

Τοπικά Χαρακτηριστικά Εικόνων

Local Image Features

Μάθημα: Υπολογιστική Οραση

Κ. Δελήμπασης

- Αυτόματη εξαγωγή χαρακτηριστικών σημείων εικόνας, όπως:
 - Ακμές –Edges (βάσει παραγώγων 1^{ης} και 2^{ης} τάξης –έχει καλυφθεί)
 - Γωνίες –Corners (Harris detector)
 - Blobs (Hessian matrix)
 - Ridges (Hessian matrix)
- Αμεταβλητότητα χαρακτηριστικών – κλίμακα της εικόνας
 - LoG, DoG, Harris – Laplace, KLT (Kanade-Loukas-Tomasi)
- Εφαρμογές
 - Εύρεση ομόλογων σημείων
 - Χωρική ταύτιση εικόνων –image registration
 - Παρακολούθηση αντικειμένων σε video (object tracking)

Ανίχνευση γωνιών

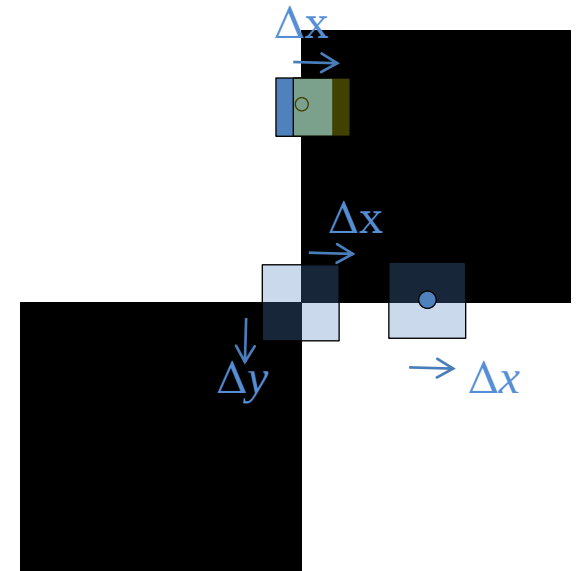
- Εστω εικόνα $I(x,y)$ και η μετατοπισμένη εικόνα $I(x+\Delta x, y+\Delta y)$

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y =$$

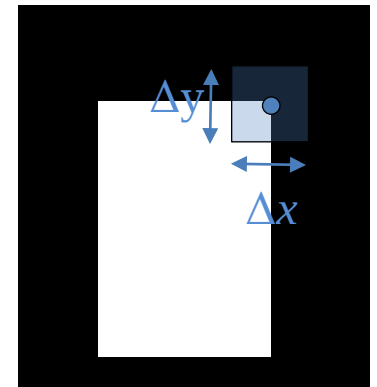
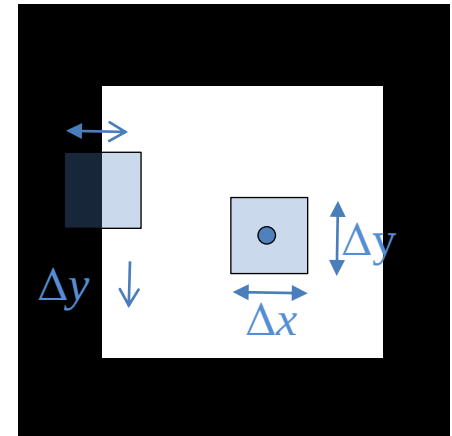
$$I(x, y) + I_x \Delta x + I_y \Delta y$$

- Υπολογίζουμε τη διαφορά τους γύρω από σημείο (x_0, y_0) , με χρήση ενός παραθύρου w .

$$\begin{aligned} E(u, v) &= \sum_{(x,y) \in W} [I(x + \Delta x, y + \Delta y) - I(x, y)]^2 \\ &\approx \sum_{(x,y) \in W} [I_x \Delta x + I_y \Delta y]^2 = \sum_{(x,y) \in W} I_x^2 \Delta x^2 + 2I_x I_y \Delta x \Delta y + I_y^2 \Delta y^2 \end{aligned}$$

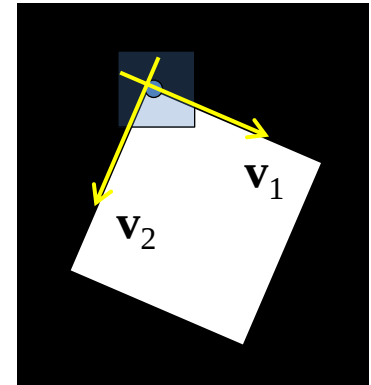


- Όταν το τρέχον pixel είναι σε περιοχή flat
- Όταν το τρέχον pixel είναι σε ακμή:
- Όταν το τρέχον pixel είναι σε οριζόντια γωνία



- Όταν η γωνία έχει τυχαία διεύθυνση, η $\max$ και η $\min$ τιμή του E παρατηρείται όταν η μετακίνηση γίνεται κατά μήκος 2 κάθετων διευθύνσεων.
- Για να βρούμε τις διευθύνσεις αυτές και τις ακρότατες τιμές $\max$, $\min$, γράφουμε το E σε μορφή πίνακα:

$$E = (\Delta x \quad \Delta y) w(x, y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$



- Ορίζουμε τον πίνακα **M**:

$$\mathbf{M} = w(x, y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} = \begin{pmatrix} \sum_{x,y \in W} I_x^2 & \sum_{x,y \in W} I_x I_y \\ \sum_{x,y \in W} I_x I_y & \sum_{x,y \in W} I_y^2 \end{pmatrix}$$

- Ο **M** μπορεί να γραφεί σε συνέλιξη με g , με σ_I (σ Integration).
Συνήθως $\sigma_D = 0.7 \sigma_I$

$$\mathbf{M} = (\sigma_D)^2 g(\sigma_I) * \begin{bmatrix} (I_x)^2 & I_x I_y \\ I_x I_y & (I_y)^2 \end{bmatrix}$$

Αν για τον υπολογισμό I_x, I_y
χρησιμοποιηθεί $g(x; \sigma_D)$, τότε
κανονικοποιούμε τον **M** (βλ. LoG
Detector, Harris – Laplace Detector)

Harris corner detection

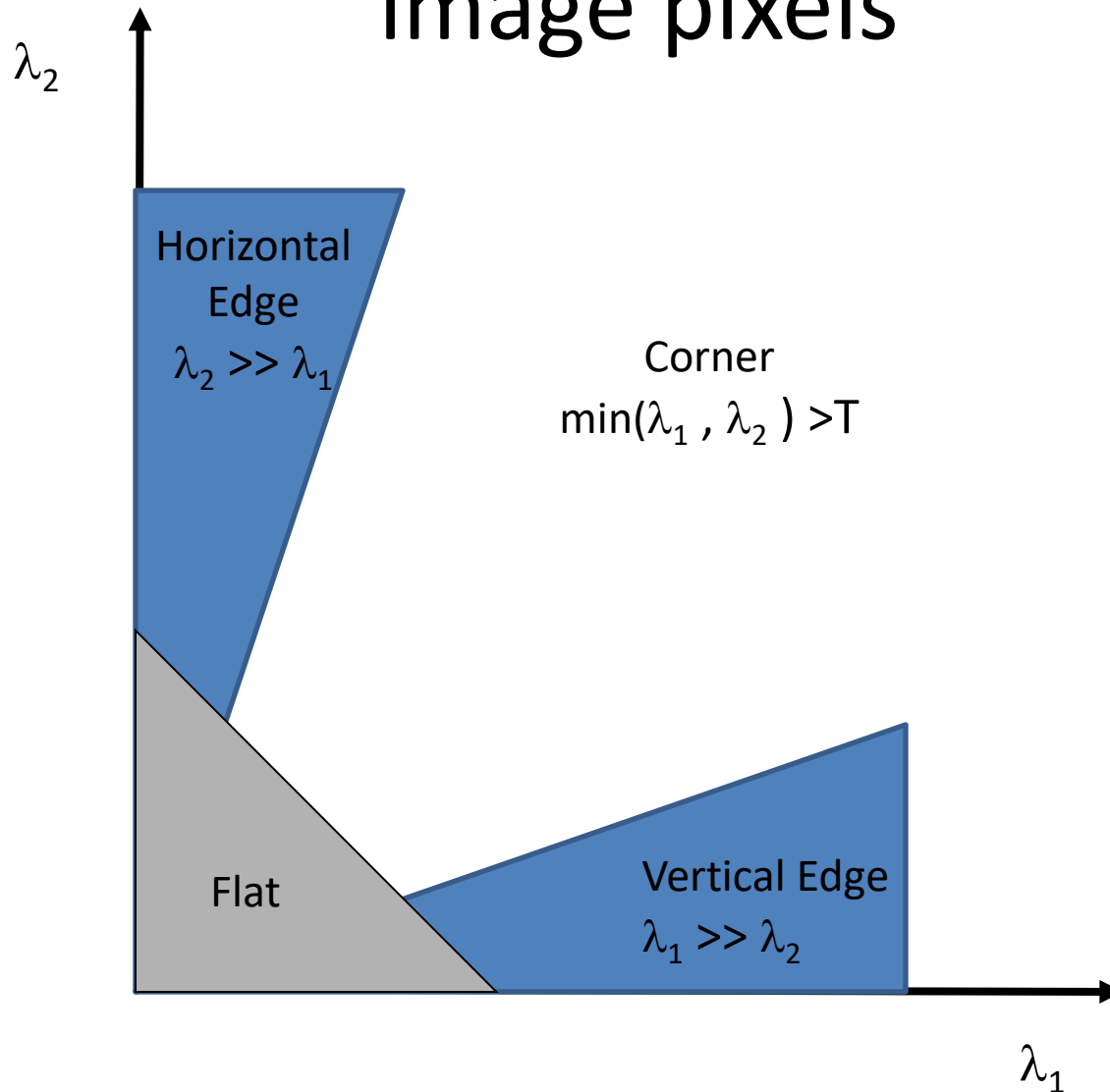
- Οι διευθύνσεις $\mathbf{v}_1, \mathbf{v}_2$ και οι αντίστοιχες τιμές δίνονται από τα ιδιοδιανύσματα και τις ιδιοτιμές (λ_1, λ_2) του πίνακα $\mathbf{M}$:

$$(\lambda_1 \cdot \lambda_2) = \det(\mathbf{M})$$

$$\lambda_1 + \lambda_2 = \text{trace}(\mathbf{M})$$

- Corner pixels are detected by thresholding the following quantities:
 - $\det(\mathbf{M}) - k(\text{trace}(\mathbf{M}))^2$, $0.04 < k < 0.06$, OR
 - $\det(\mathbf{M}) / \text{trace}(\mathbf{M})$

Eigenvalues of **M** and classification of image pixels



- Εντοπισμός γωνιών: υπολογίζεται ο $\mathbf{M}$ και εντοπίζονται τα pixels με $\min(\lambda_1, \lambda_2) > \text{κατώφλι}$
- Για την αποφυγή υπολογισμού των λ_1, λ_2 , συχνά χρησιμοποιείται το κριτήριο:

$$K = \det(\mathbf{M}) - k (\text{trace}(\mathbf{M}))^2 > \text{κατώφλι}$$

$$k \text{ in } [0.02, 0.06]$$

Παραγωγή εικόνας με κεντρικές διαφορές -Central Derivatives

- κεντρικές διαφορές: συνέλιξη της εικόνας με κατάλληλες μάσκες (FIR masks):
 - Sobel, Roberts, Prewit masks, etc.
 - 5-point and 7-point 1st deriv. order FIR masks [Farid, Simoncelli, IEEE TIP, 13 (2004)]
 - Symbolic Generation of Finite Difference Formulas described in [Keller, Pereyra, Mathematics of Computation, 1978]
 - gaussian derivatives

Παραγωγή εικόνας με χρήση παραγώγων Gaussian

- The Harris corner detection [Harris C., Stephens M., ALVEV Vision Conference, 1988] is based on approximating partial image derivatives with convolutions with gaussians, where σ_D is the standard dev. for the differentiation

$$I_x(x, y, \sigma_D) = \frac{\partial}{\partial x} g(\sigma_D) * I(x, y),$$

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

$$\frac{\partial}{\partial x} g_{\sigma_D} = -\frac{x}{\sigma_D^2} g_{\sigma_D}$$

Η Γκαουσιανή σαν φίλτρο εξομάλυνσης

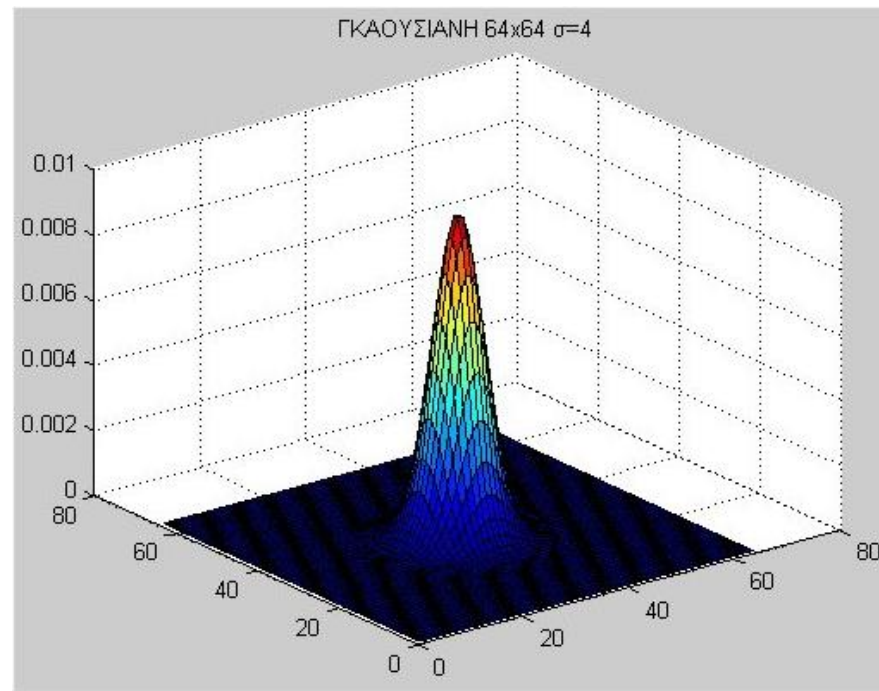
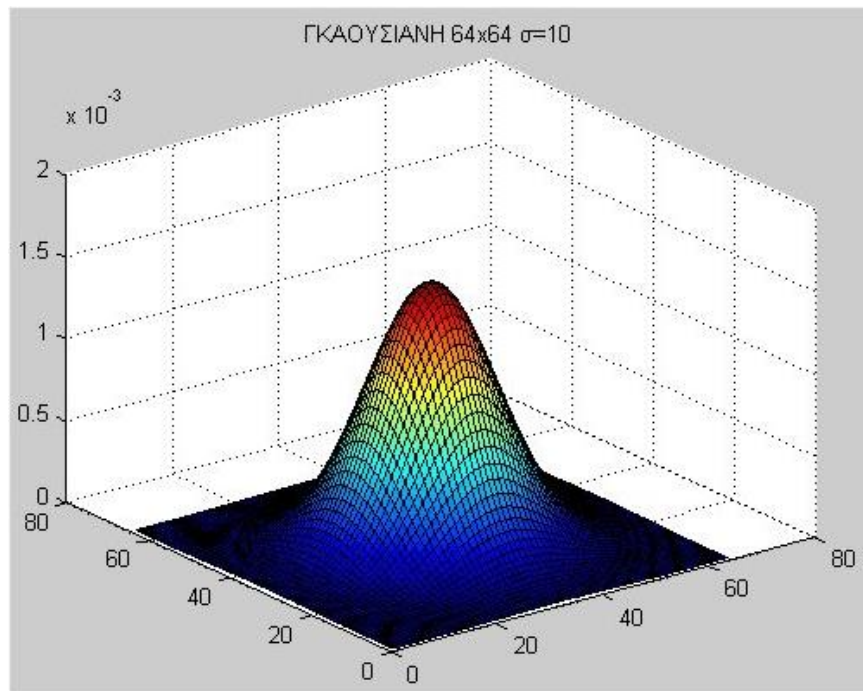
- Η γκαουσιανή επιφάνεια $g(x,y)$ ορίζεται βάσει της κανονικής κατανομής m διασπορά σ και κορυφή στο $(0,0)$.
 - Μεγαλύτερο $\sigma \rightarrow$ πιο ανοικτή και πιο κοντή επιφάνεια, έτσι ώστε ο όγκος που περικλείεται από την επιφάνεια και το επίπεδο XY να είναι ίσος με 1.

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

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

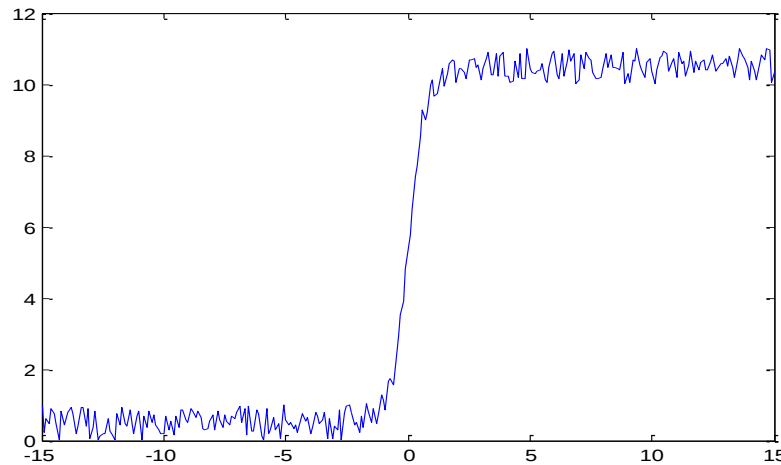
- Η 2D γκαουσιανή υπολογίζεται σε διακριτή μορφή ως εξής:
 - καθορίζεται το σ
 - Υπολογίζονται οι τιμές σε ένα πίνακα συμμετρικό γύρω από το $(0,0)$, τουλάχιστον $[-3\sigma \dots 3\sigma] \times [-3\sigma \dots 3\sigma]$, ώστε ο πίνακας να περιλαμβάνει μέχρι και πολύ χαμηλές τιμές.
 - Όσο μεγαλύτερο το σ , τόσο ισχυρότερη η εξομάλυνση από τη συνέλιξη της εικόνας με την γκαουσιανή.

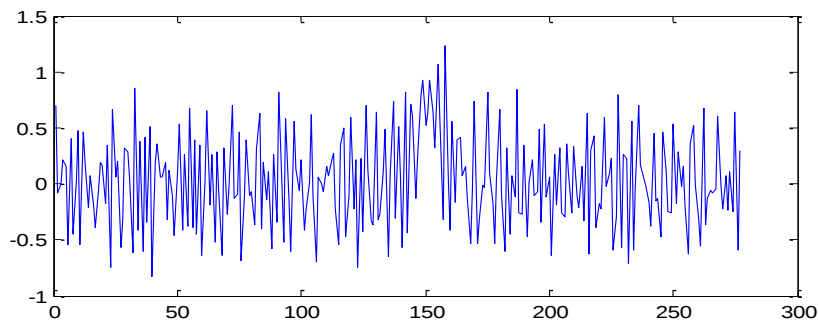
Η Γκαουσιανή σαν φίλτρο εξομάλυνσης



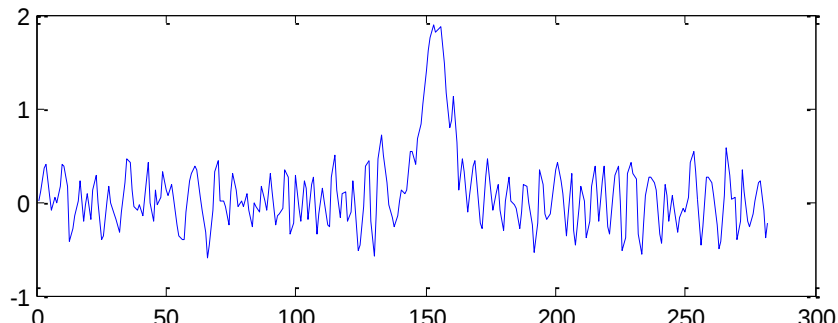
Παράγωγιση με γκαουσιανή: επιλογή του σ_D

- Εστω 1D ακμή με μη συσχετισμένο προσθετικό λευκό θόρυβο.
- Εφαρμόζουμε την συνέλιξη με τις μονοδιάστατες μάσκες Sobel και τις μονοδιάστατες γκαουσιανές παραγώγους, με διάφορο πλήθος στοιχείων.

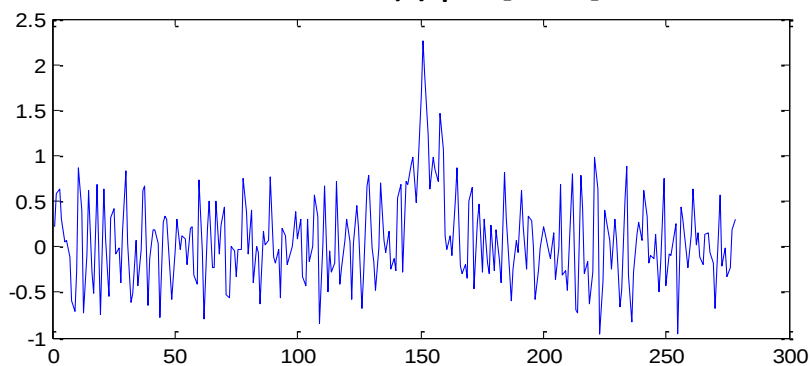




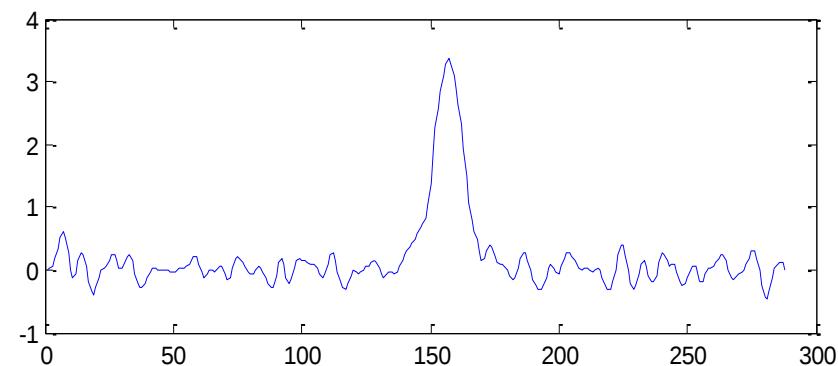
Συνέλιξη με [1,-1]



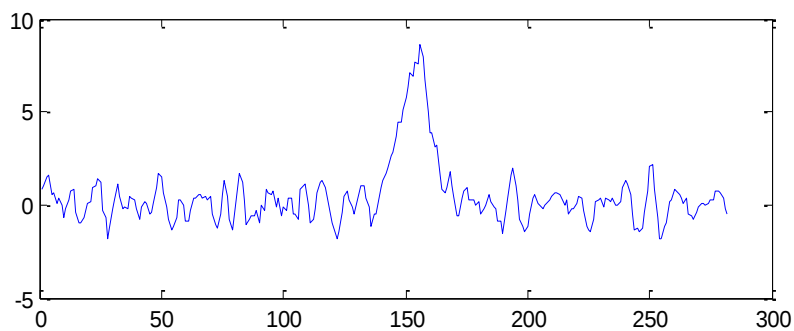
Συνέλιξη με 1^η παράγωγο γκαουσιανής, $\sigma_D=1$



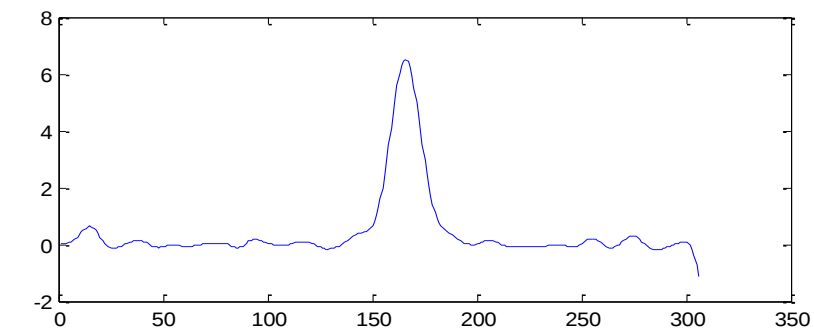
Συνέλιξη με [1,0,-1]



Συνέλιξη με 1^η παράγωγο γκαουσιανής, $\sigma_D=2$



Συνέλιξη με [1,1,1,1,0,-1,-1,-1,-1]



Συνέλιξη με 1^η παράγωγο γκαουσιανής, $\sigma_D=5$

Παράγωγοι της γκαουσιανής

- Η παραγωγή της γκαουσιανής $g^{(n)}(x)$ παράγει συναρτήσεις που είναι γινόμενο της ίδιας γκαουσιανής με πολυώνυμα που ονομάζονται Hermite

$$g^{(n)}(x) = (-1)^n \left(\frac{1}{\sigma\sqrt{2\pi}} \right)^n H_n \left(\frac{x}{\sigma\sqrt{2\pi}} \right) g(x)$$

1

2 x

-2 + 4 x²

-12 x + 8 x³

12 - 48 x² + 16 x⁴

120 x - 160 x³ + 32 x⁵

-120 + 720 x² - 480 x⁴ + 64 x⁶

-1680 x + 3360 x³ - 1344 x⁵ + 128 x⁷

Κατασκευή FIR-derivative masks

- Definitions:

$$O_n = \begin{bmatrix} 1 & 2 & \cdots & n \\ 1 & 2^3 & \cdots & n^3 \\ \vdots & & & \\ 1 & 2^{2n-1} & & n^{2n-1} \end{bmatrix} \quad E_n = \begin{bmatrix} 1 & 2^2 & \cdots & n^2 \\ 1 & 2^4 & \cdots & n^4 \\ \vdots & \vdots & & \\ 1 & 2^{2n} & & n^{2n} \end{bmatrix}$$

- $DO_{i,k}$ and $DE_{i,k}$ is the determinant of order $n-1$, which follows by deleting the i^{th} row, by replacing the k^{th} column with the n^{th} column and then deleting the n^{th} column of O_n and E_n respectively.
- $DO_{i,n}$ and $DE_{i,n}$ is the negative of the determinant of order $n-1$, obtained by deleting the i^{th} row and n^{th} column of O_n and E_n respectively

- **Theorem 1.** Given a real sequence $f(x)$, its n -point derivative $f^{(2i-1)}(x)$ of any odd order $(2i-1)$ with $i=1, \dots, n-1$, is given by

$$f^{(2i-1)}(x) = \frac{(2i-1)!}{2h^{2i-1} \sum_{k=1}^n k^{2i-1} DO_{i,k}} \sum_{k=1}^n DO_{i,k} (f(x+kh) - f(x-kh)) + O(h^{2n+1})$$

- **Theorem 2.** Given a real sequence $f(x)$, its n -point derivative $f^{(2i)}(x)$ of any even order $(2i)$ with $i=1, \dots, n-1$, is given by:

$$f^{(2i)}(x) = \frac{(2i)!}{2h^{2i} \sum_{k=1}^n k^{2i} DE_{i,k}} \left[\frac{1}{2} \sum_{k=1}^n DE_{i,k} (f(x+kh) - f(x-kh)) + f(x) \sum_{k=1}^n DE_{i,k} \right] + O(h^{2n+1})$$

- Assuming that $f(x)$ is uniformly sampled ($h=1$), previous equations can be written as linear convolutions

$$f^{(2i-1)}(x) = (f * M_{odd})(x)$$

$$f^{(2i)}(x) = (f * M_{even})(x)$$

- where

$$M_{even}(k) = \frac{(2i)!DE_{i,k}}{2\sum_{k=1}^n k^{2i} DE_{i,k}} \left[-1, -1, \dots, -1, \frac{1}{2}, 1, \dots, 1, 1 \right]$$

$$M_{odd}(k) = \frac{(2i-1)!DO_{i,k}}{2\sum_{k=1}^n k^{2i-1} DO_{i,k}} [-1, -1, \dots, -1, 0, 1, \dots, 1, 1]$$

deriv. order	Num. of points in Mask	Mask
1	3	$[1/2, 0, -1/2]$
1	5	$[-1, 8, 0, -8, 1]/12$
1	7	$[1, -9, 45, 0, -45, 9, -1]/60$
1	9	$[-1/280, 4/105, -1/4, 0, 1/4, -4/105, 1/280]$
1	11	$[4, -5, +30, -120, +430, 0, -430, +120, -30, 5, -4]/504$
2	3	$[1, -2, 1]$
2	5	$[-1/12, 4/3, -2.5, 4/3, -1/12]$
2	7	$[1/90, -3/20, 1.5, -49/18, 1.5, -3/20, 1/90]$
2	9	$[-1/560, 8/315, -1/5, 8/5, 205/144, 8/5, -1/5, 8/315, -1/560]$

The resulting masks M_{even} and M_{odd} for 1st and 2nd order differentiation, of length up to 11 points

Examples: Sobel and Laplacian masks

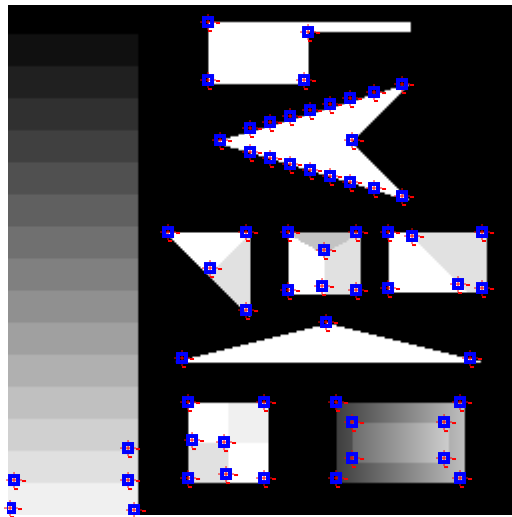
- The 3x3 Sobel operator can be generated by linear convolution, using the first mask of prev. Table

$$\begin{bmatrix} 1/2 & 0 & -1/2 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}^T = \begin{bmatrix} 0.5 & 0 & -0.5 \\ 1 & 0 & -1 \\ 0.5 & 0 & -0.5 \end{bmatrix}$$

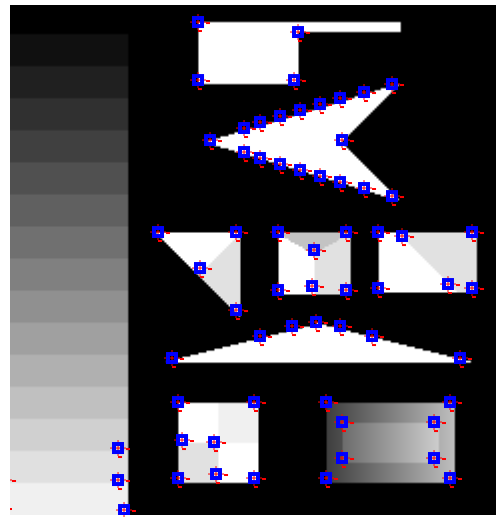
- The well-known Laplacian operator can be generated as following

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

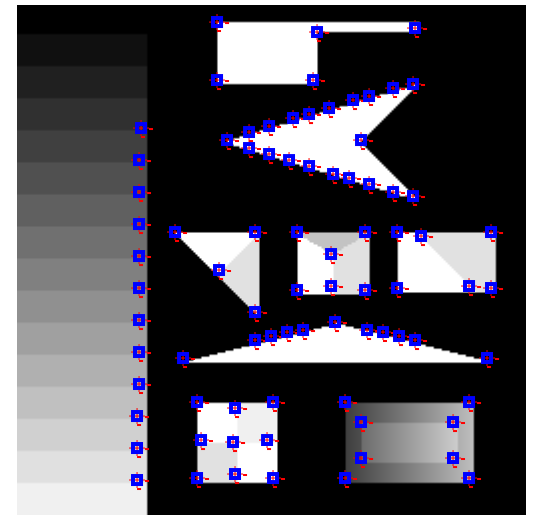
Corner Harris detection with different derivative approximations



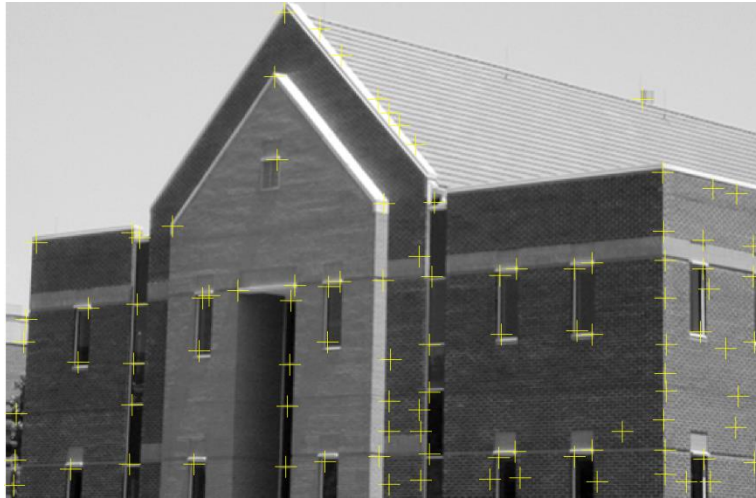
gaussian



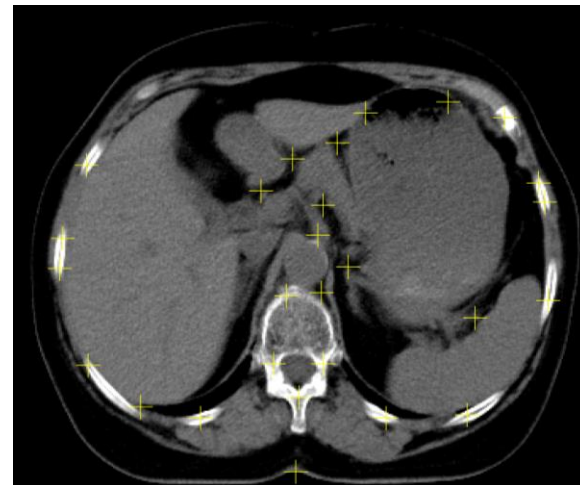
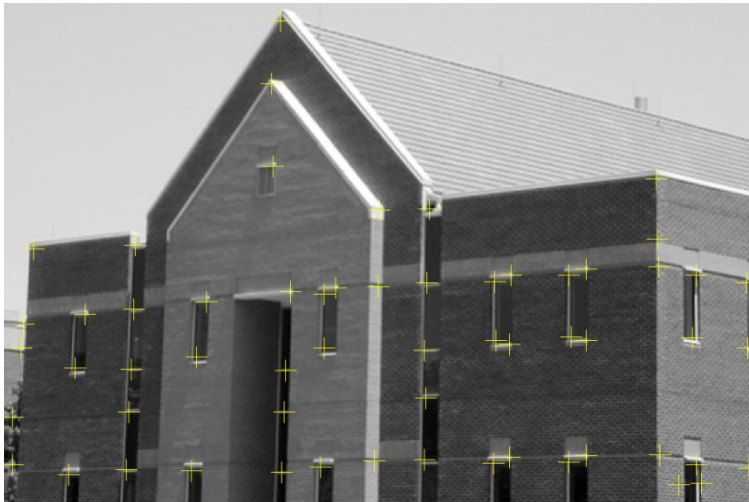
FIR 7p



IIR 10,8



IIR derivative mask with order of accuracy=10

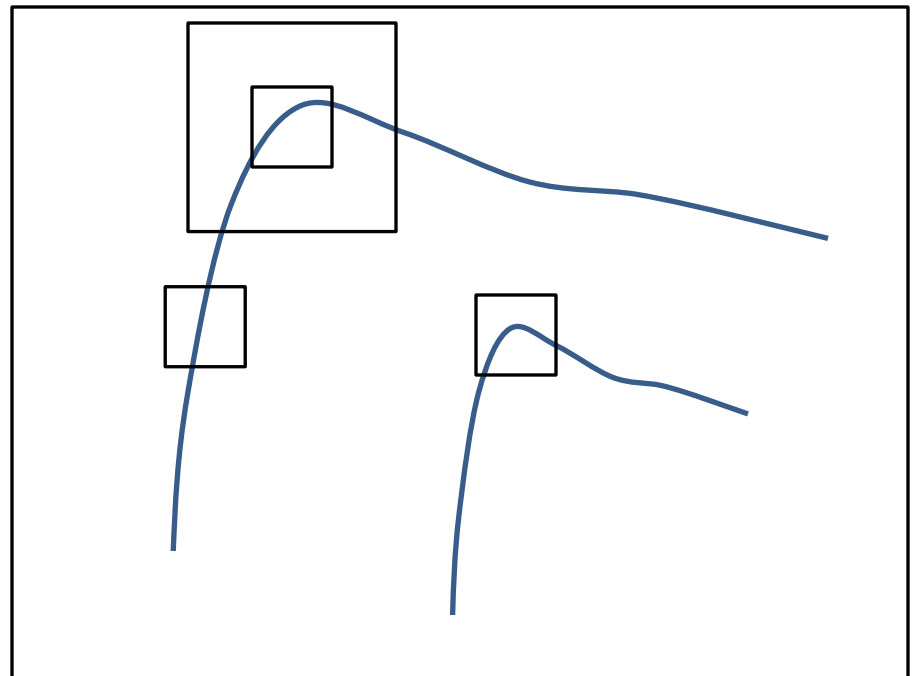


the 7-point FIR derivative mask generated by the proposed algorithm

Αμεταβλητότητα: Invariance

- Η ανίχνευση γωνιών με τον πίνακα **M** είναι ανεξάρτητη από μετατόπιση, περιστροφή και αλλαγές φωτεινότητας.
- Εξαρτάται όμως από την αλλαγή κλίμακας

- Το ίδιο σχήμα εκλαμβάνεται ως γωνία, ή όχι, ανάλογα με την κλίμακα του παραθύρου
- Ισοδύναμα, παράθυρο w ίδιου μεγέθους δίνει διαφορετικά αποτελέσματα, όταν η κλίμακα της εικόνας αλλάζει



Ανίχνευση της κλίμακας της εικόνας (Image scale space)

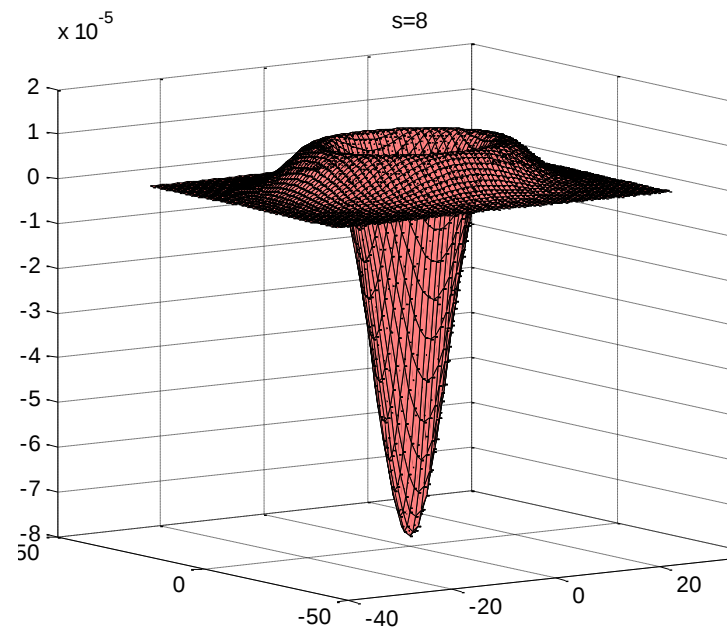
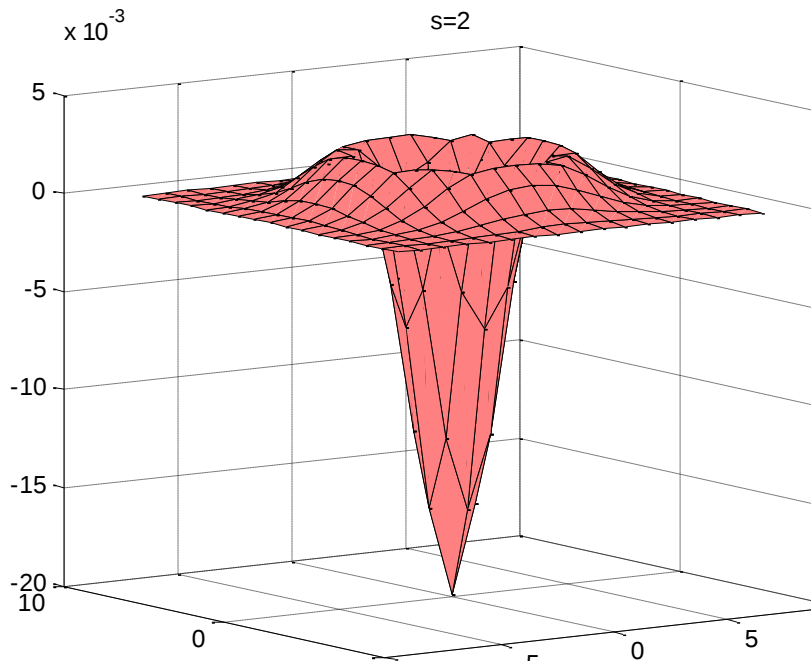
Laplacian of Gaussian LoG Detector

- Ορισμός της Laplacian of Gaussian LoG

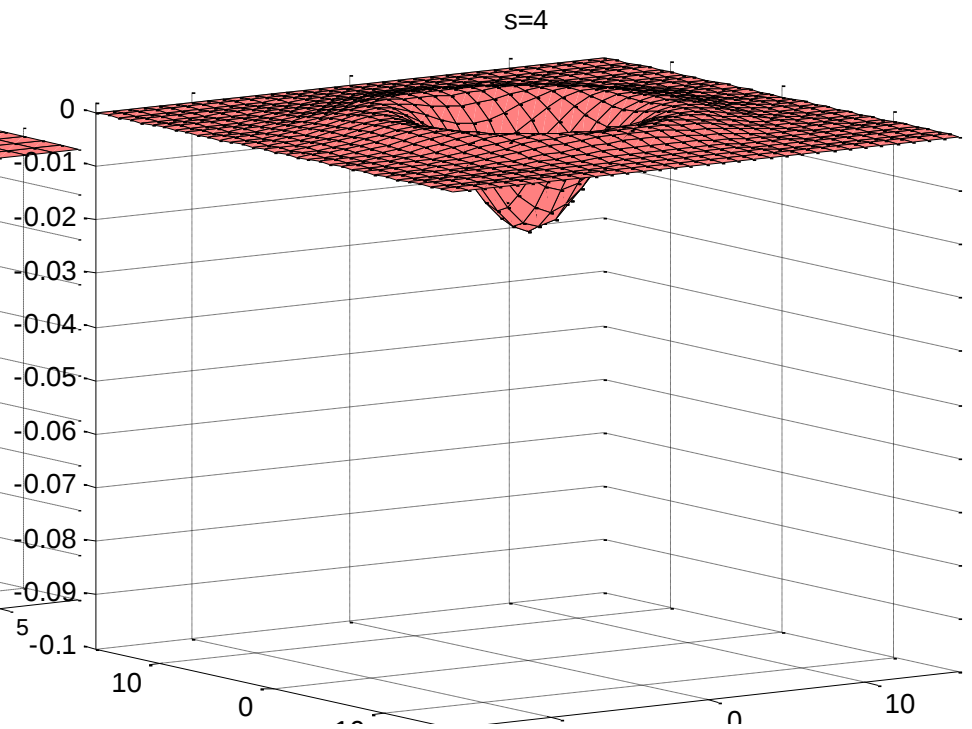
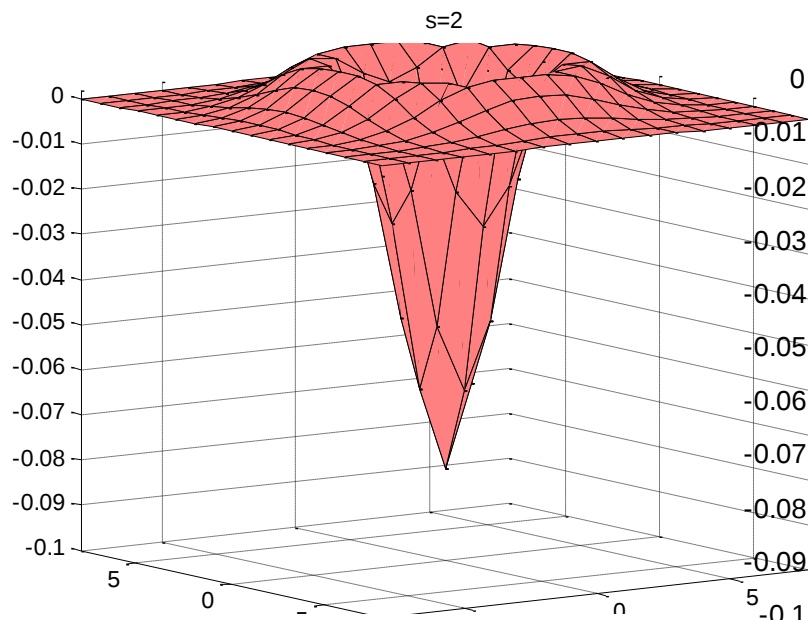


- Scale-normalized LoG: $L(x;\sigma)=\sigma^2(g_{xx} + g_{yy})$

- LoG για $\sigma=2$ και $\sigma=8$ (μη κανονικοποιημένα γραφήματα με διαφορετικές κλίμακες)



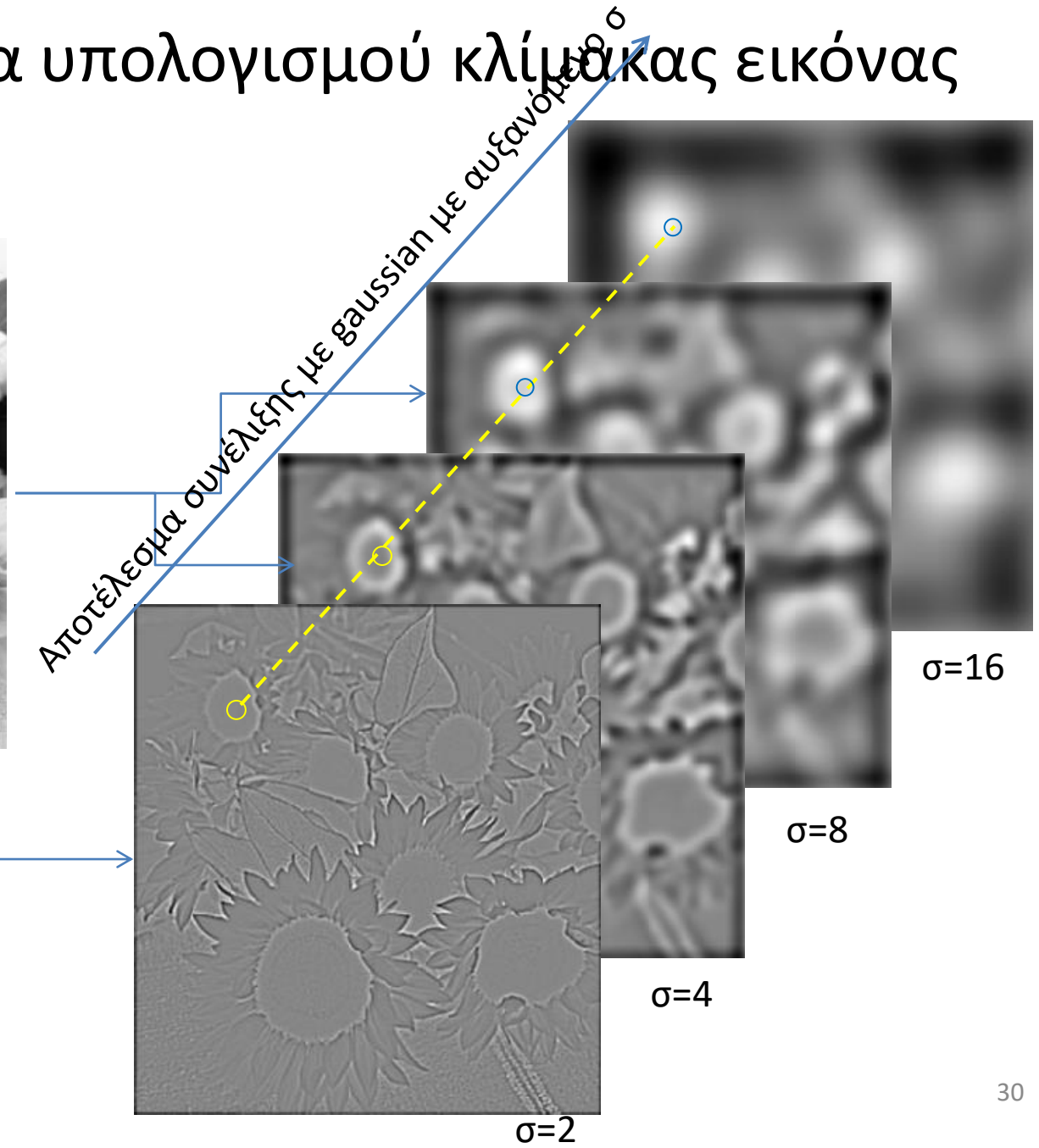
- Η Log έχει <0 τιμές στο κέντρο, >0 τιμές περιφερειακά και $\rightarrow 0$ όταν $|x|, |y| \rightarrow \infty$
- Τα (x,y) : $\text{Log}(x,y)=0$, είναι σημεία κύκλου **με ακτίνα ανάλογη του σ** .
- Έτσι, η συνέλιξη της Log με μία εικόνα παράγει: υψηλές απόλυτες τιμές σε θέσεις που υπάρχουν κύκλοι, όταν η ακτίνα του κύκλου είναι $r=\sigma/\sqrt{2}$
- Για να είναι συγκρίσιμες οι αποκρίσεις LoG (συνέλιξη της LoG με την εικόνα) πρέπει να κανονικοποιηθούν ως προς το σ :
Scale-normalized LoG: $\text{LoG}_\sigma = \sigma^2(g_{xx} + g_{yy})$

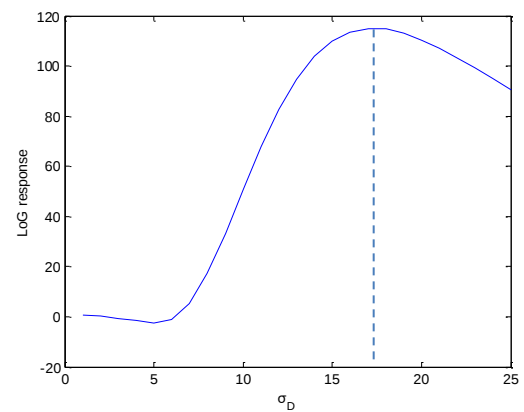
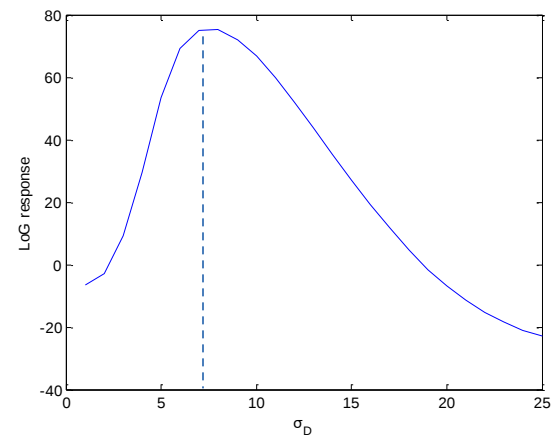


Αλγόριθμος υπολογισμού Image scale

- Επιλέγονται τα όρια μεταβολής του σ , σ_1 , σ_2 ,
- Για όλα τα σ από σ_1 έως σ_2 ,
Υπολογίζεται η συνέλιξη της αρχικής I_0 με την κανονικοποιημένη $\sigma^2 \cdot LoG_\sigma$: $I_\sigma = I_0 * \sigma^2 \cdot LoG_\sigma$,
- end
- For each pixel (i,j)
 - $m \leftarrow \max(I_\sigma(i,j))$
 - Επιλέγεται το $\sigma_{\max}$ με τη μέγιστη απόκριση
 $\sigma_{\max} \leftarrow \operatorname{argmax}(I_\sigma(i,j))$
 - If m τοπικό μέγιστο σε περιοχή με ακτίνα R , τότε $\sigma_{\max}$ είναι ανάλογο της κλίμακας της εικόνας στο (i,j)

Παράδειγμα υπολογισμού κλίμακας εικόνας





- Υπολογισμός $\sigma_{\max}$ για 2 διαφορετικά pixel.

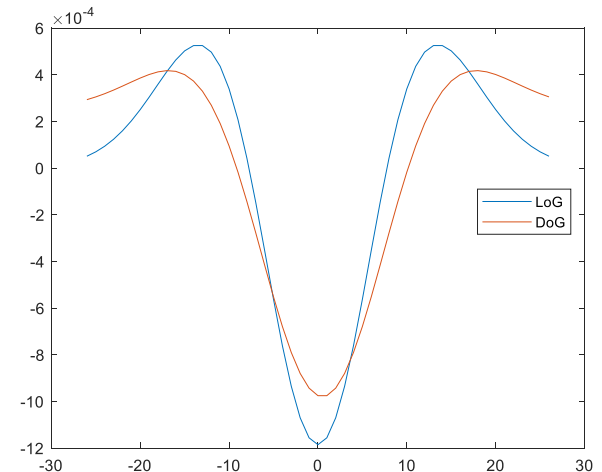
Προσέγγιση της LoG με διαφορά Gaussians (DoG)

- Η συνέλιξη 2 Gaussians με σ_1, σ_2 παράγει Gaussian με $\sigma^2 = \sigma_1^2 + \sigma_2^2$,
— [<http://www.tina-vision.net/docs/memos/2003-003.pdf>, page 6]

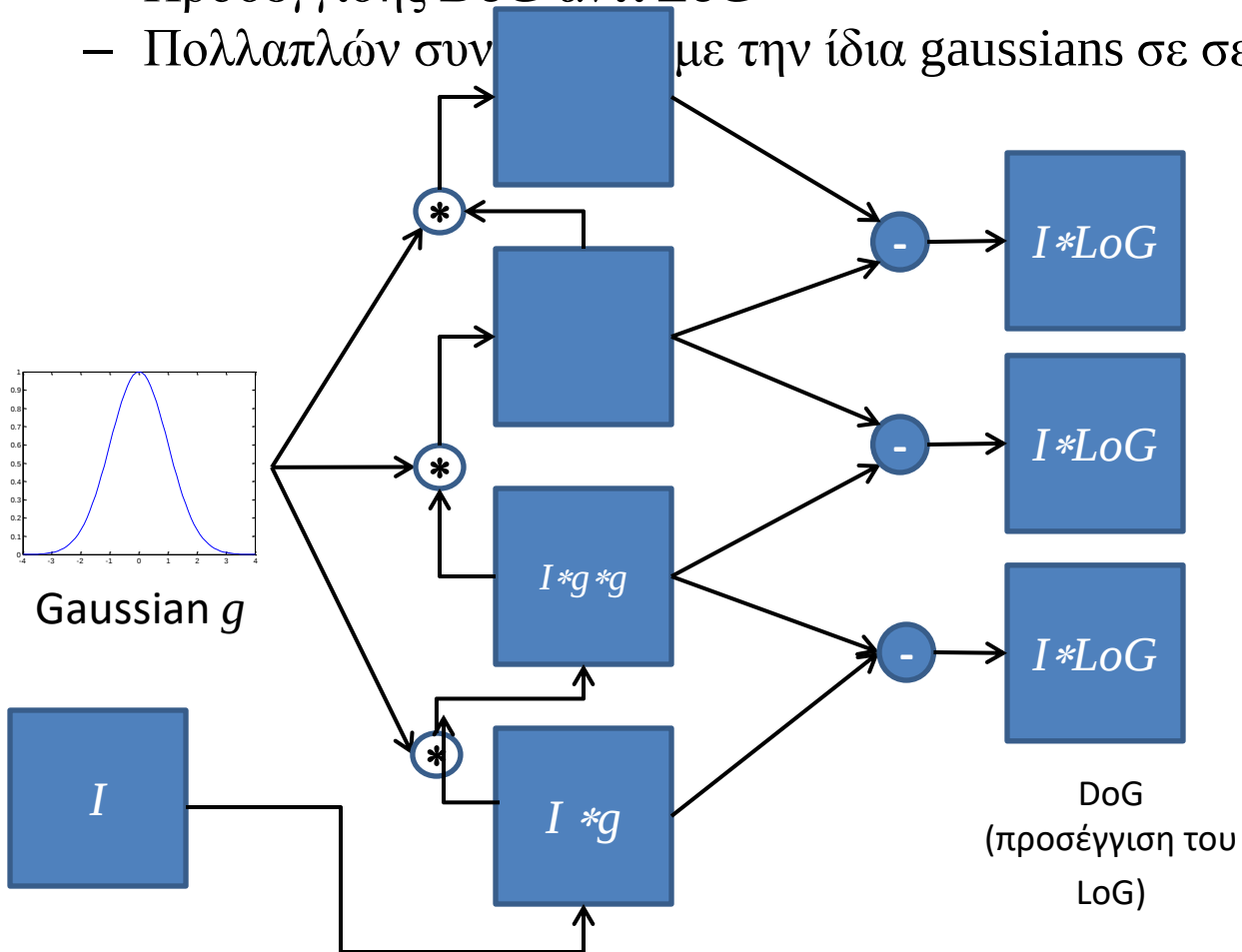
$$\left. \begin{aligned} g_1(x) &= \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_1^2}} \leftrightarrow G_1(\omega) = e^{-\frac{\sigma_1^2 \omega^2}{2}} \\ g_2(x) &= \frac{1}{\sigma_2 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_2^2}} \leftrightarrow G_2(\omega) = e^{-\frac{\sigma_2^2 \omega^2}{2}} \end{aligned} \right\} g_1 * g_2 \leftrightarrow G_1 G_2 = e^{-\frac{(\sigma_1^2 + \sigma_2^2) \omega^2}{2}}$$

- Η LoG προσεγγίζεται από τη διαφορά 2 Gaussians με $\sigma_1, k\sigma_1$,

$$g(x; k\sigma) - g(x; \sigma) \approx (k - 1) \sigma^2 LoG$$



- Αποτελεσματική υλοποίηση του LoG με χρήση:
 - Προσέγγισης DoG αντί LoG
 - Πολλαπλών συνυπολογισμών με την ίδια gaussians σε σειρά (cascade conv).



Harris-Laplace corner detection

- Ο Harris detector
 - είναι αμετάβλητος ως προς αλλαγή φωτεινότητας και περιστροφή,
 - Εξαρτάται από την κλίμακα (Image scale)
- Harris-Laplace: συνδιάζει το Harris corner detection με LoG image scale detection, ώστε να είναι:
 - αμετάβλητος ως προς αλλαγή κλίμακας - (Image scale invariant) [K. Mikolajczyk, C. Schmid, IJCV 1(60), 2004, ICCV 2001].

Αλγόριθμος Harris - Laplace

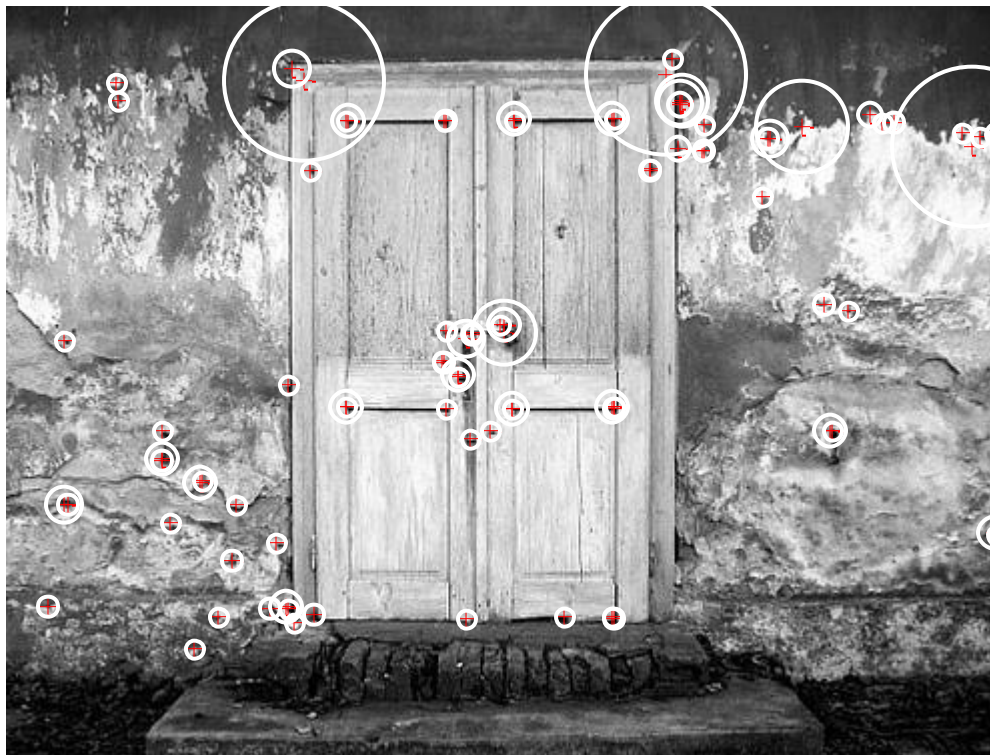
- Η απόκριση του Harris K_n υπολογίζεται για διάφορες τιμές του σ . Συνήθως $\sigma_I = s^n \sigma_0$, $n=1,2,\dots,N$, $0.7 < s < 1.4$, $\sigma_D = 0.7 \sigma_I$.
- Για κάθε Pixel της εικόνας:
 - Κατωφλιώνονται τα K_n και
 - Εντοπίζονται τα pixel με K_n τοπικό μέγιστο **σε μία περιοχή της εικόνας** με ακτίνα R pixels (συνήθως $R=1$)
- Για κάθε Pixel του προηγούμενου βήματος:
 - Υπολογίζεται η συνέλιξη της I_0 με τη LoG για διαφορετικά $\sigma = s^n \sigma_0$: $I_\sigma = I_0 * \sigma^2 \cdot LoG_\sigma$
- Αν το I_σ παρουσιάζει τοπικό μέγιστο **για τις τιμές του σ** , τότε το τρέχον pixel θεωρείται pixel γωνίας.

Παραλλαγή του Αλγορίθμου Harris - Laplace

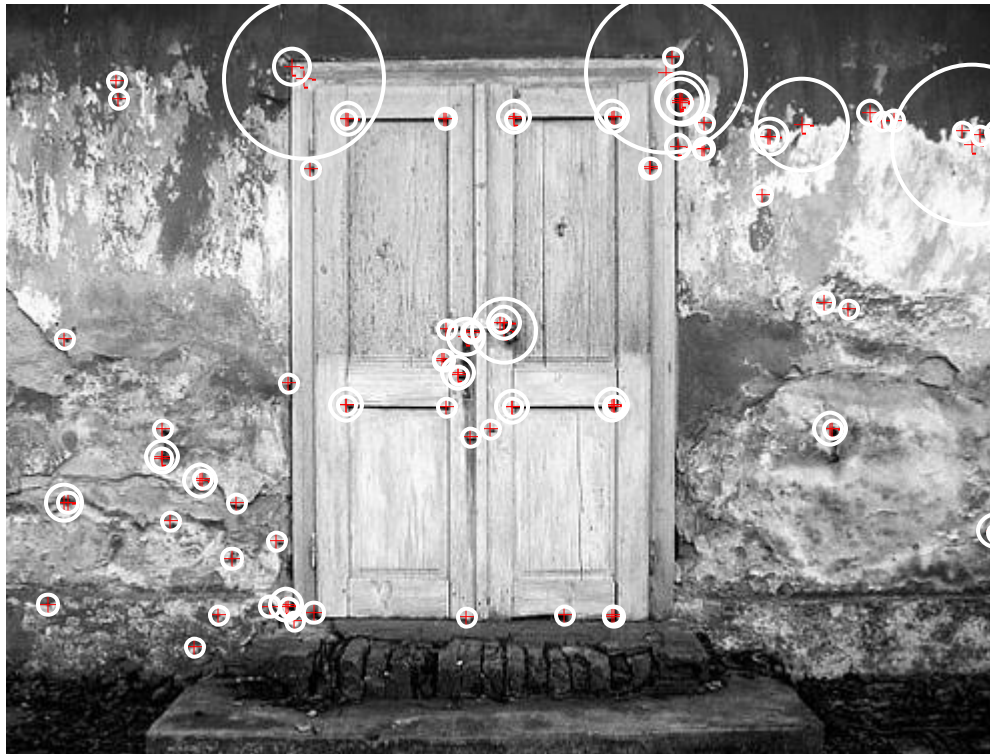
- Μία παραλλαγή του Harris – Laplace χρησιμοποιεί αυστηρότερο κριτήριο για την επιλογή των γωνιών:
 - Το τοπικό μέγιστο στην κλίμακα της εικόνας για ένα σημείο πρέπει να εμφανίζεται στο ίδιο σ , στην απόκριση του Harris και στην απόκριση της LoG.

Example of the application of Harris-Laplace

- The points of interest that have been identified are illustrated in the figure, along with their estimated scale (radius of circle = 3σ , σ the standard dev. of the LoG mask size that produced maximum response).



Derivative of Gaussian



Harris-Laplace feature points with the proposed FIR 7p



IIR derivatives, order of accuracy 10,8

Εσσιανός πίνακας

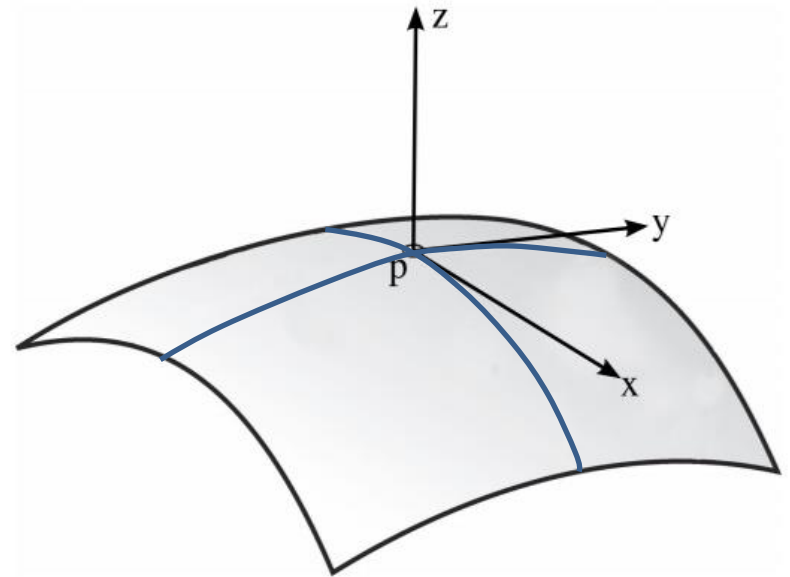
- Η μέθοδος βασίζεται στην θεώρηση της εικόνας ως επιφάνεια και στις ιδιότητες της διαφορικής γεωμετρίας
- Βασικά στοιχεία διαφορικής γεωμετρίας
- Εστω επιφάνεια $S(u,v)$. Σε κάθε σημείο της ορίζεται το κάθετο διάνυσμα, 2 κάθετα μεταξύ τους εφαπτόμενα διανύσματα και 2 καμπυλότητες:

Οι 2 καμπυλότητες:

- Κύριες κ_1, κ_2 , ή ισοδύναμα:
- Gaussian και μέση καμπυλότητα

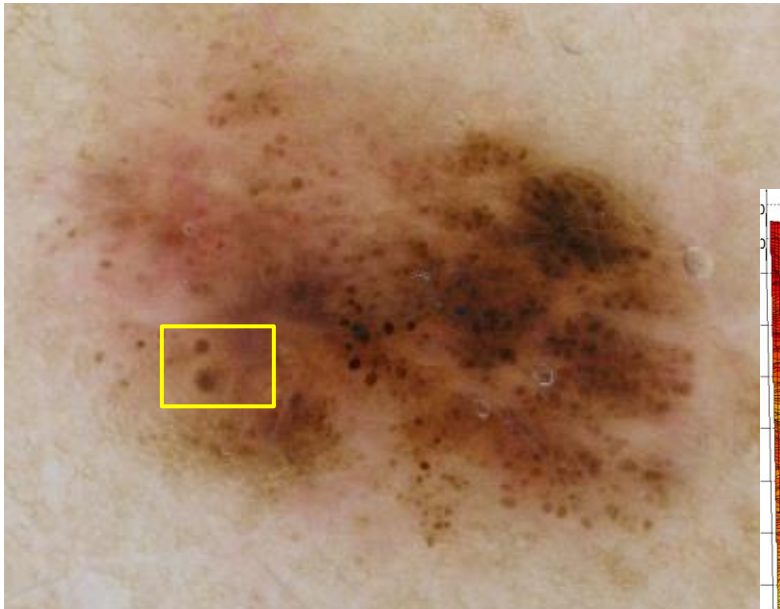
Οι κατεύθυνσεις και οι τιμές των κ_1, κ_2 , υπολογίζονται από τα ιδιοδιαδύσματα και τις ιδιοτιμές του Εσσιανού πίνακα

$$\mathbf{H}(u,v) = \begin{pmatrix} S_{uu} & S_{uv} \\ S_{uv} & S_{vv} \end{pmatrix}_{(u,v)}$$

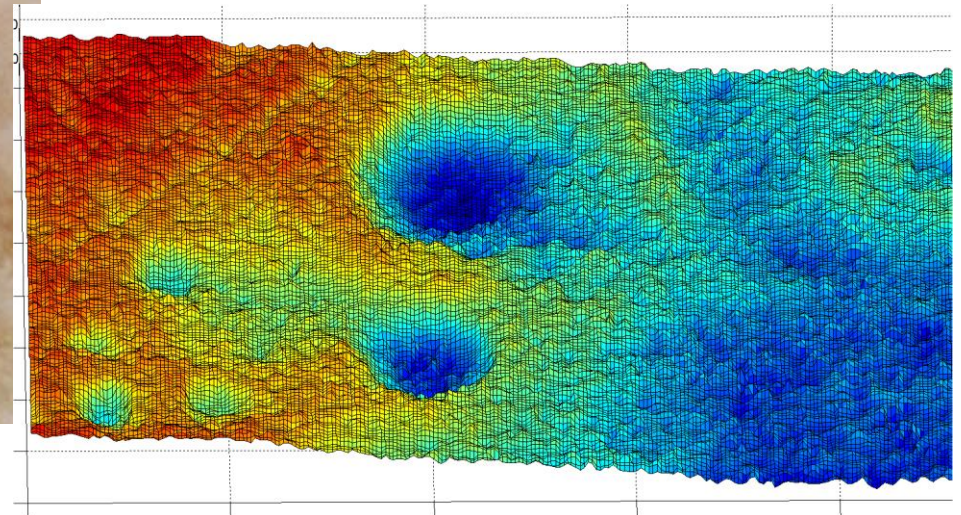
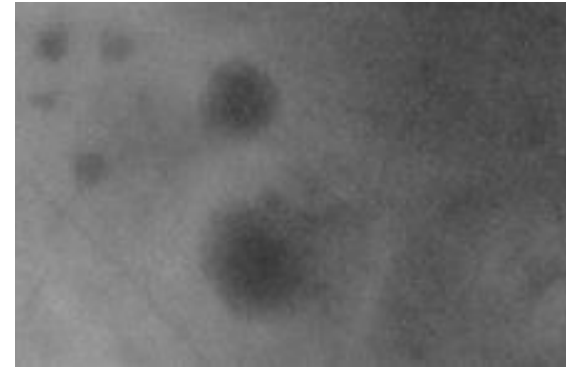


Παράδειγμα: image blobs

- Τα κεντρικά pixels των image blobs έχουν σχεδόν ίσες κύριες καμπυλότητες



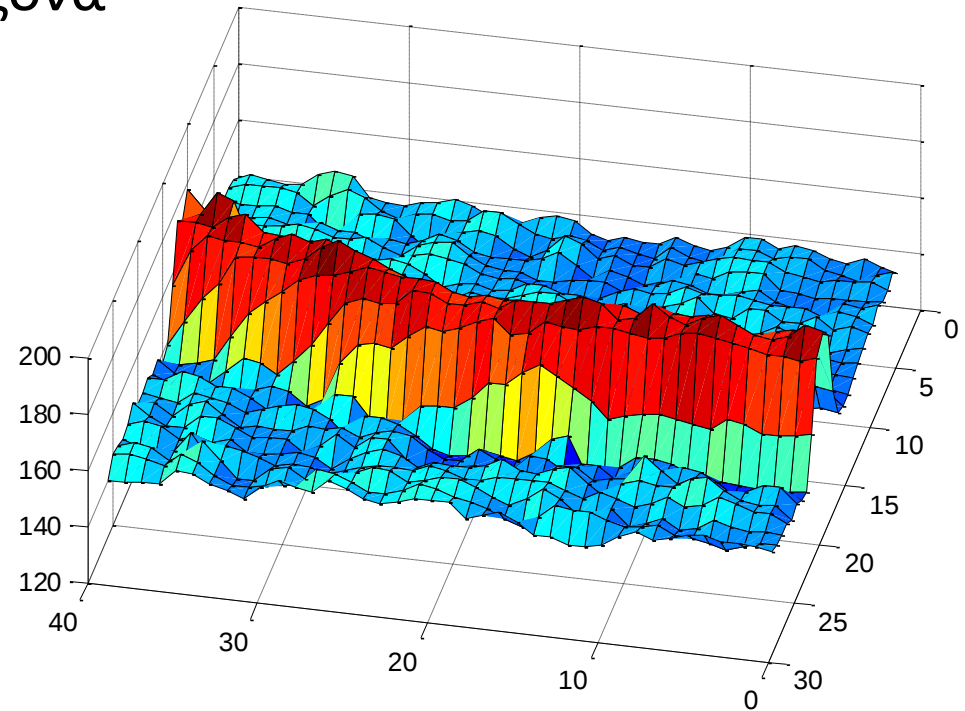
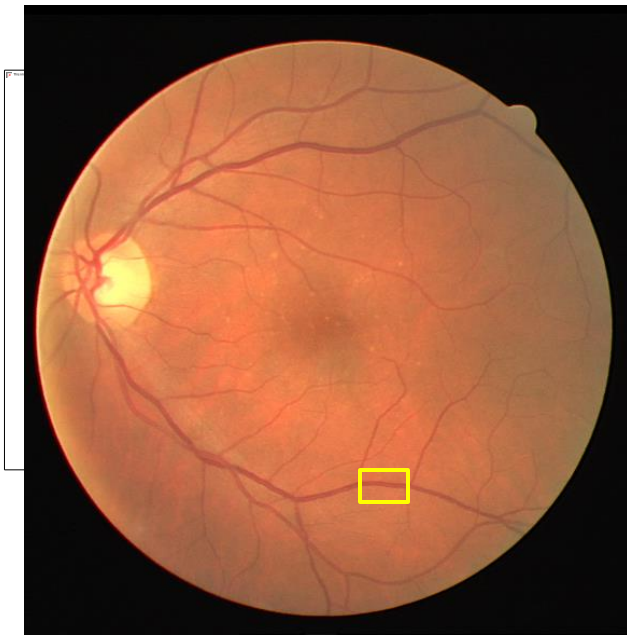
Εικόνα δερμοσκοπίας



Η εικόνα σαν επιφάνεια

Εντοπισμός Αγγείων

- Τα pixels του κεντρικού άξονα αγγείων έχουν πολύ μικρή κύρια καμπυλότητα κ_1 παράλληλα στον άξονα και πολύ μεγάλη κ_2 κάθετα στον άξονα



Τμήμα εικόνας με αγγείο

Τμηματοποίηση αγγείων

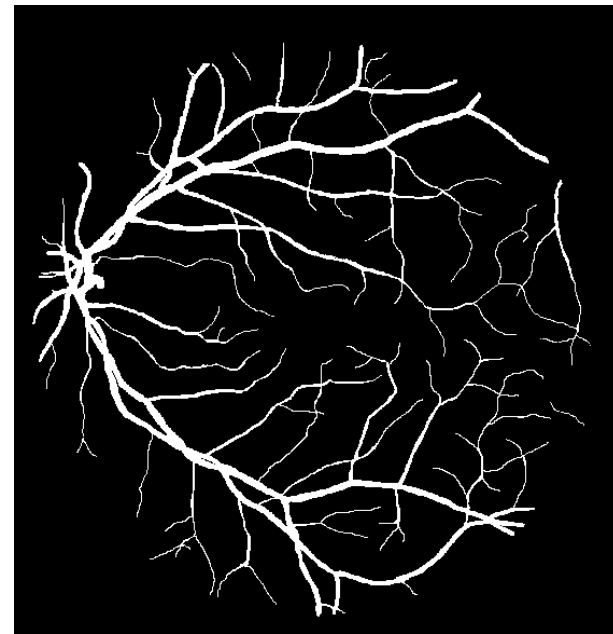
Vessel segmentation

- Αγγεία απεικονίζονται σε πολλά είδη εξετάσεων, σε 2Δ και 3Δ.

<http://www.isi.uu.nl/Research/Databases/DRIVE/results.php>



Εγχρωμη εικόνα αμφιβληστροειδούς



Αγγεία τμηματοποιημένα από ειδικό

Τμηματοποίηση αγγείων βάσει του Hessian πίνακα της εικόνας

- Για κάθε pixel (x,y) της εικόνας I υπολογίζεται ο Hessian πίνακας που παρέχει πληροφορία για την τοπική δομή (σχήμα) της I .

$$\mathbf{H}(x, y) = \begin{pmatrix} I_{xx}(x, y) & I_{xy}(x, y) \\ I_{xy}(x, y) & I_{yy}(x, y) \end{pmatrix}$$

- Οι δείκτες συμβολίζουν παραγωγή. Ο υπολογισμός των παραγώγων γίνεται με συνέλιξη με την παράγωγο μίας γκαουσιανής με σ της επιλογής του χρήστη

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

- Η παράμετρος σ χρησιμοποιείται για να επιλεγεί ή διάμετρος του αγγείου που θα τμηματοποιηθεί

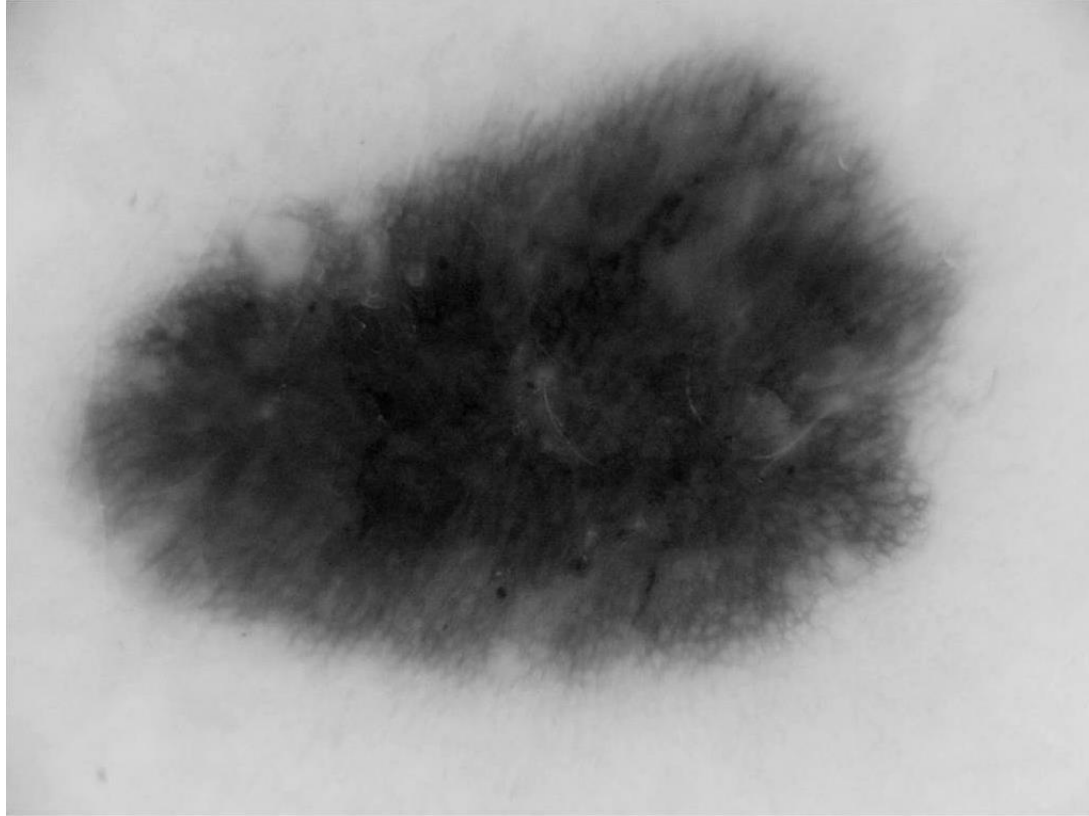
- Η τιμή της παραγώγου κανονικοποιείται με χρήση του παράγοντα σ^2 , ώστε οι αποκρίσεις της συνάλιξης να είναι συγκρίσιμες για διαφορετικά σ .

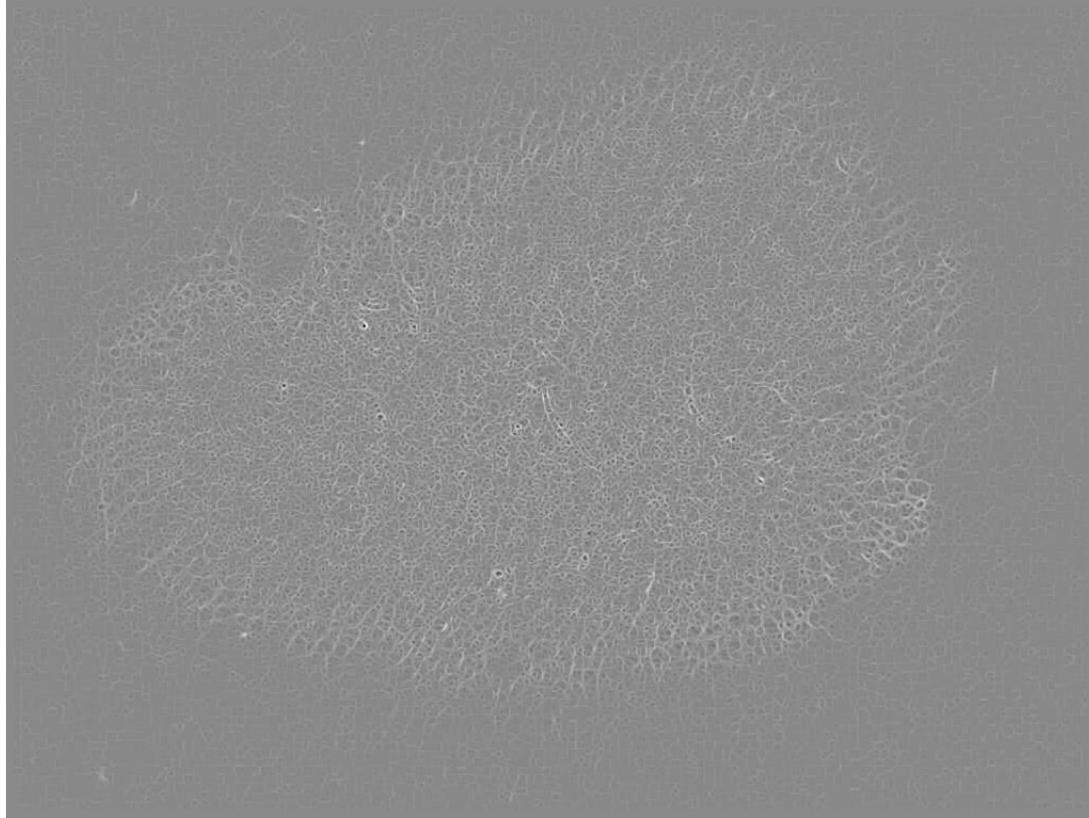
$$I_{xx}(x, y; \sigma) = \sigma^{2\gamma} I(x, y) * \frac{\partial^2 g(x, y; \sigma)}{\partial x^2}$$

- Συνήθως $\gamma=1$.
- Ο Hessian $\mathbf{H}$ της εικόνας I προσεγγίζεται ως εξής:.

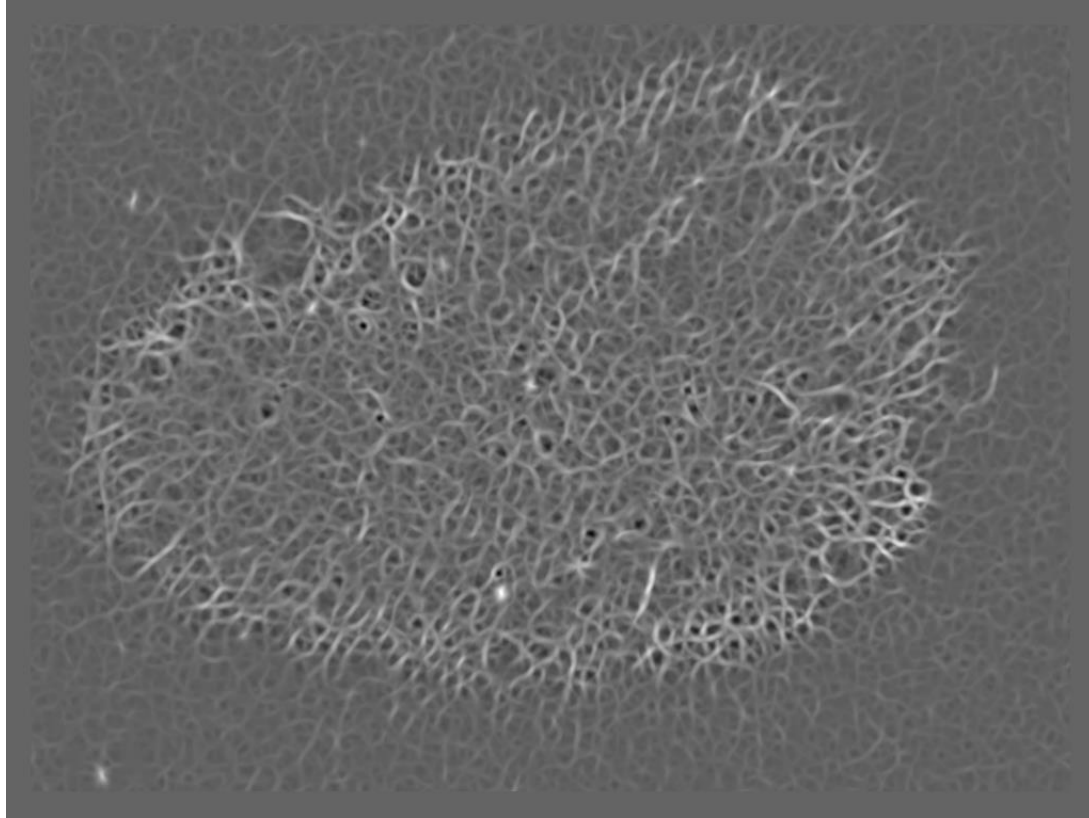
$$\mathbf{H}(x, y; \sigma) = \sigma^2 \begin{pmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{pmatrix} * I(x, y)$$

- Εστω λ_1, λ_2 οι πραγματικές ιδιοτιμές (eigenvalues) του $\mathbf{H}$ σε κάθε pixel (x,y) για συγκεκριμένο σ .
- $\lambda_{max} = \max(\lambda_1, \lambda_2) \gg \lambda_{min} = \min(\lambda_1, \lambda_2) \rightarrow 0$
 - D : ορίζουσα, Tr : ίχνος του $\mathbf{H}$.
- $|\lambda_{max}| \gg \lambda_{min} \rightarrow 0$ ΚΑΙ $\lambda_{max} > 0$ ΤΟΤΕ το (x,y) ανήκει σε **σκούρο** αγγείο
- $|\lambda_{max}| \gg \lambda_{min} \rightarrow 0$ ΚΑΙ $\lambda_{max} < 0$ ΤΟΤΕ το (x,y) ανήκει σε **φωτεινό** αγγείο
- Το ιδιοδιάνυσμα με λ_{max} , καθορίζει την κατεύθυνση με τη maximum καμπυλότητα

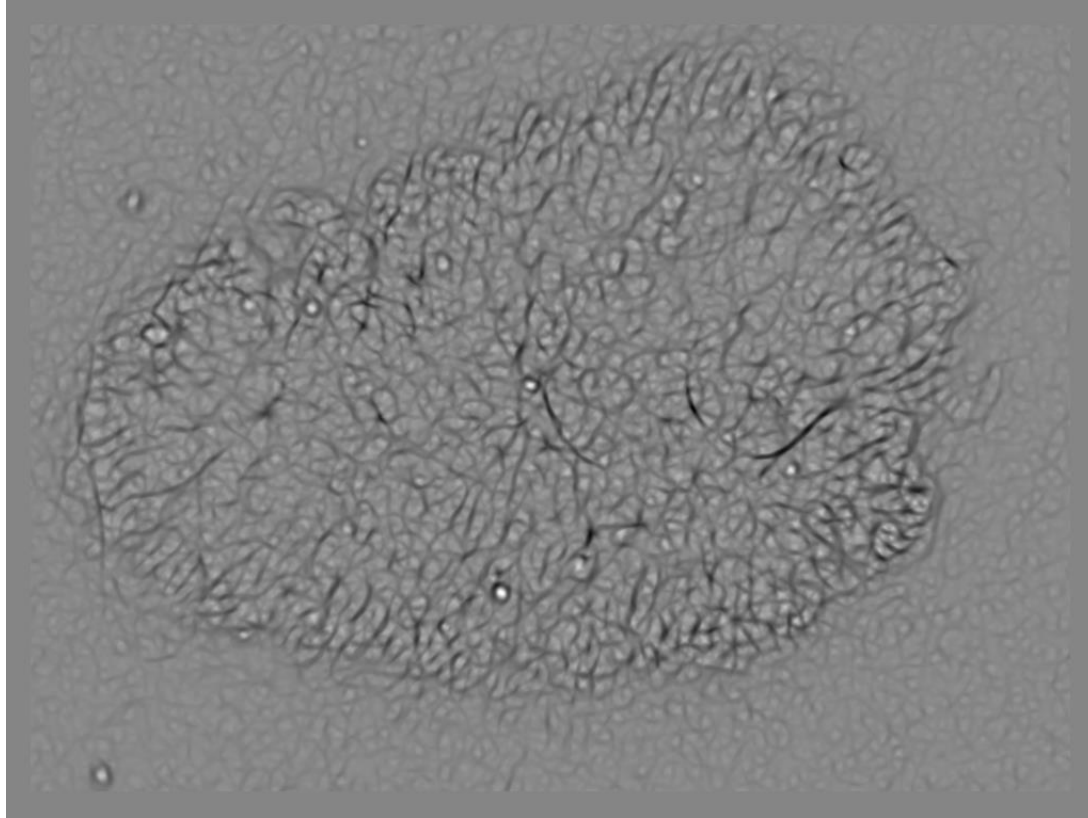




- The large eigenvalue l_{max} of Hessian for $\sigma=1$

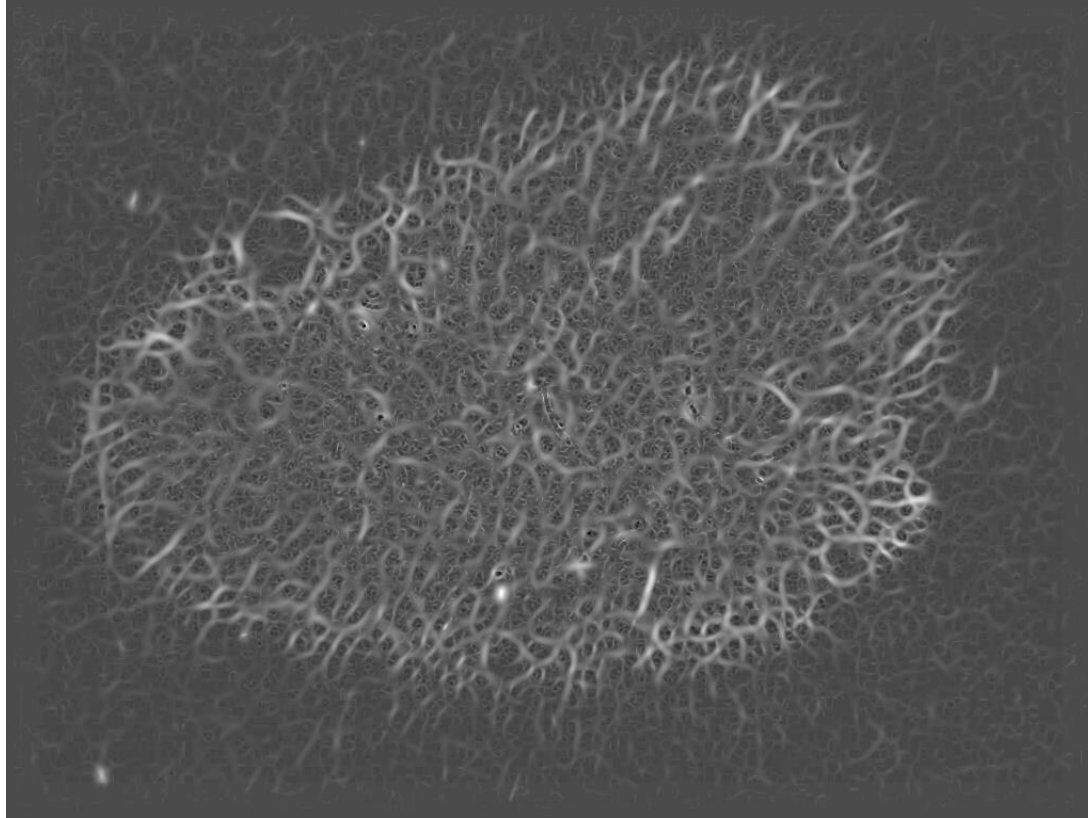


- The large eigenvalue l_{max} of Hessian for $\sigma=4$



- The small eigenvalue l_{min} of Hessian for $\sigma=4$

$\max(\text{Hessian } L_{\max})$



- The $\max(l_{\max})$ of Hessian for $\sigma=1,2,3,4$

Vesselness: δείχνει αν ένα pixel είναι σε αγγείο

- Τα προηγούμενα συνδυάζονται στην ακόλουθη έκφραση που ποσοτικοποιεί την πιθανότητα να ανήκει ένα pixel σε αγγείο

$$\text{vesselness} = \left(1 - e^{-\frac{(\lambda_{\max}^2 + \lambda_{\min}^2)}{2c^2}} \right) e^{-\frac{1}{2b^2} \frac{|\lambda_{\min}|}{|\lambda_{\max}|}} = A \cdot B$$

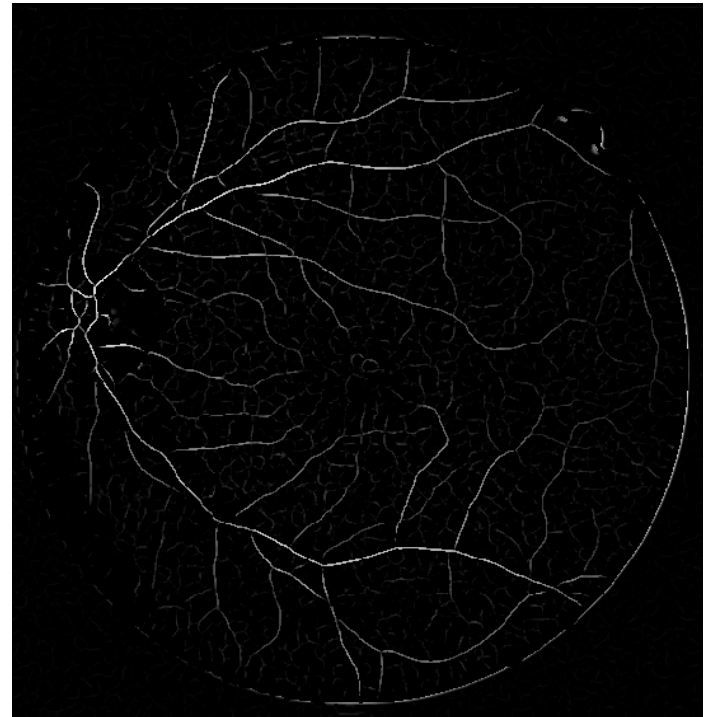
- Ο όρος $A \rightarrow 1$ όταν $|\lambda_{\max}|$ και $|\lambda_{\min}| > 0$ (χρησιμοποιείται για να εξαιρέσει pixels του υποβάθρου)
- $B \rightarrow 1$ όταν $|\lambda_{\max}| \gg$ και $|\lambda_{\min}| \rightarrow 0$

Εντοπισμός αγγείων με διαφορετικές διαμέτρους

- Για κάθε pixel της εικόνας:
 - Το vesselness υπολογίζεται για διαφορετικές τιμές του σ που καλύπτουν την ακτίνα διαμέτρου του αγγείου
 - Υπολογίζεται το μέγιστο vesselness για όλα τα σ .
 - Αν το μέγιστο vesselness είναι Μεγαλύτερο ή ίσο από ένα κατώφλι, τότε το pixel ανήκει σε αγγείο

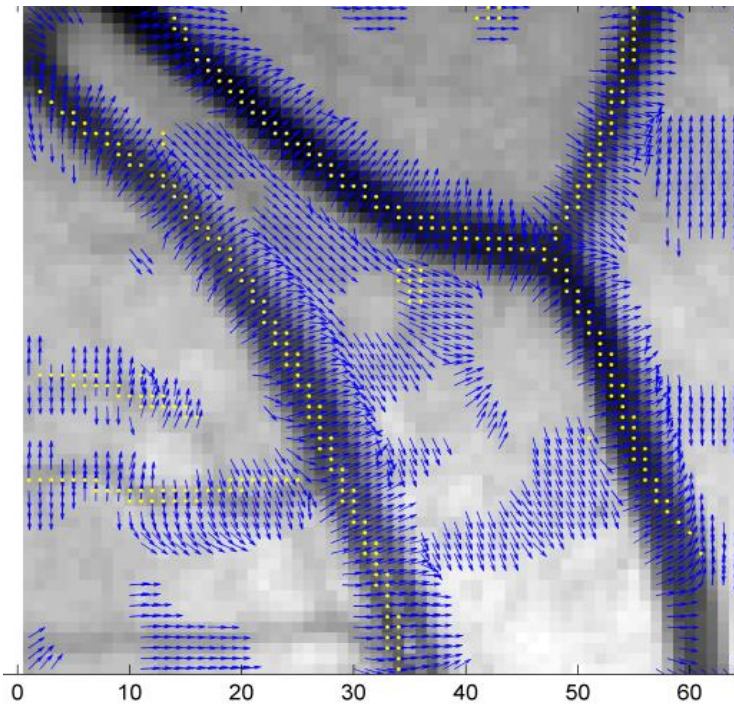


Αρχική εικόνα αμφιβληστροειδούς



Συνάρτηση Vesselness

Illustration of eigenvector $\mathbf{u}$ for $\lambda_{\max} > 0$ in the case of an image with dark vessels, for $\sigma=2$.



$\mathbf{u}$ for pixels with $\lambda_{\max} > 0$



the resulting vessel segmentation

Κατευθυντικά φίλτρα

Steerable filters: Gabor filter-bank

- Πλήρης ορισμός των φίλτρων Gabor

Complex

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right) \quad \text{where}$$

Real

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \quad \text{and}$$

Imaginary

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

$$x' = x \cos \theta + y \sin \theta$$

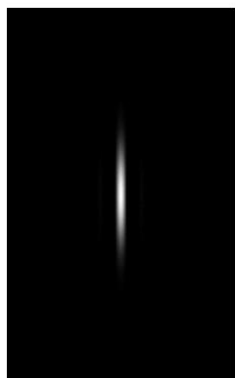
$$y' = -x \sin \theta + y \cos \theta$$

- ορισμός των φίλτρων Gabor με λιγότερες παραμέτρους

$$G_c[i, j] = B e^{-\frac{(i^2 + j^2)}{2\sigma^2}} \cos(2\pi f(i \cos \theta + j \sin \theta))$$

$$G_s[i, j] = C e^{-\frac{(i^2 + j^2)}{2\sigma^2}} \sin(2\pi f(i \cos \theta + j \sin \theta))$$

The effect of Gabor parameters



Sigma (σ) = 10



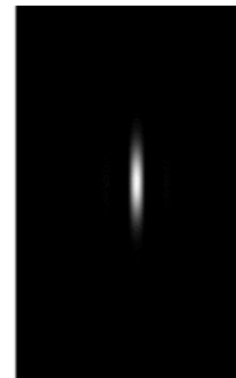
Sigma (σ) = 30



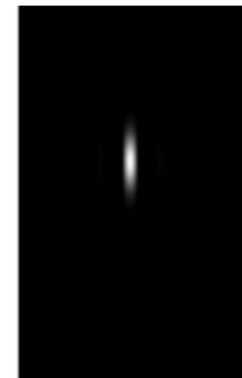
Sigma (σ) = 45



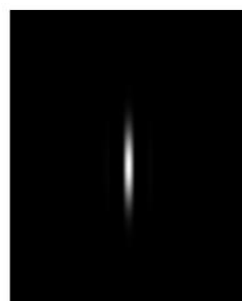
Gamma (γ) = 0.25



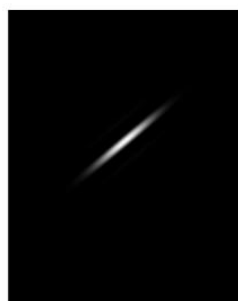
Gamma (γ) = 0.5



Gamma (γ) = 0.75



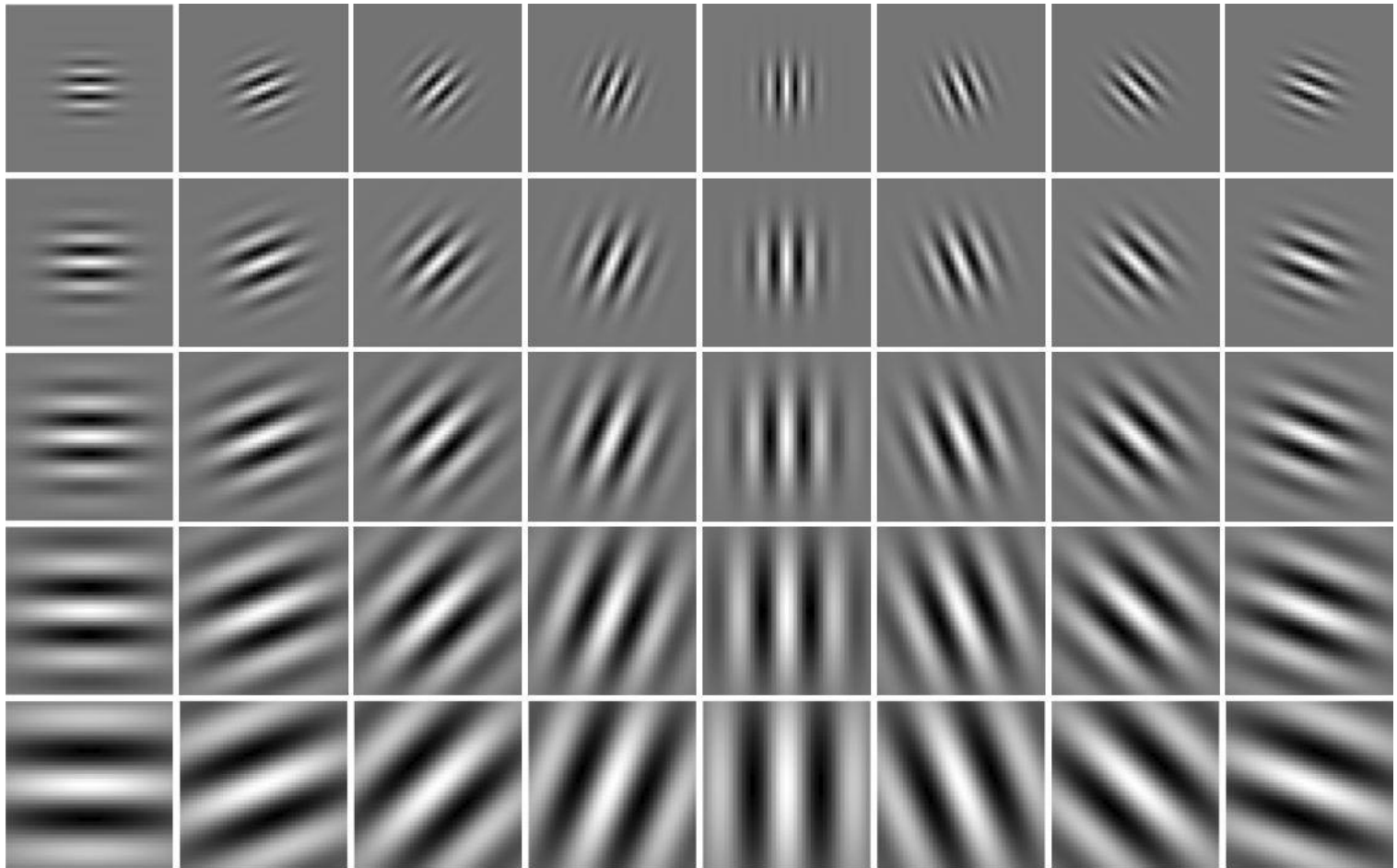
Theta (θ) = 0



Theta (θ) = 45

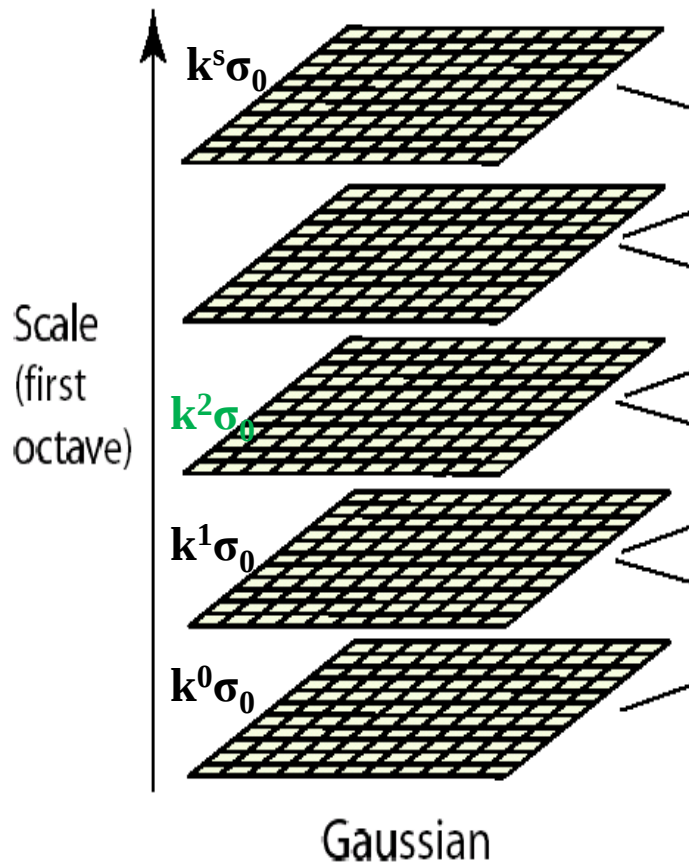


Theta (θ) = 90



The SIFT (Scale Invariant Feature Transform) Detector and Descriptor

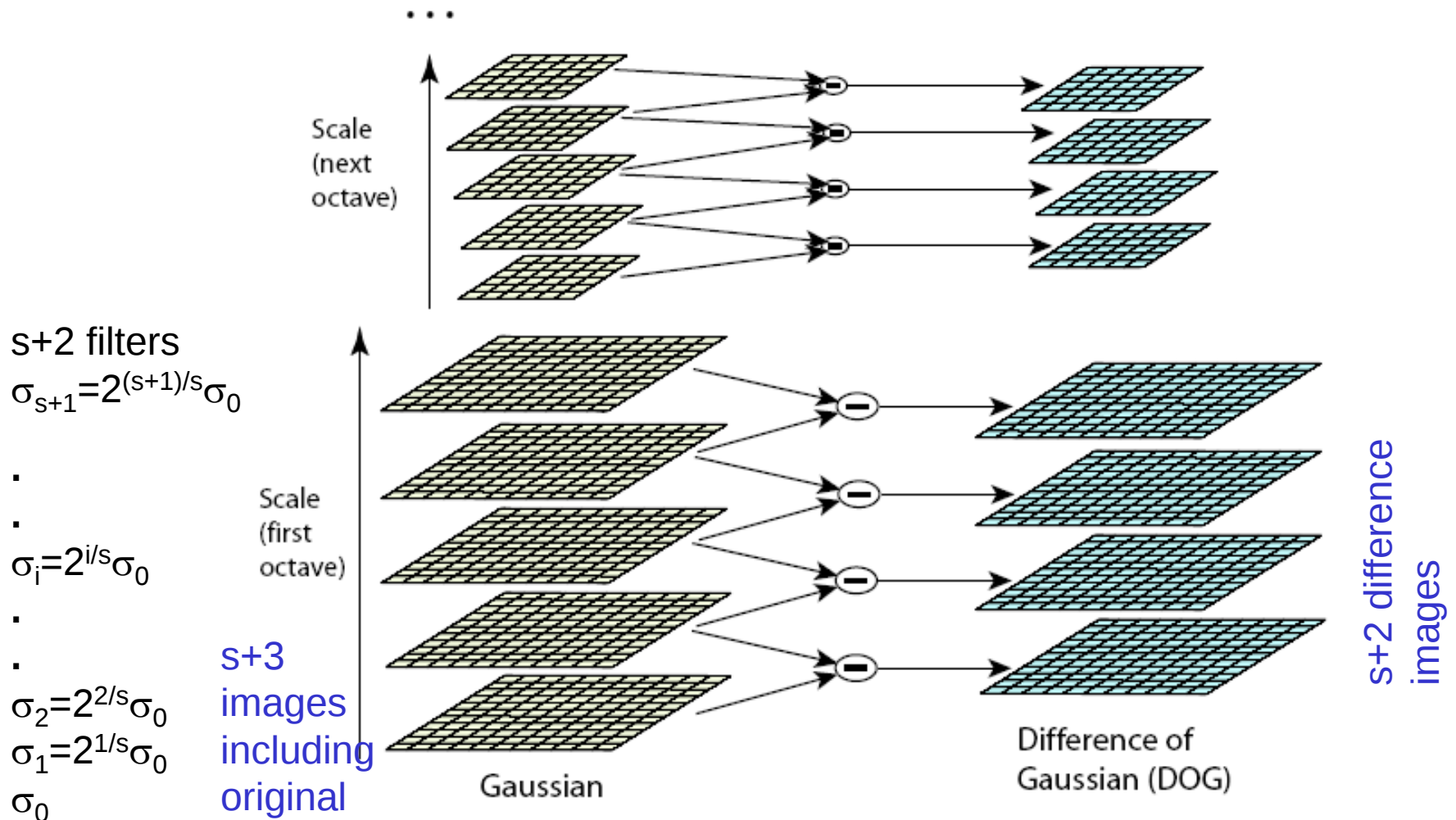
- SIFT is quite an involved algorithm.
- [Constructing a scale space](#) representations of the original image to ensure scale invariance.
- [LoG Approximation](#) LoG is computationally expensive. So approximate it using the representation created earlier.
- [Finding keypoints](#) These are maxima and minima in the Difference of Gaussian image we calculate in step 2
- [Get rid of bad key points](#) Edges and low contrast regions are excluded using [the Harris Corner Detector](#).
- [Assigning an orientation to the keypoints](#) An orientation is calculated for each key point. Any further calculations are done relative to this orientation. This effectively cancels out the effect of orientation, making it rotation invariant.
- [Generate SIFT features](#) for each point. This helps uniquely identify features. Lets say you have 50,000 features. With this representation, you can easily identify the feature you're looking for



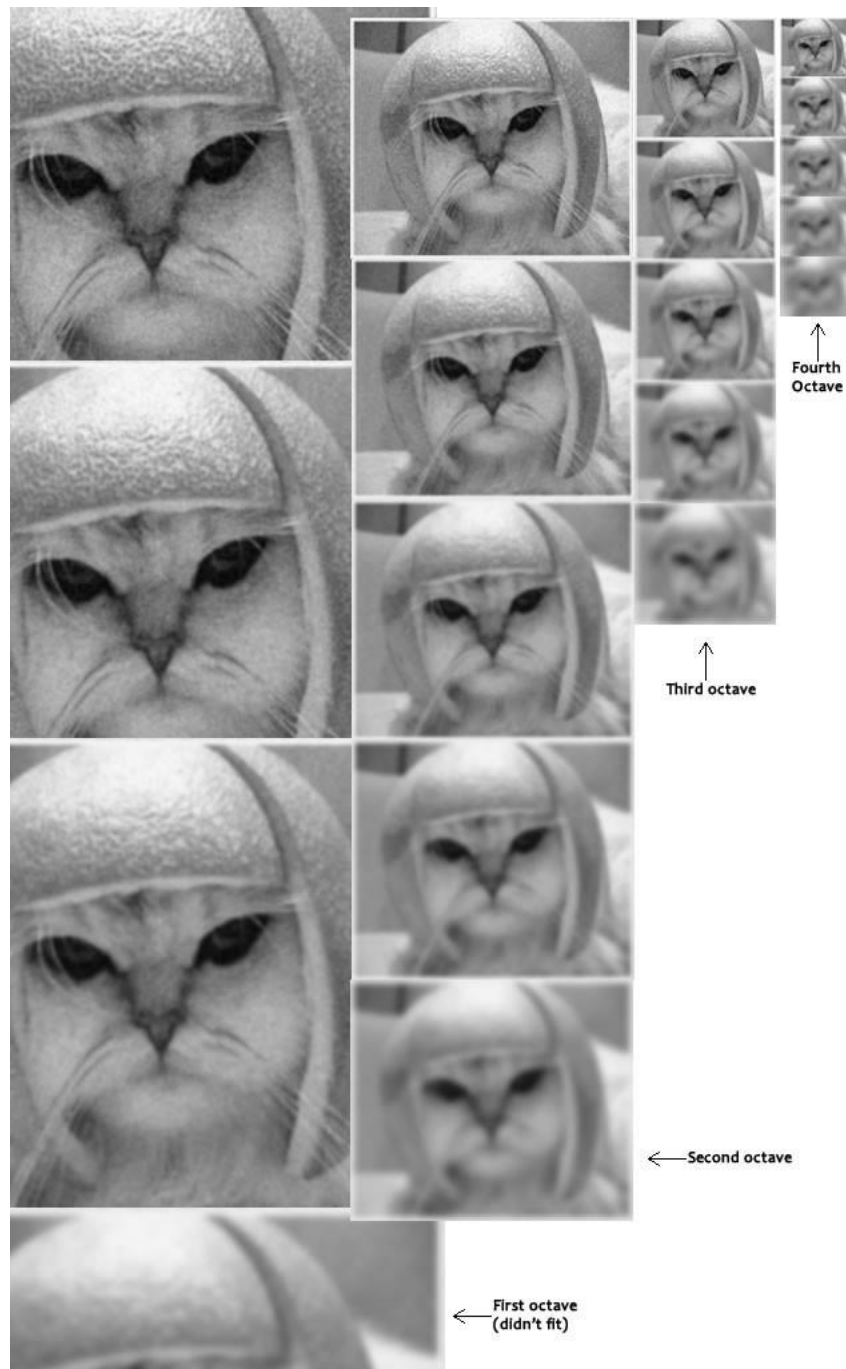
- Images within each octave are separated by a constant factor **k**
- If each octave is divided in **s-1** intervals:

$$\mathbf{k^s=2 \quad or \quad k=2^{1/s}}$$

Lowe's Pyramid Scheme



The parameter s determines the number of images per octave.



SIFT Steps - Review

(1) Scale-space extrema detection

- Extract scale and rotation invariant interest points (i.e., keypoints).

(2) Keypoint localization

- Determine location and scale for each interest point.
- Eliminate “weak” keypoints

(3) Orientation assignment

- Assign one or more orientations to each keypoint.

(4) Keypoint descriptor

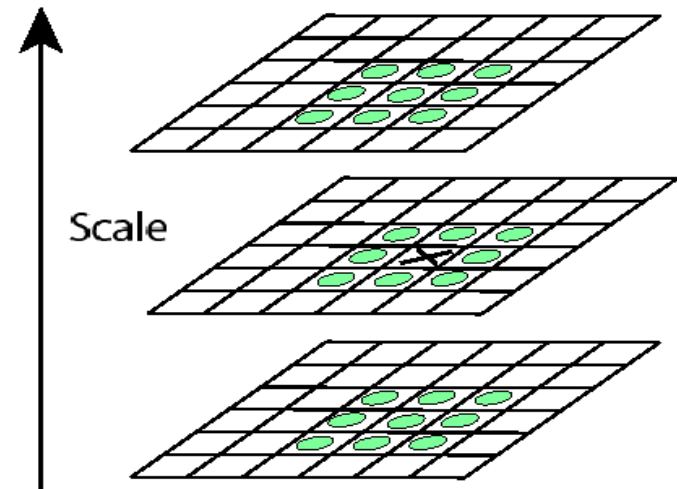
- Use local image gradients at the selected scale.

D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, **International Journal of Computer Vision**, 60(2):91-110, 2004.

Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below

$s+2$ difference images.
top and bottom ignored.
 s planes searched.



For each max or min found, output is the **location** and the **scale**.

Key point localization (cont.)

- The method of LoG max response is used
- Flat points and points on edges have to be rejected:

- Reject flats:

- $|D(\hat{\mathbf{x}})| < 0.03$

- Reject edges:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let α be the eigenvalue with larger magnitude and β the smaller.

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

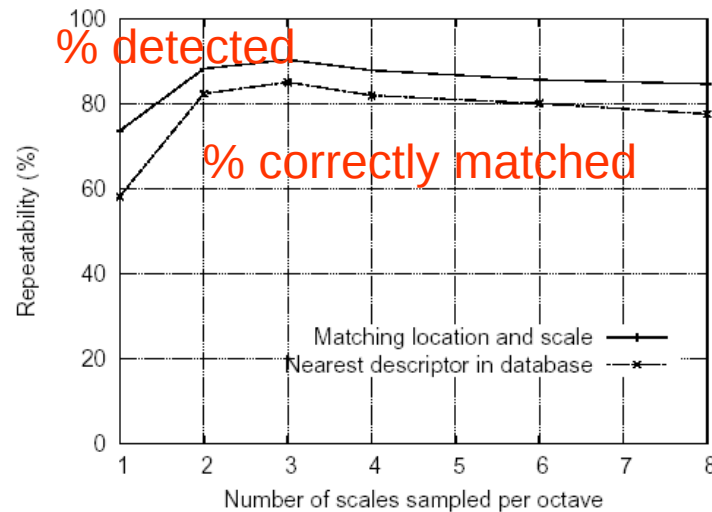
Let $r = \alpha/\beta$.
So $\alpha = r\beta$

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

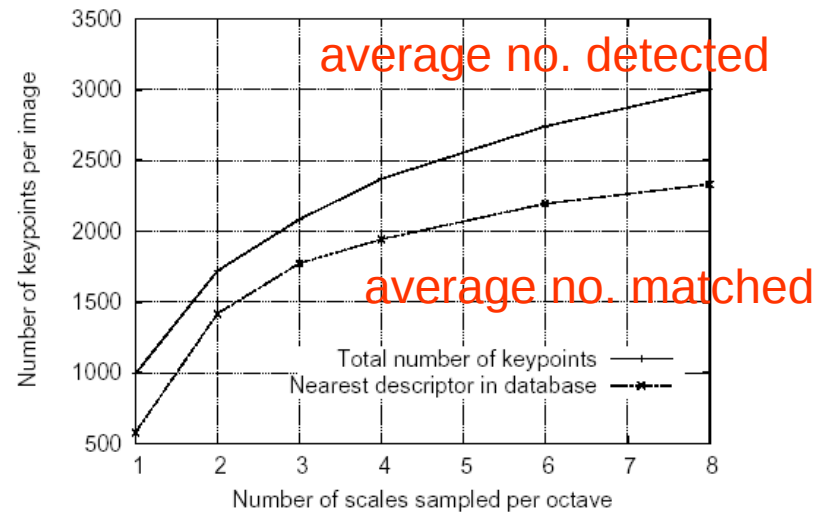
$(r+1)^2/r$ is at a min when the 2 eigenvalues are equal.

- $r < 10$

Scale-space extrema detection: experimental results over 32 images that were synthetically transformed and noise added.



Stability



Expense

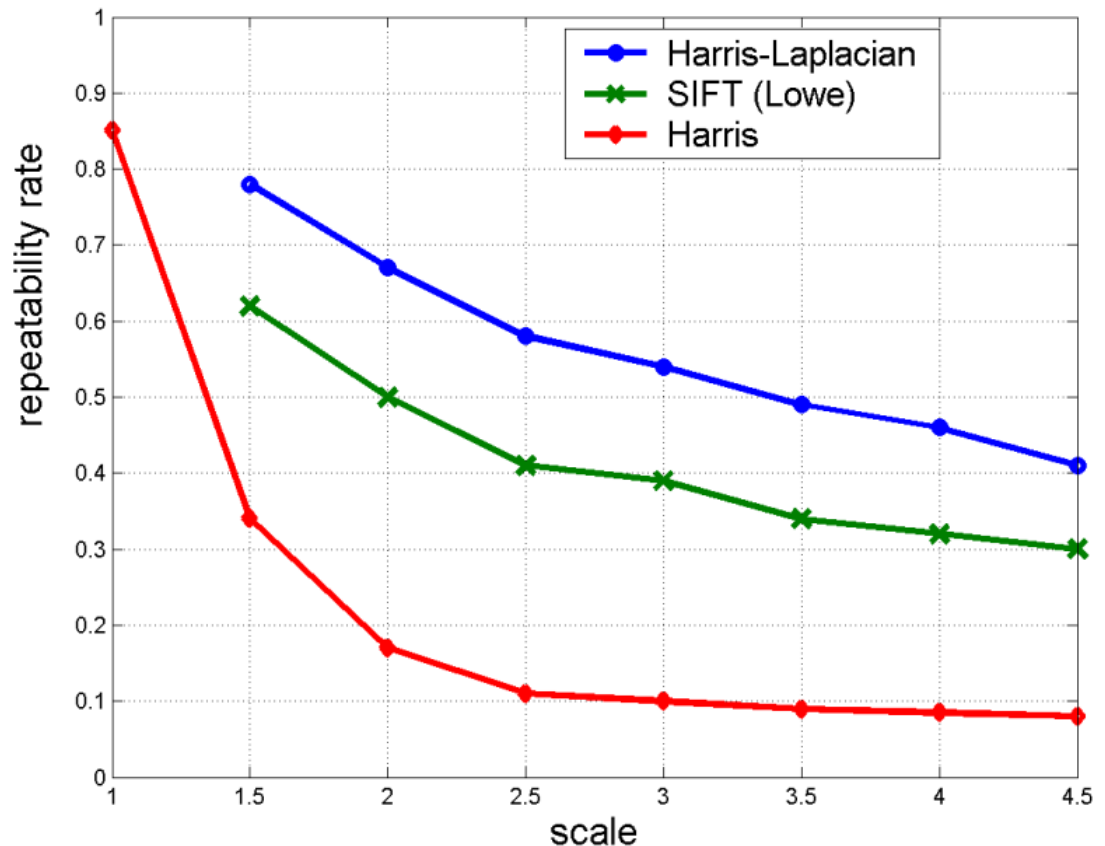
- Sampling in scale for efficiency
 - How many scales should be used per octave? $S=?$
 - More scales evaluated, more keypoints found
 - $S < 3$, stable keypoints increased too
 - $S > 3$, stable keypoints decreased
 - $S = 3$, maximum stable keypoints found

Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

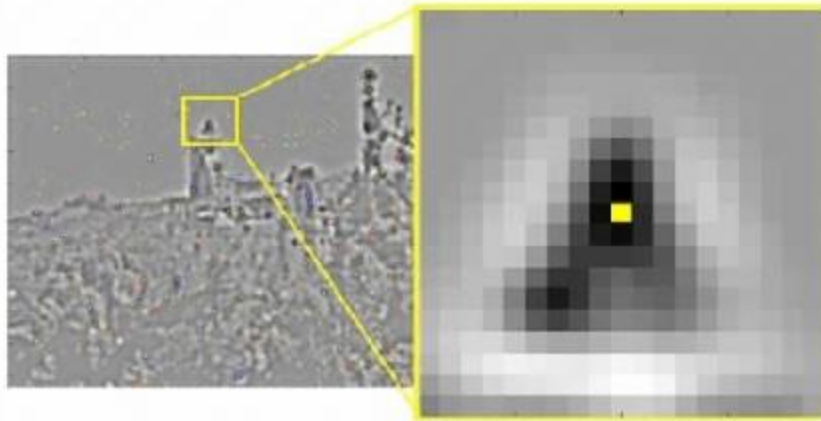
$$\frac{\# \text{ correspondences}}{\# \text{ possible correspondences}}$$



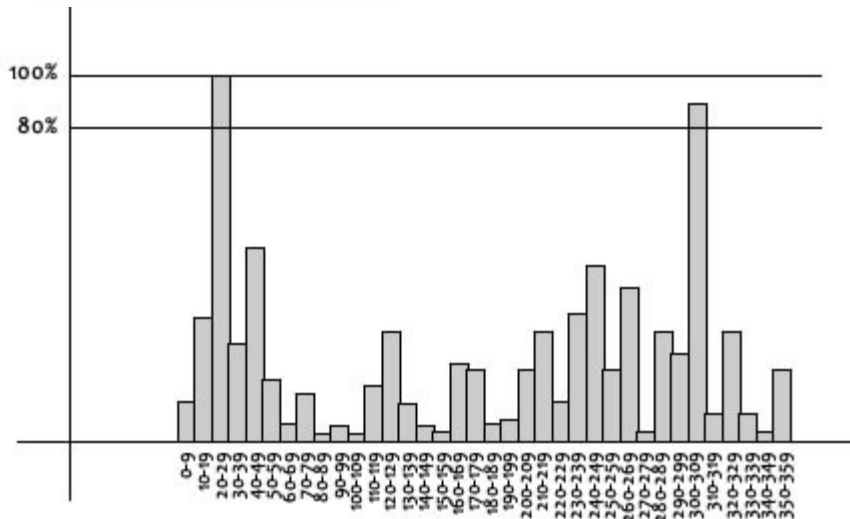
Keypoint localization

- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
 - location, scale, and ratio of principal curvatures
- In initial work keypoints were found at location and scale of a central sample point.
- In newer work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.

3. Orientation assignment



A keypoint



- Create histogram of local gradient directions at selected scale
- Generate smoothed orientation histogram with 36 bins (10^0 each)
- Detect histogram peaks $\geq 80\%$
- Assign this orientation to this point
- If multiple peaks in hist orientation, create copies of the point, one for each peak

Keypoint localization with orientation

233x189



832

initial keypoints

729



keypoints after
gradient threshold



536

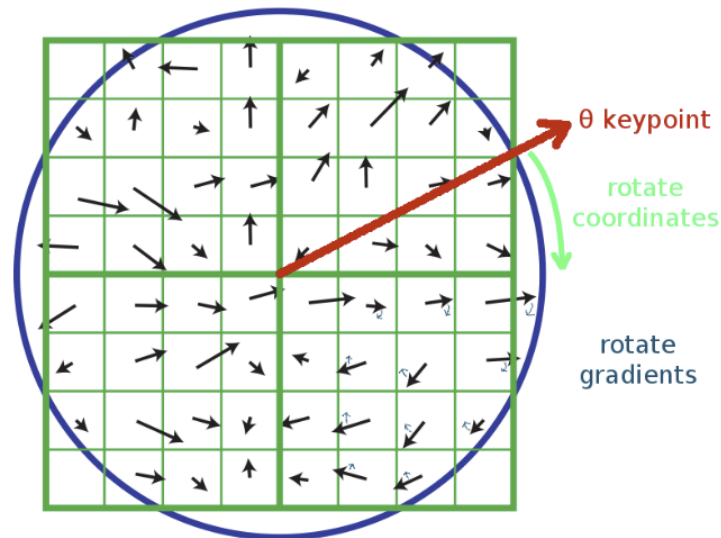
keypoints after
ratio threshold

4. Keypoint Descriptors

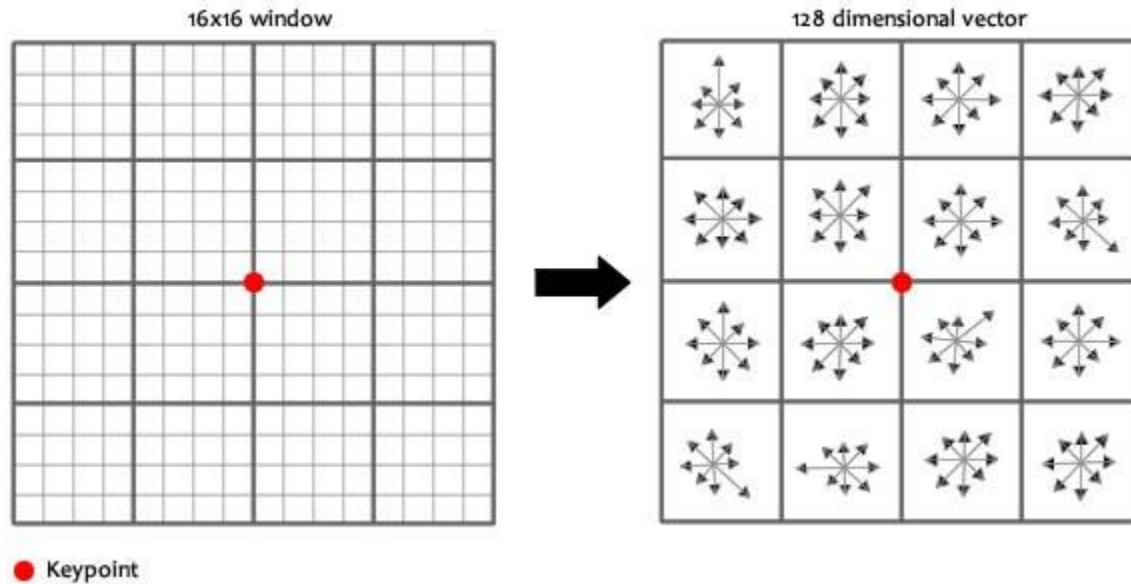
- At this point, each keypoint has
 - location
 - scale
 - orientation
- Next is to compute a descriptor for the local image region about each keypoint that is
 - highly distinctive
 - invariant as possible to variations such as changes in viewpoint and illumination

Normalization

- Rotate the window to standard orientation
- Scale the window size based on the scale at which the point was found.



Lowe's Keypoint Descriptor



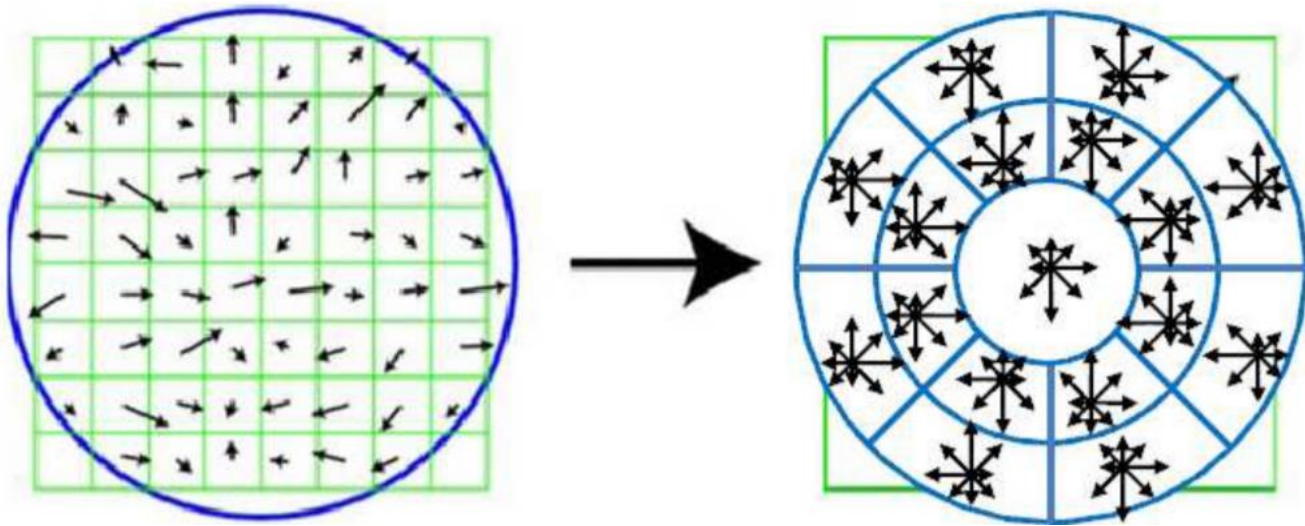
Lowe's Keypoint Descriptor

- use the **normalized** region about the keypoint
- compute gradient magnitude and orientation at each point in the region
- **weight them by a Gaussian** window overlaid on the circle
- create an **orientation histogram** over the 4 X 4 subregions of the window
- 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 times 8 directions gives a **vector of 128 values**.



SIFT variant: GLOH

Gradient Location Oriented Histograms



SIFT variant: SIFT PCA

- PCA is often used for dimensionality reduction

SIFT-PCA: training phase

- Off-line (at the beginning): use many relevant images
- Apply SIFT to detect points
- Define feature **fv** for each point:
 - Define 41x41 image patch
 - For each pixel in 41x41 calculate the x,y gradient component $\rightarrow 2 \times 39 \times 39 = 3042$ long feature vector **fv**
 - Use Gaussian with $\sigma=1.5$, normalize the **fv**.
- In [SIFT-PCA] 21.000 patches from a number of images were extracted

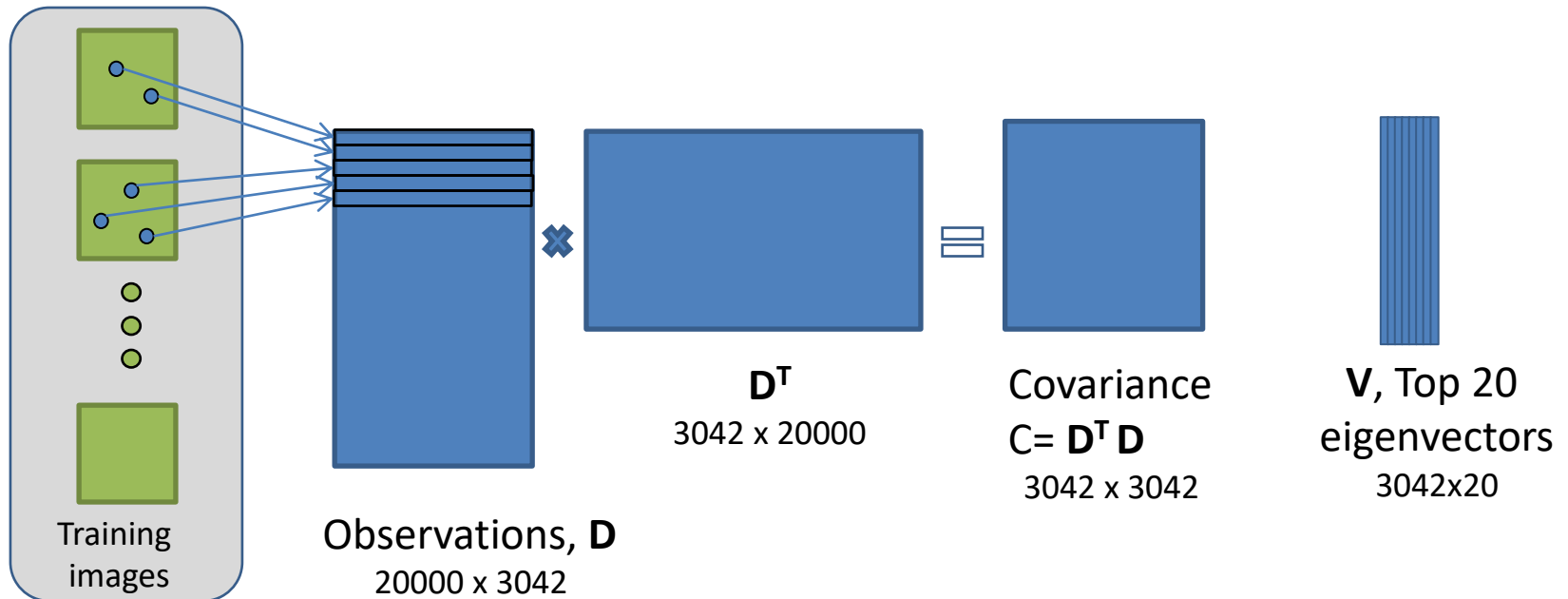
Off-line part

- Generate the covariance matrix $\mathbf{M}$
- Create the eigen-space using the K eigenvectors $\mathbf{v}_k$ of $\mathbf{M}$ with the greatest eigenvals

Online part

- For each new $\mathbf{f}_v$ from a new image:
 - Project the $\mathbf{f}_v$ onto the $\mathbf{v}_k$ to generate a vector of dimensionality k ($k=20$)

Off-line training



Testing



SURF: Speeded Up Robust Features

- Speed-up computations by fast approximation of (i) Hessian matrix and (ii) descriptor using “integral images”.

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix},$$

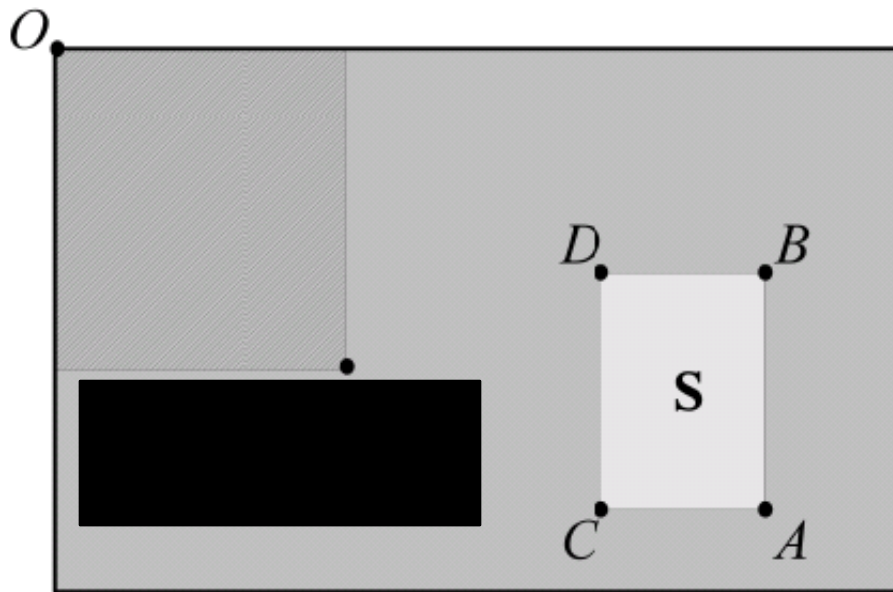
- What is an “integral image”?

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, “SURF: Speeded Up Robust Features”, **European Computer Vision Conference (ECCV)**, 2006.

Integral Image

- The integral image $I_{\Sigma}(x,y)$ of an image $I(x,y)$ represents the sum of all pixels in $I(x,y)$ of a rectangular region formed by $(0,0)$ and (x,y) .

$$I_{\Sigma}(i,j) = I_{\Sigma}(i-1,j) + I_{\Sigma}(i,j-1) - I_{\Sigma}(i-1,j-1) + I(i,j)$$



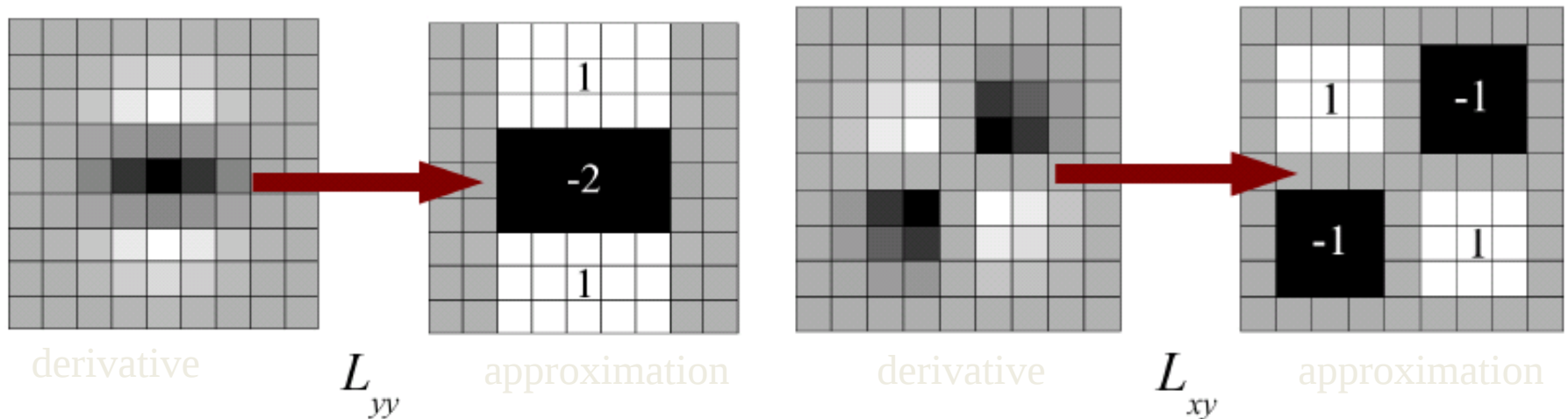
Using integral images, it takes only **four** array references to calculate the sum of pixels over a rectangular region of any size.

$$S = A - B - C + D$$

SURF: Speeded Up Robust Features (cont'd)

- Approximate L_{xx} , L_{yy} , and L_{xy} using box filters.

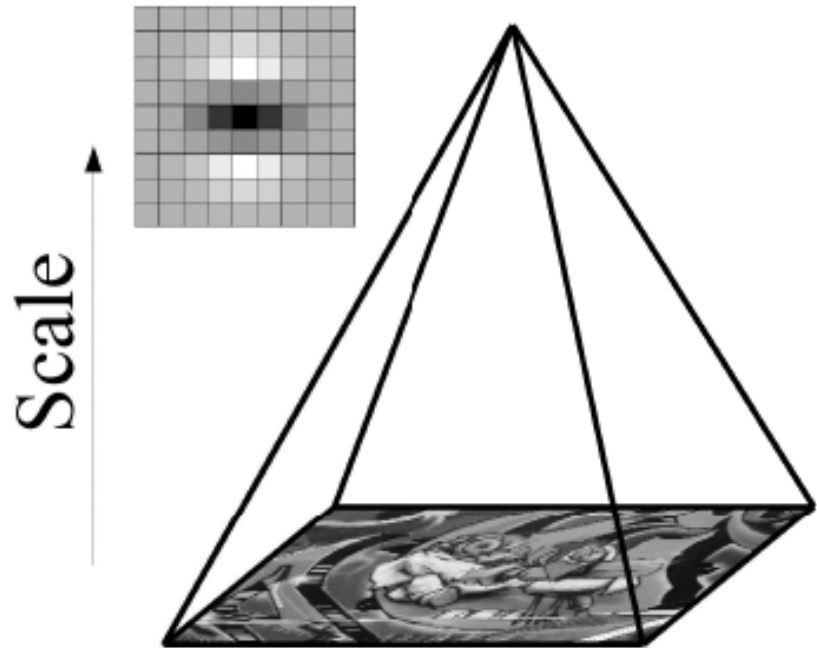
(box filters shown are 9 x 9 – good approximations for a Gaussian with $\sigma=1.2$)



Can be computed very fast using integral images!

SURF: Speeded Up Robust Features (cont'd)

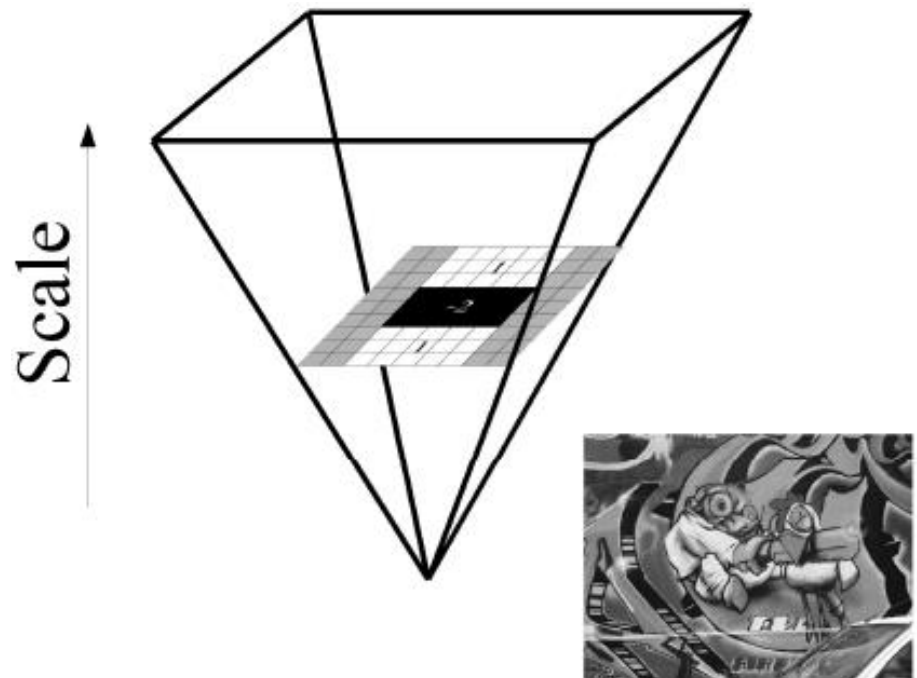
In SIFT, images are repeatedly smoothed with a Gaussian and subsequently sub-sampled in order to achieve a higher level of the pyramid.



SURF: Speeded Up Robust Features (cont'd)

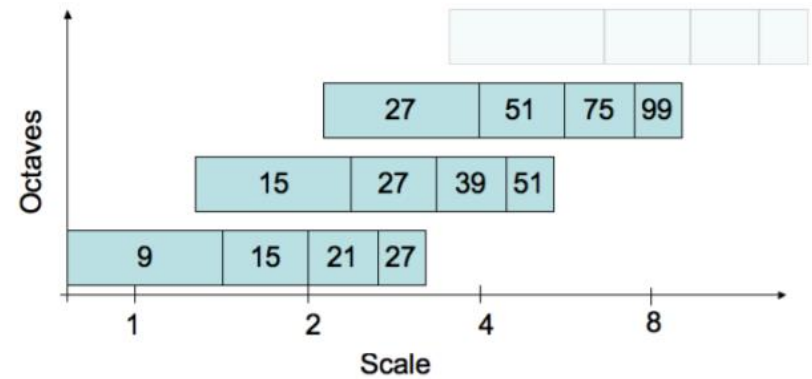
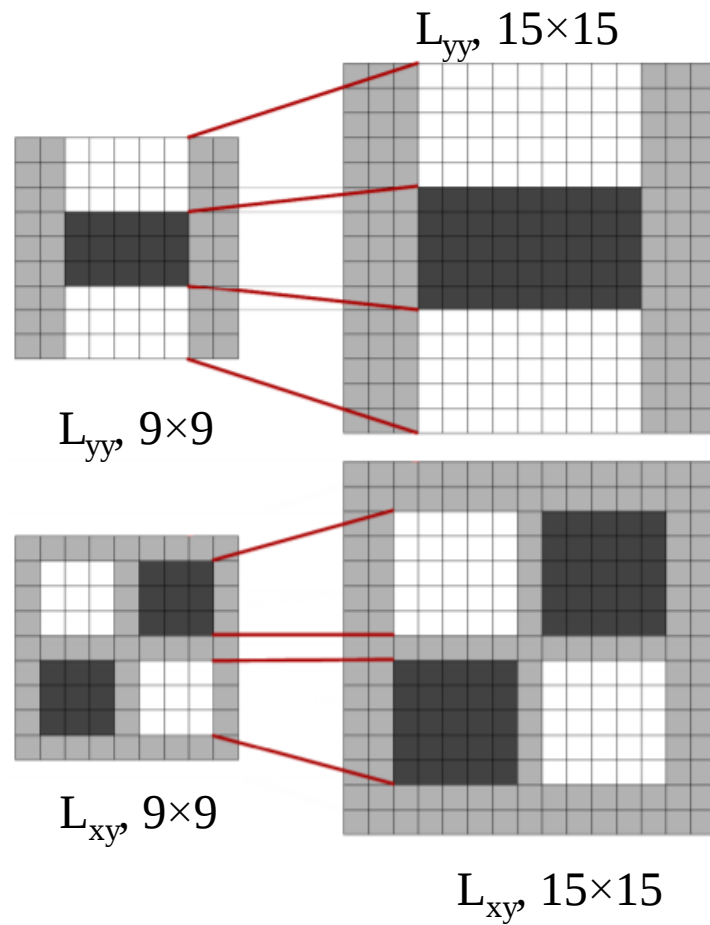
Alternatively, we can use filters of larger size on the original image.

Due to using integral images, filters of any size can be applied at exactly the same speed!



(see Tuytelaars' paper for details)

- Filter sizes used in SURF:



Filter sizes for different octaves

SURF: Speeded Up Robust Features (cont'd)

- Approximation of H :

Using DoG

Using box filters

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix},$$
$$SIFT : H_{approx}^{SIFT} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$
$$SURF : H_{approx}^{SURF} = \begin{bmatrix} \hat{L}_{xx} & \hat{L}_{xy} \\ \hat{L}_{yx} & \hat{L}_{yy} \end{bmatrix}$$

SURF: Speeded Up Robust Features (cont'd)

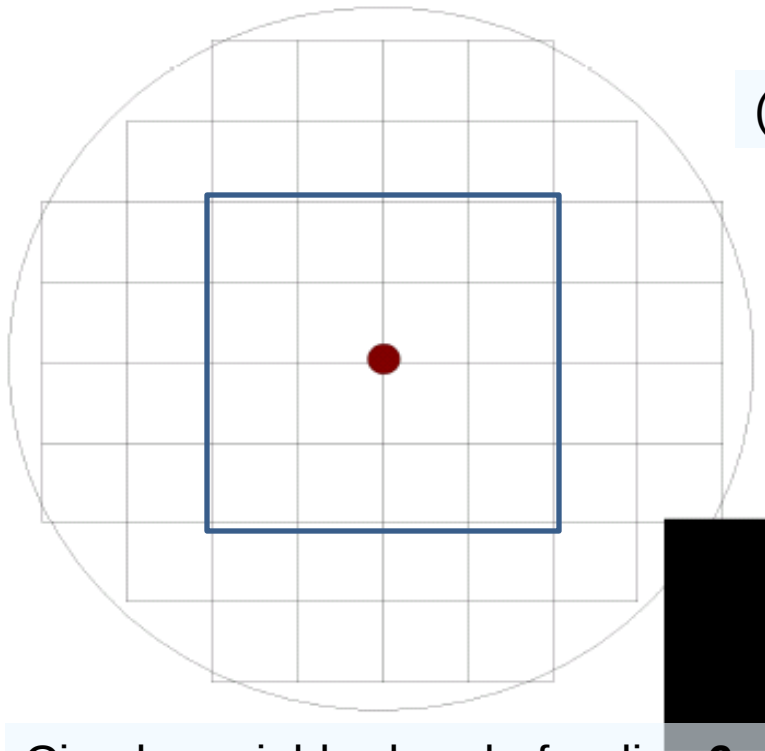
- Instead of using a different measure for selecting the location and scale of interest points (e.g., Hessian and DOG in SIFT), SURF uses the determinant of H_{approx}^{SURF} to find both.
- Determinant elements must be weighted to obtain a good approximation:

$$\det(H_{approx}^{SURF}) = \hat{L}_{xx}\hat{L}_{yy} - (0.9\hat{L}_{xy})^2$$

SURF: Speeded Up Robust Features (cont'd)

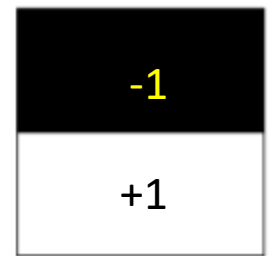
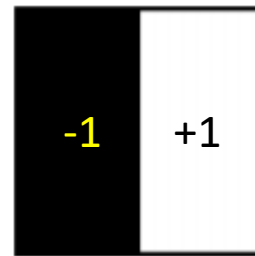
- Once interest points have been localized both in space and scale, the next steps are:
 - (1) Orientation assignment
 - (2) Keypoint descriptor

SURF: Orientation assignment



Circular neighborhood of radius 6σ
around the interest point
(σ = scale used for point detection)

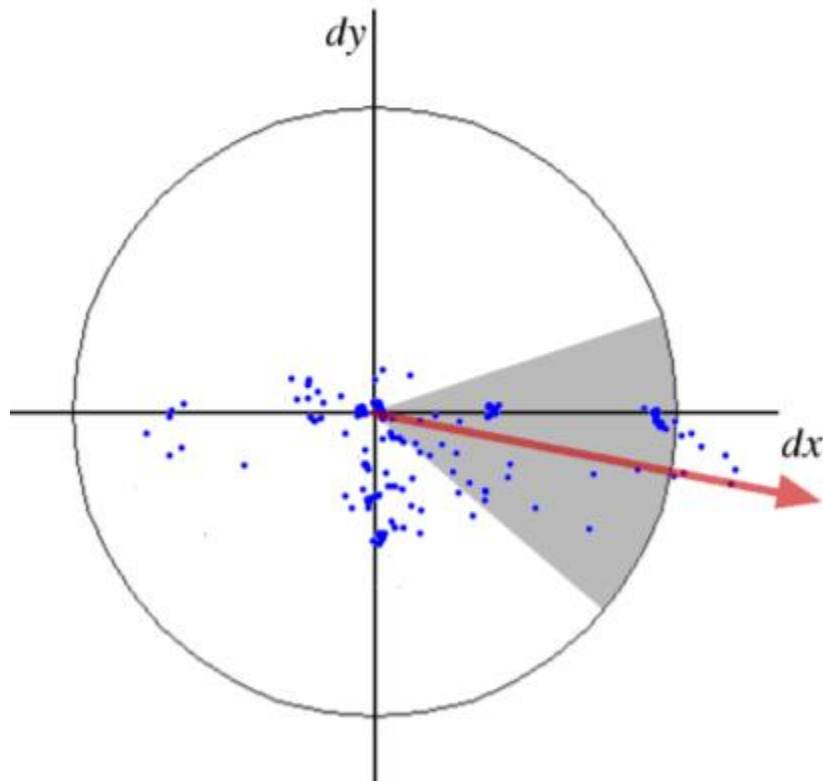
$$(x \text{ response}, y \text{ response}) = (\sum dx, \sum dy)$$



Haar wavelets (responses weighted
with Gaussian)
Side length = 4σ

Can be computed very fast using integral images!

SURF: Orientation assignment



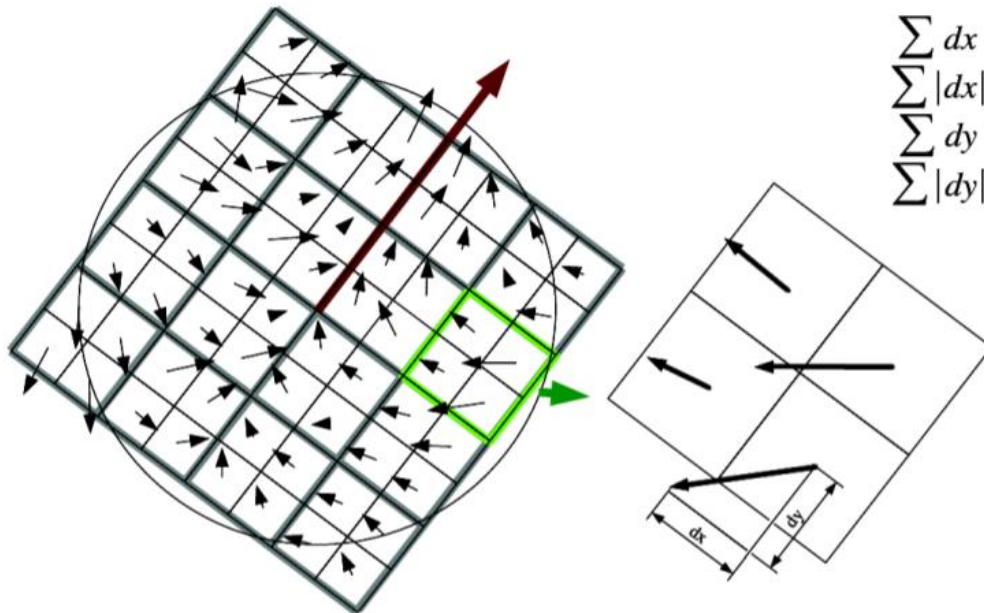
Circular neighborhood of radius 6σ around the interest point (σ = the scale at which the point was detected)

$$(\sum dx, \sum dy)$$

Example of orientation detection 60° angle

SURF: Speeded Up Robust Features (cont'd)

- Keypoint descriptor (square region of size 20σ)



$$\begin{aligned} &\sum dx \\ &\sum |dx| \\ &\sum dy \\ &\sum |dy| \end{aligned}$$

- Sum the response over each sub-region for d_x and d_y separately.
- To bring in information about the polarity of the intensity changes, extract the sum of absolute value of the responses too.

Feature vector size:

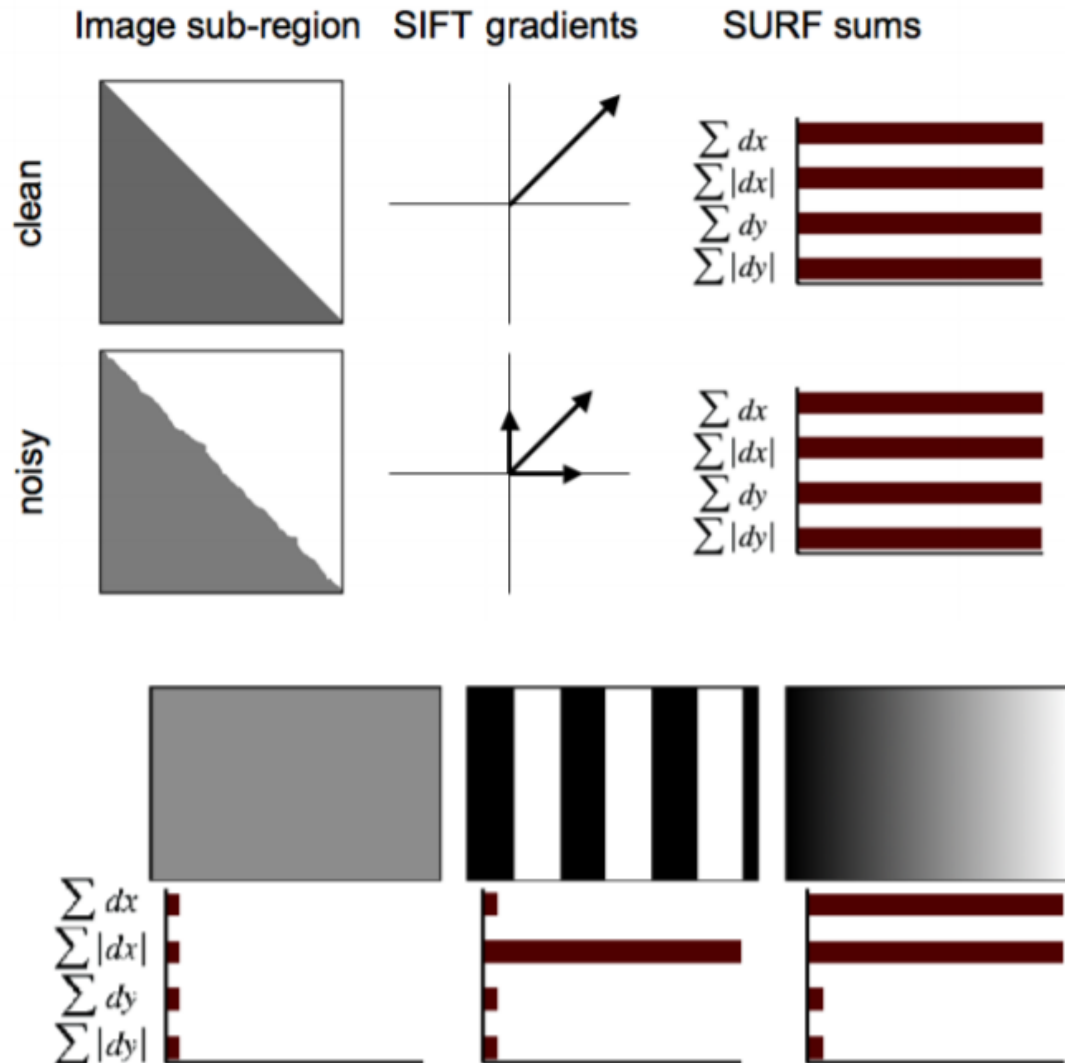
$$4 \times 16 = 64$$

SURF: Speeded Up Robust Features (cont'd)

SURF-128

- The sum of d_x and $|d_x|$ are computed separately for points where $d_y < 0$ and $d_y > 0$
- Similarly for the sum of d_y and $|d_y|$
 - **More discriminatory!**

SURF: Features examples



KAZE features:

Similar to SURF, but non-linear scale-space

- Instead of consecutive convolutions with LoG, or DoG, non-linear diffusion is applied on the initial image
- A set of increasingly diffused images is constructed (similar to the Laplacian pyramid)
- The concept of octaves and levels within each octave is retained. As in SURF, no image subsampling is performed
- The filtered (diffused) images are normalized (using the equivalent of σ^2) and local extrema (along space and scale) are detected
- The principal orientation is extracted as in SURF
- The feature vector is constructed in a way similar to SURF

- Diffusion equation

$$\frac{\partial I}{\partial t} = I_t = \text{div}(a \nabla I) = \alpha \cdot \text{div}(\nabla I) = \alpha (\mathbf{i} \partial_x + \mathbf{j} \partial_y) (\mathbf{i} I_x + \mathbf{j} I_y) = I_{xx} + I_{yy}$$

- Non linear diffusion

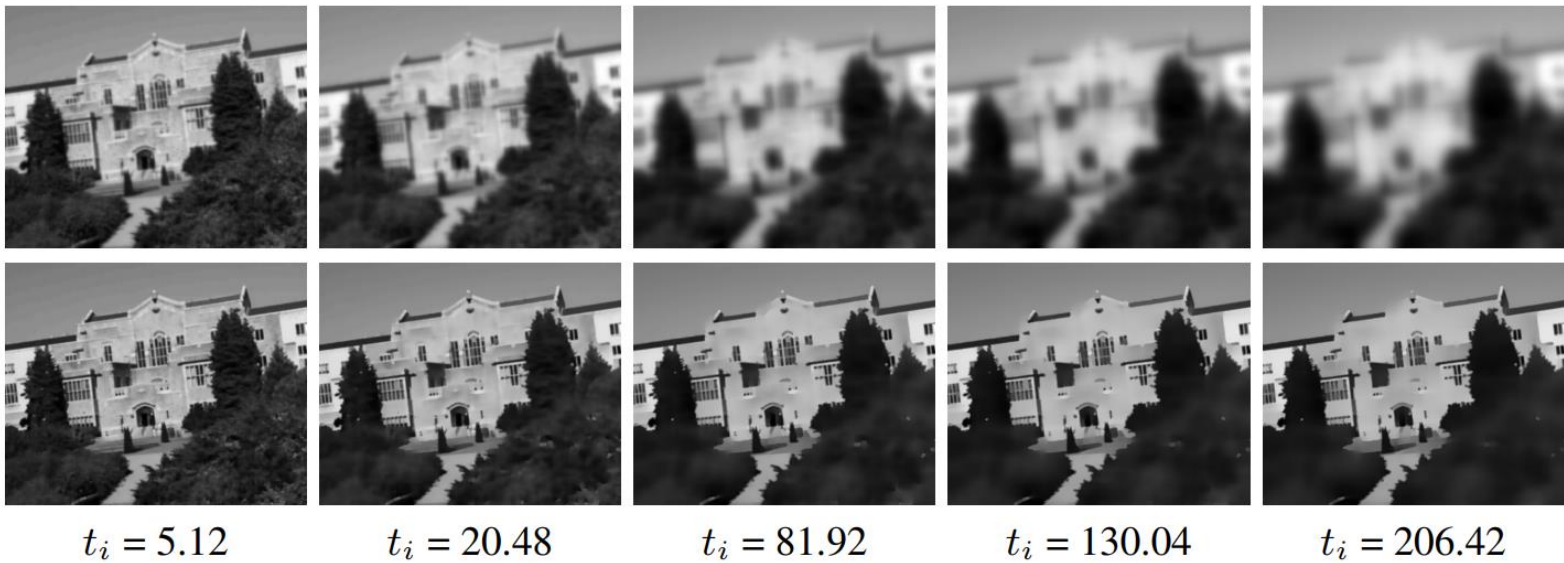
$$\frac{\partial I}{\partial t} = I_t = \text{div}(a(x, y) \nabla I)$$

- Υπολογιστική υλοποίηση της εξίσωσης διάχυσης

$$\frac{\partial I}{\partial t} = I_t = I_{xx} + I_{yy} \Rightarrow \frac{I_{t+\delta t} - I_t}{\delta t} = I_t * LoG \Rightarrow$$

$$I_{t+\delta t} = I_t + \delta t \cdot I_t * LoG$$

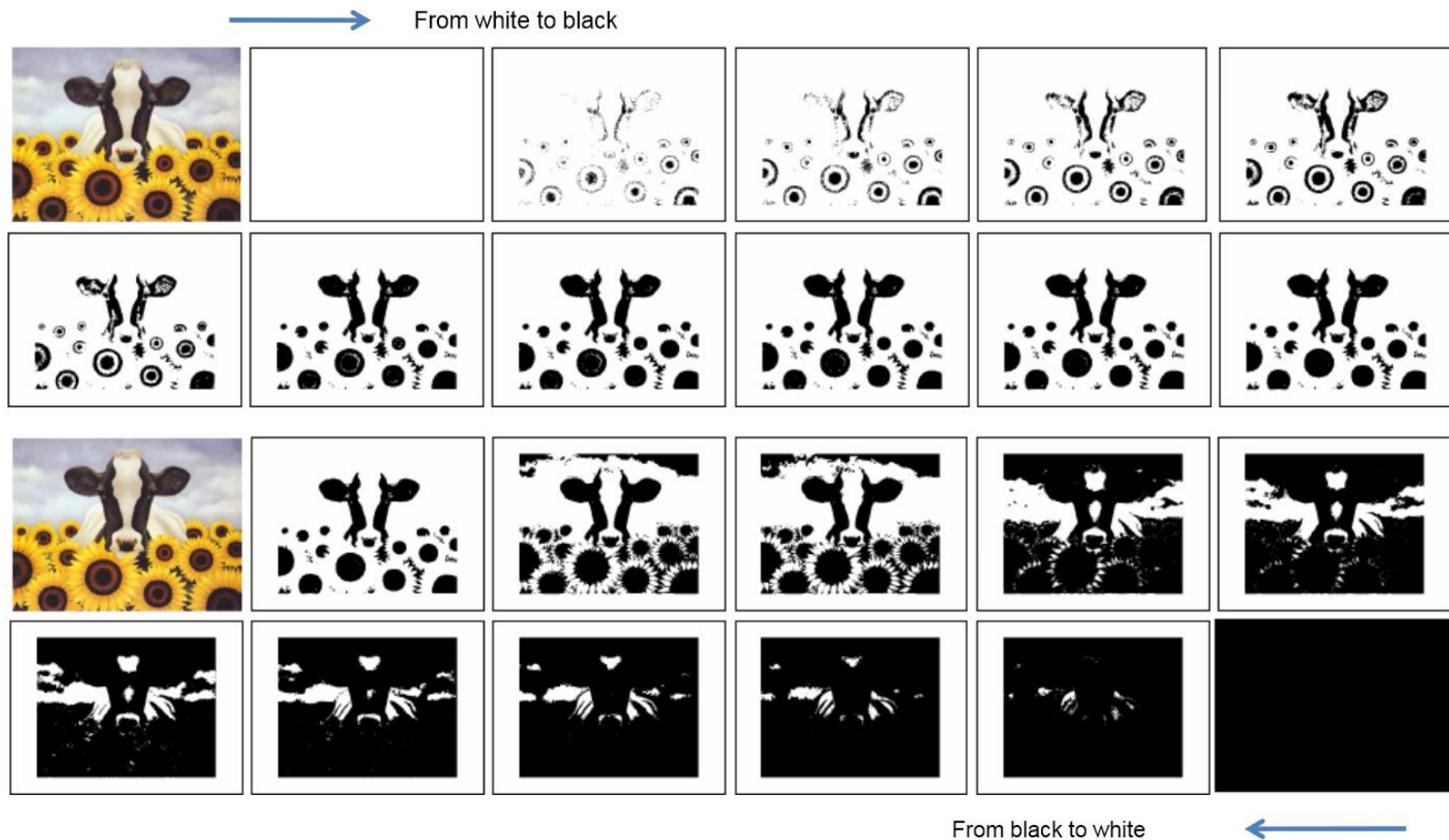
- Αποδεικνύεται ότι η λύση με πεπερασμένες διαφορές είναι ευσταθής για $\delta t < 0.25$.
- Άλλες αριθμητικές μέθοδοι υπάρχουν για μεγαλύτερα δt και ταχύτερη εκτέλεση (πχ Additive operator splitting –AOS)



- Παράδειγμα εφαρμογής της μη ισοτροπικής διάχυσης

Affine invariant feature detectors

- SIFT, SURF and their variants are rotation and scale invariant
- To achieve Affine invariant features → estimate different scale along each axis
- One example: Maximally Stable Extremal Regions (MSER)



Example of threshold sweeping and MSER detection

http://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A34%20MSER.pdf

- MSER performs well on images containing homogeneous regions with distinctive boundaries.
- MSER works well for small regions
- MSER doesn't work well with images with any motion blur
- Multiresolution MSER provides better robustness to large scale changes and blurred images and improves matching performance over large scale changes and for blurred images
- Good repeatability
- Affine invariant
- A smart implementation makes it one of the fastest region detectors

- T. Tuytelaars, K. Mikolajczyk (2008), "Local Invariant Feature Detectors: A Survey", Foundations and Trends® in Computer Graphics and Vision: Vol. 3(3), pp 177-280.
 - <http://epubs.surrey.ac.uk/726872/>
- D. G. Lowe, Distinctive image features from scale-invariant keypoints. IJCV, vol. 2, no. 60, pp. 91-110, 2004
- K. Mikolajczyk and C. Schmid, An affine invariant interest point detector. ICCV, vol. 2350, pp. 128-142, 2002
- K. Mikolajczyk, PhD Thesis
- [<https://courses.cs.washington.edu/courses/cse576/06sp/notes/Interest2.pdf>]

- Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998, October). Multiscale vessel enhancement filtering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 130-137). Springer Berlin Heidelberg.
- Delibasis, K. K., Kechriniotis, A., & Maglogiannis, I. (2013, September). On centered and compact signal and image derivatives for feature extraction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 318-327). Springer Berlin Heidelberg.