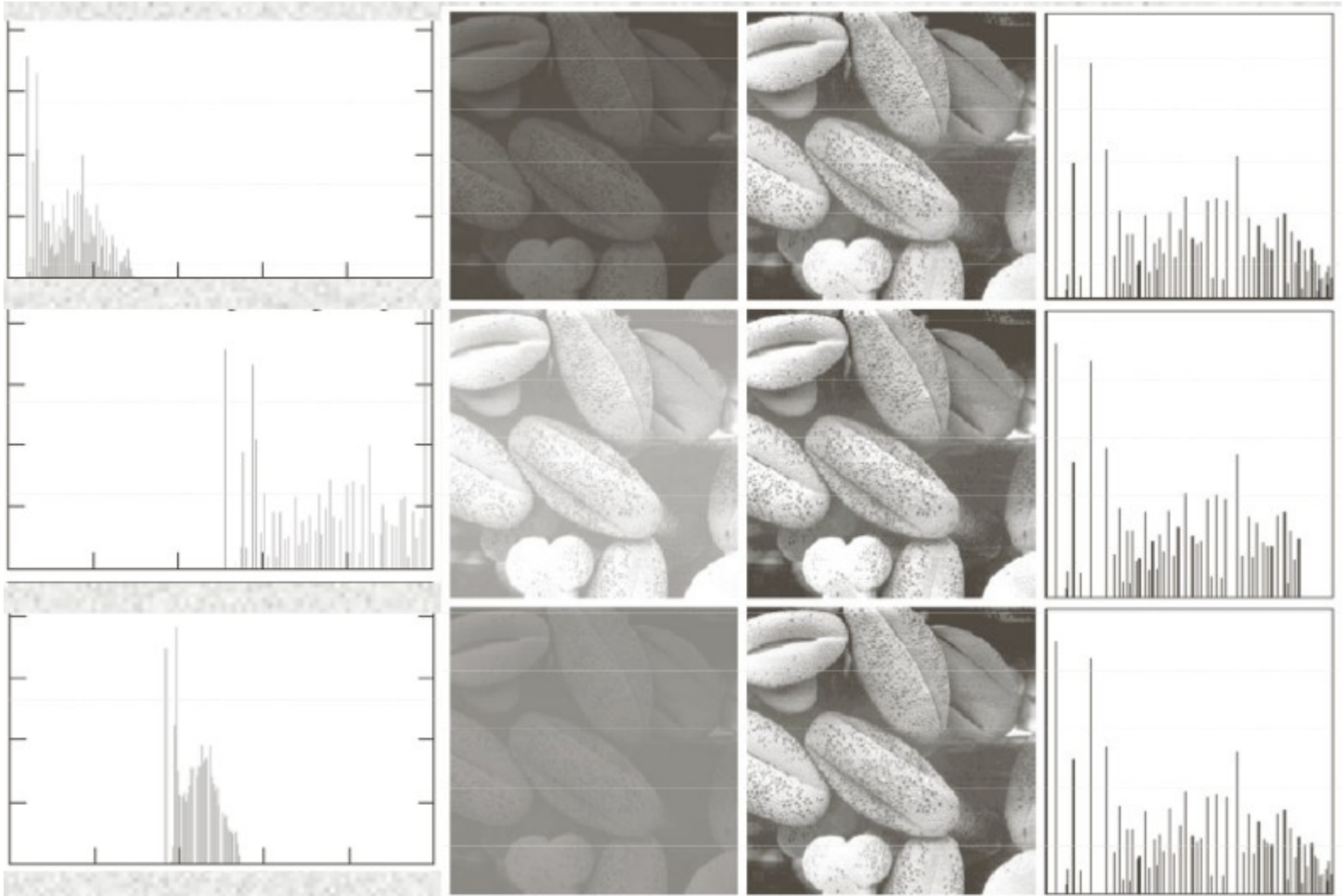
Ecualizado:

1	0	790
2	1	0
3	2	1023
4	3	0
5	4	850
6	5	656
7	6	329
8	7	448

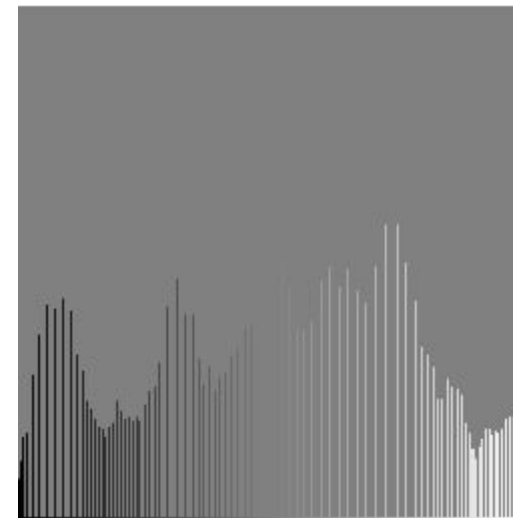
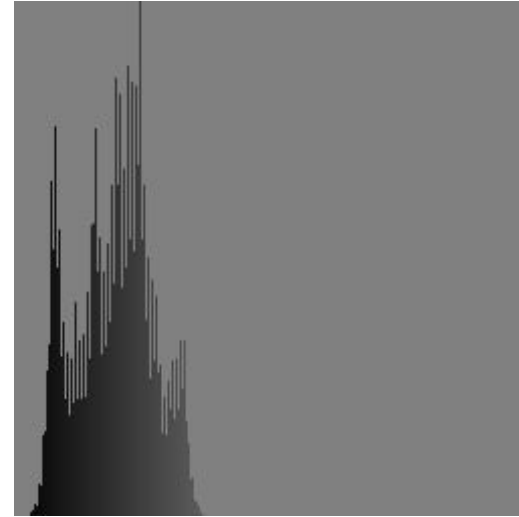




# Resultado de la ecualización



# Resultado de la ecualización





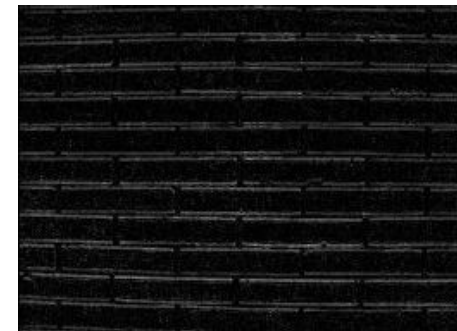
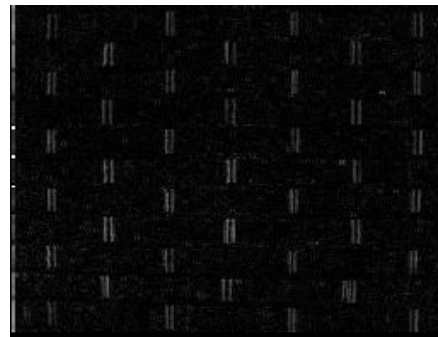
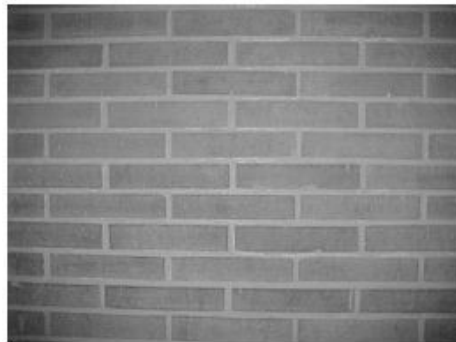
# Aristas (“Edges”)

- Edges:
  - Puntos de alto contraste en la imagen, es decir donde la intensidad de la imagen aumenta o disminuye bruscamente.
  - Edges=puntos de alta derivada en valor absoluto.  
Derivamos numéricamente la imagen (entendida como función bidimensional de variable real) para identificar las aristas.
- Operador básico:
  - Diferencias de primer orden.
  - En dos fases: horizontal y vertical.

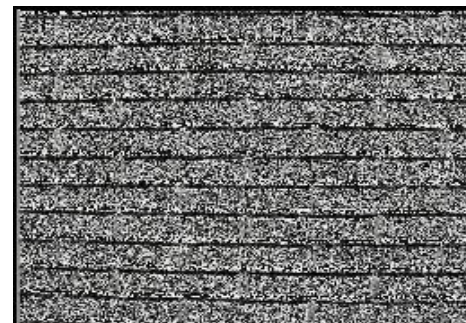
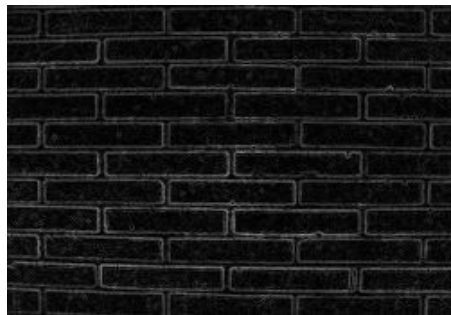
$$\nabla I = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} f(x+1, y) - f(x, y) \\ f(x, y+1) - f(x, y) \end{bmatrix}$$

# Aristas

- Operadores de gradiente:
- Diferencias de primer orden:



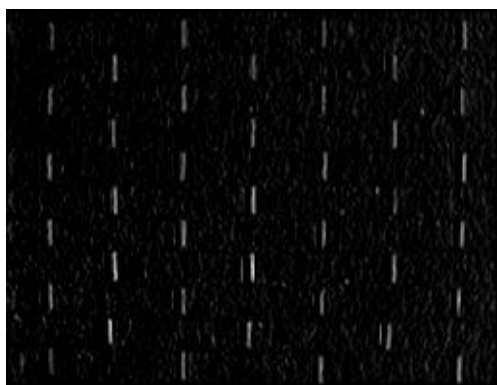
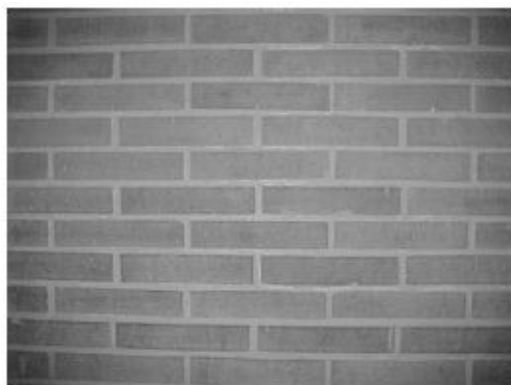
- Podemos calcular la magnitud y la orientación del gradiente



# Aristas

- Operadores de gradiente:
- Operadores de Sobel: (promediado)

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



# Detector de Canny

- Fundamentos:

- Criterios (formalización):

- Buena detección: minimizar el número de falsos positivos y falsos negativos.
    - Buena localización: las aristas deben marcarse en el lugar real.
    - Respuesta única: como resultado deben generarse aristas de un pixel de anchura.

- Supuestos:

- Ruido gaussiano y aristas tipo “escalón”.

- Resultado:

- Filtro óptimo: es la primera derivada de la gaussiana.

$$\nabla(G_{\sigma} * I) = \nabla(G_{\sigma}) * I$$

# Aristas (“Edges”)

- Identificación de contornos:
  - Detección:
    - Identificación de píxeles candidatos a formar parte de un contorno. Normalmente aquellos con “alto contraste local”.
    - Filtros lineales. Como veremos, existen máscaras de convolución que aplicadas sobre la imagen destacan los píxeles candidatos e inhiben el resto.
    - Selección o eliminación de no-máximos. Si resulta que hay más de un candidato en una misma vecindad hay que elegir uno de ellos y eliminar el resto.
  - Agrupamiento:
    - Proceso de “local” a “global” que permite enlazar píxeles vecinos siempre y cuando se cumplan ciertas reglas de “continuidad” y “curvatura”.

# Detector de Canny. Resumen

El detector Canny es considerado, el más efectivo a la hora de detectar bordes, debido a su eficacia. Este detector, realiza su trabajo de la siguiente manera:

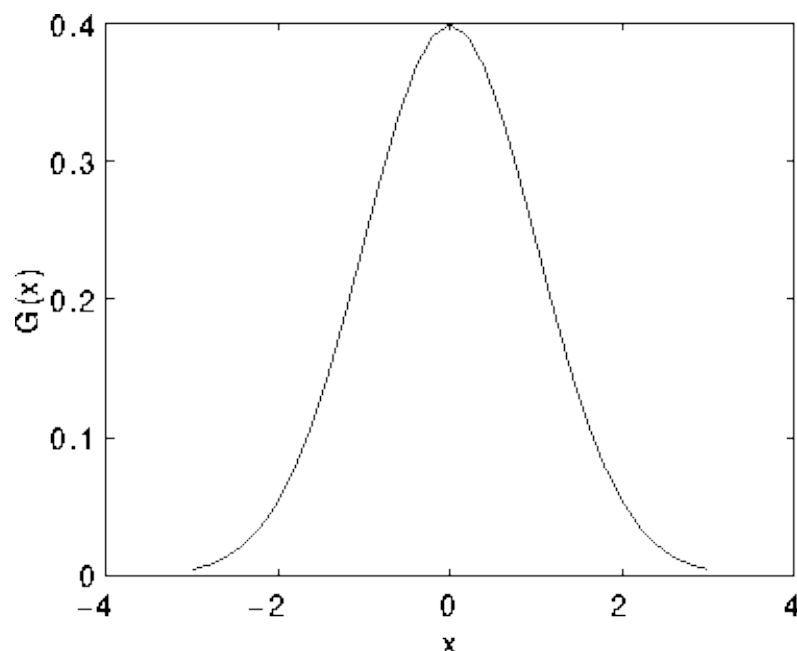
- 1) La imagen se suaviza usando un filtro Gausiano con una desviación estándar, esto para reducir el ruido. El gradiente local, y la dirección del borde, son computadas en cada punto. El detector Sobel, puede ser usado para computar  $G_x$  y  $G_y$ .
- 2) Se define un punto de borde, como un punto cuyo peso es localmente máximo en la dirección del gradiente.
- 3) Los puntos de borde determinados en (2) originan crestas en la dirección de crecimiento del gradiente de la imagen.
  - El algoritmo luego rastrea a lo largo de la cima de estas crestas, y lleva a cero los píxeles que no están en realidad sobre la cima de la cresta, originando una línea delgada en la salida, un proceso conocido como supresión no máxima.
  - Los píxeles de crestas son luego comparados usando dos umbrales,  $T_1$  y  $T_2$  con  $T_1 < T_2$ . Los píxeles de crestas con valores mayores que  $T_2$  se dice que son “probables candidatos” para ser píxeles de borde.
  - Los píxeles en crestas con valores entre  $T_1$  y  $T_2$  se dice que son “candidatos poco probables” para ser píxeles de borde.
- 4) Por último, el algoritmo realiza la unión de los píxeles, incorporando “candidatos débiles” que están 8-conectados a los píxeles “probables”.



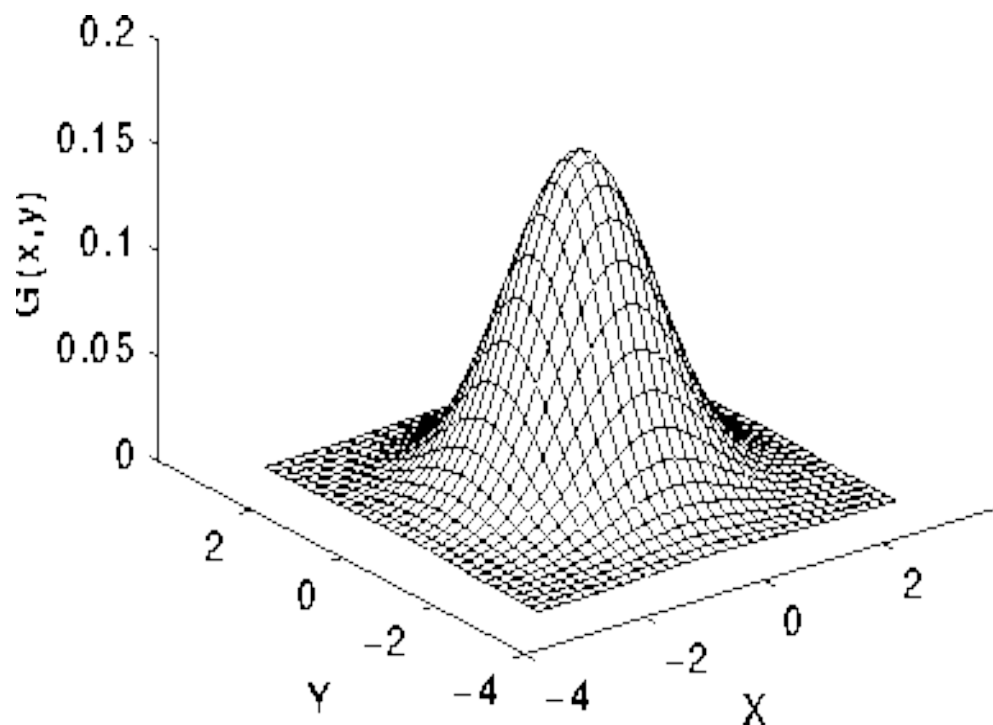
# Detector de Canny

- Filtrado gaussiano:
  - Funciones 1D y 2D:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$





# Detector de Canny

- Filtrado gaussiano:
  - Máscara de convolución:

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

# Detector de Canny

## • Algoritmo:

- Convolució:
  - Filtro gaussiano
  - Gradiente
- Vectorizació
  - Mòdul
  - Direcció
- Post-proceso
  - Supresión no-máximos.
  - Histéresis.

**Algoritmo CANNY {**

$S \leftarrow G_{\sigma} * I$

$[S_x \ S_y] \leftarrow \nabla(S)$

**Para** cada *pixel*  $(i, j)$

$|\nabla S(i, j)| \leftarrow \sqrt{S_x^2(i, j) + S_y^2(i, j)}$

$S_{\phi}(i, j) \leftarrow \arctan \frac{S_x(i, j)}{S_y(i, j)}$

**SUPRESIÓN**

**HYSTERESIS**

**Devolver**  $\{|\nabla S|, S_{\phi}\}$

}

# Detector de Canny

- Supresión no-máximos:
  - Tomar un conjunto discreto de direcciones.
  - Para cada píxel:
    - Considerar la dirección más similar
    - Comprobar si es mejor que sus vecinos en dicha dirección

**Algoritmo** SUPRESIÓN DE NO-MÁXIMOS {

$d_i \leftarrow [0^\circ, 45^\circ, 90^\circ, 135^\circ]$

**Para** cada *pixel*  $(i, j)$

$\hat{d}_k \leftarrow \arg \min_{d_k} |d_k - S_\phi(i, j)|$

**Si**  $|\nabla S(i, j)|$  es menor que uno de sus dos vecinos en la dirección  $\hat{d}_k$

$S_e(i, j) \leftarrow 0$

**Si no**

$S_e(i, j) \leftarrow |\nabla S(i, j)|$

**Devolver**  $S_e$

}

# Detector de Canny

## Histéresis:

### Algoritmo HYSTERESIS {

$p \leftarrow$  Umbral a nivel alto

$q \leftarrow$  Umbral a nivel bajo

$L_{visitados} \leftarrow \phi$

$L_{bordes} \leftarrow \phi$

**Para** cada *pixel*  $(i, j)$

**Si**  $(i, j) \notin L_{visitados}$  y  $S_e(i, j) > p$

$(k, l) \leftarrow (i, j)$

$B_{ij} \leftarrow \phi$

**Hacer**

$B_{ij} \leftarrow B_{ij} \cup \{(k, l)\}$

$L_{visitados} \leftarrow L_{visitados} \cup \{(k, l)\}$

$(k, l) \leftarrow$  Pixel adyacente a  $(i, j)$  en dirección perpendicular a  $S_\phi(k, l)$

**Mientras**  $S_e(k, l) > q$

$L_{bordes} \leftarrow L_{bordes} \cup B_{ij}$

**Devolver**  $L_{bordes}$

}

### Fijar umbrales:

- Primario (p)
- Nivel bajo (q)

### Filtrado por p:

- Eliminar aristas que no alcancen el umbral alto.

### Añadir por q:

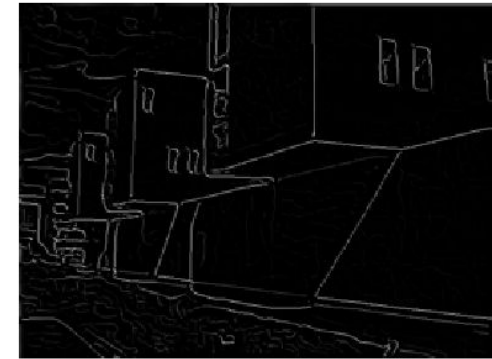
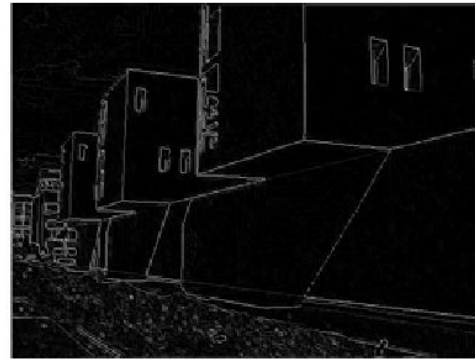
- Si  $> p$ , completar la arista con adyacentes mientras  $> q$

# Detector de Canny

## Resultados:

Sigma = 1

Sigma = 4



(a)

(b)

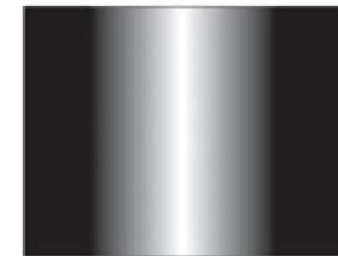
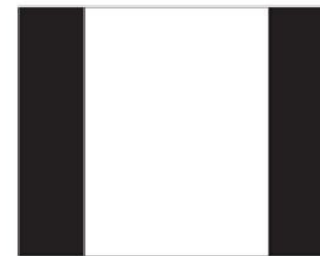
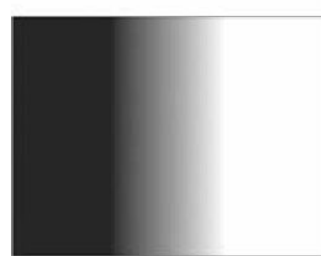
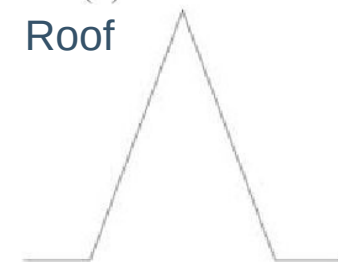
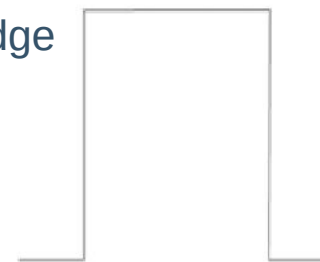
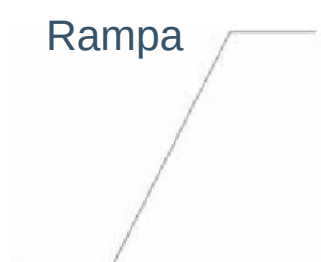
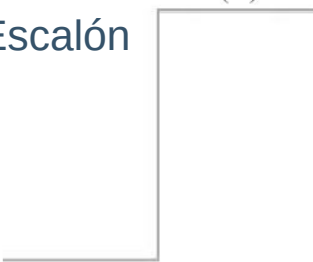
(c)

Escalón

Rampa

Ridge

Roof



(d)

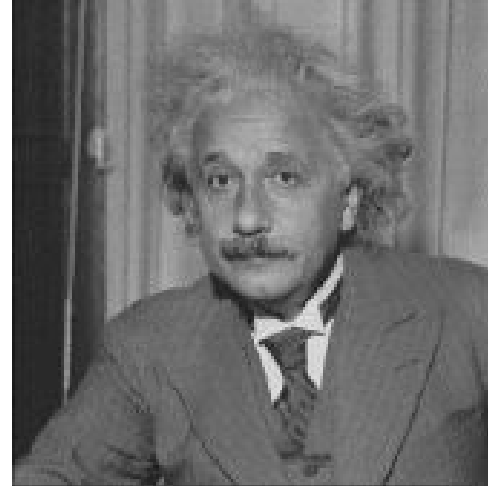
(e)

(f)

(g)

# Detector de Canny

- Trade-off:
  - Debido al filtro lineal (convolución)
  - No es posible conjugar “detección” y “localización”
  - Si aumentamos “sigma” reducimos ruido pero difuminamos los bordes y perdemos calidad en la localización.
  - La anchura de la gaussiana controla el nivel de ruido pero también el de detalle.



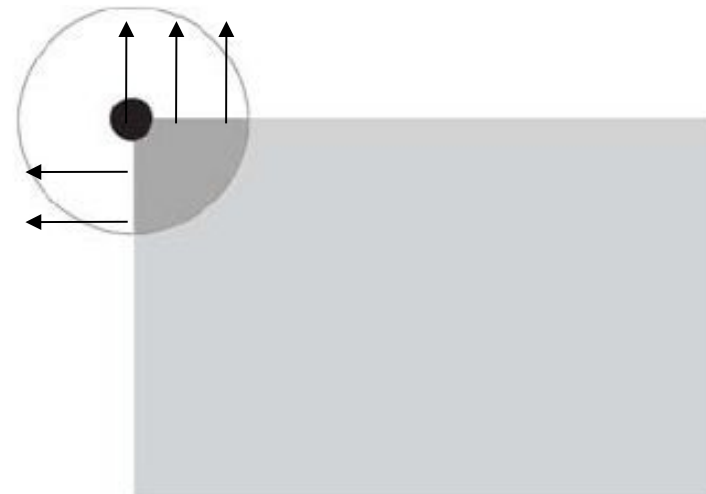
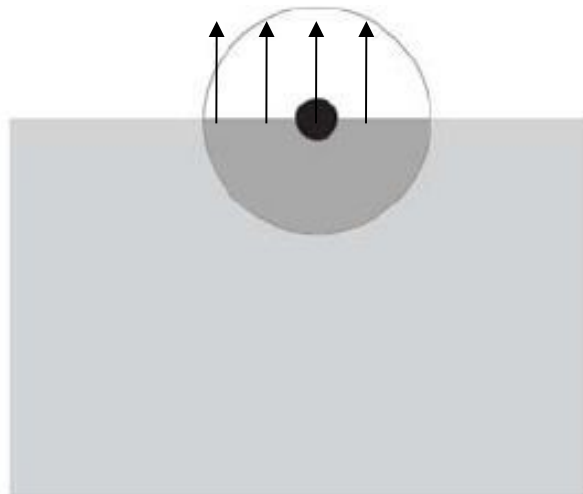


# Puntos esquina o “corners”

- Fundamentos:
  - Definición y utilidad:
    - En un contorno: puntos de elevada curvatura (discontinuidad en la orientación del vector gradiente)
    - En uniones (“junctions”): punto de intersección de varias aristas, normalmente con un significado geométrico (oclusión, etc).
  - Tipos de detectores:
    - Basados en “edges”: a partir de la información de curvatura.
    - Basados en “niveles de gris”: utilizan directamente el nivel de gris en una vecindad del píxel candidato para encontrar evidencia de que hay una esquina.
  - Ejemplos:
    - Nitzberg/Harris

# Detector Nitzberg-Harris

- Principios de N-H:
  - Gradiente + filtrado lineal (gaussiano):
    - Construir una matriz 2x2 que codifica las estadísticas locales de gradiente y extraer sus autovalores y autovectores.
  - Se trata de determinar si existe o no alguna dirección de gradiente dominante. En un corner NO EXISTE dicha dirección.



# Detector Nitzberg-Harris

- Matriz de momentos:
  - Construcción:
    - Calcular gradientes horizontal y vertical para cada uno de los píxeles de la vecindad
    - Aplicar promediado y registrar la matriz para cada punto.

$$I_x(x, y) = \frac{I(x+1, y) - I(x-1, y)}{2}$$

$$I_y(x, y) = \frac{I(x, y+1) - I(x, y-1)}{2}$$

$$\nabla I(x, y) = \begin{bmatrix} I_x(x, y) \\ I_y(x, y) \end{bmatrix}$$

$$A(\mathbf{x}) = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2(\mathbf{x}) & I_x I_y(\mathbf{x}) \\ I_x I_y(\mathbf{x}) & I_y^2(\mathbf{x}) \end{bmatrix} \quad w(x, y) = g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

# Detector Nitzberg-Harris

## Matriz de momentos:

- Donde la matriz  $A(x, y)$  captura la estructura de la intensidad de la vecindad local.
- Sean  $\lambda_1$  y  $\lambda_2$  los valores propios de la matriz  $A(x, y)$ . Los valores propios forman una descripción rotacionalmente invariante.
- Autovalores y autovectores:  $(\lambda_1, e_1)$ ,  $(\lambda_2, e_2)$ 
  - $(\lambda_1, e_1)$ : El autovalor indica la fuerza de la dirección dominante y el autovector indica dicha dirección.
  - $(\lambda_2, e_2)$ : El autovalor indica la fuerza de la dirección secundaria y el autovector indica una dirección perpendicular a la principal.
- Interpretación:
  - Zona uniforme: no hay gradiente y entonces los autovalores son ambos cero.
  - Edges: solo hay una dirección primaria y no hay secundaria por lo que el segundo autovalor será cero.
  - Corners: Ambos autovalores son significativos, el autovector segundo se utiliza como “*cornerness*” y el primero como “*edgeness*”.

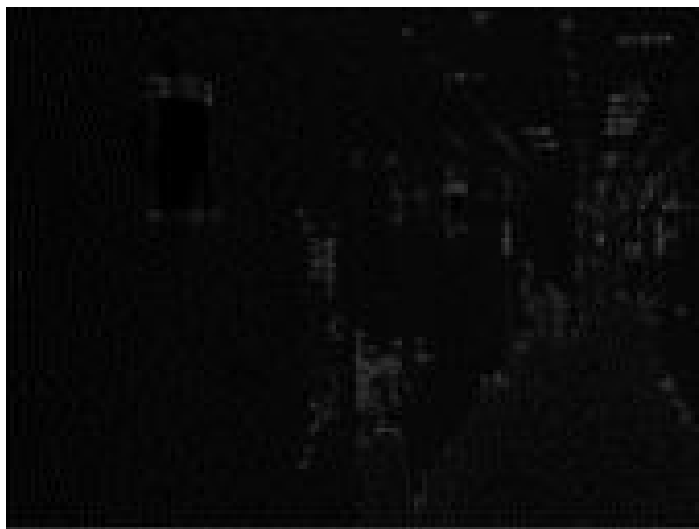
# Detector Nitzberg-Harris



original



edgeness



cornerness



corners

# Referencias

- Detector de Harris [http://en.wikipedia.org/wiki/Harris\\_affine\\_region\\_detector](http://en.wikipedia.org/wiki/Harris_affine_region_detector)
- Comparativa de distintos detectores de características K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, A comparison of affine region detectors. In IJCV 65(1/2):43-72, 2005
- Histograma de una imagen <http://es.wikipedia.org/wiki/Histograma>
- Ecualización del histograma [http://es.wikipedia.org/wiki/Ecualizaci%C3%B3n\\_del\\_histograma](http://es.wikipedia.org/wiki/Ecualizaci%C3%B3n_del_histograma)
- Algoritmo de Canny: Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986