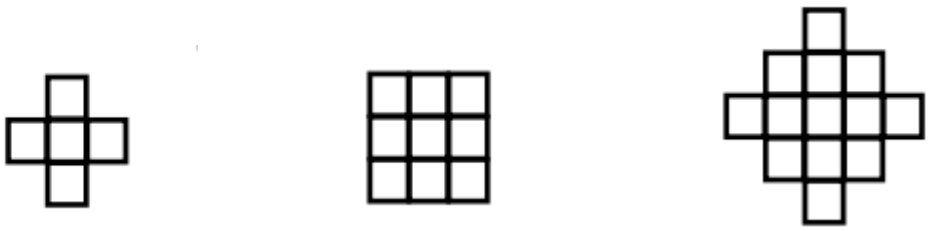
Πανεπιστήμιο Κύπρου
Τμήμα Πληροφορικής

ΕΠΛ445:
Ψηφιακή Επεξεργασία Εικόνας

Εργαστήριο 2

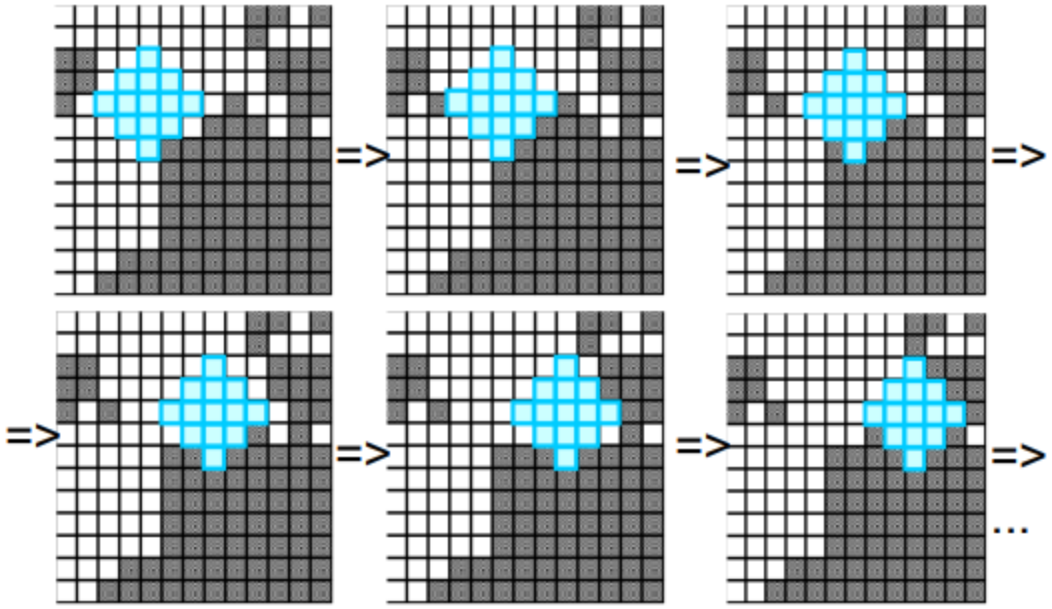
Δομικό στοιχείο (structuring element) ή πυρήνας (kernel)

- Είναι μια γεωμετρική συσχέτιση μεταξύ εικονοστοιχείων (pixels)
- Αποτελεί επίσης έναν τρόπο συλλογής της τοπικής φωτεινότητας μιας εικόνας
- Συχνά αναφέρεται ως ένα κινούμενο παράθυρο
- Συνήθως εφαρμόζεται γραμμή προς γραμμή και στήλη προς στήλη



Μορφολογικοί μετασχηματισμοί

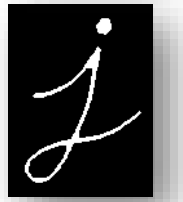
- Ορίζονται με τη μετακίνηση ενός δομικού στοιχείου (structuring element) πάνω σε μια εικόνα, με τέτοιο τρόπο ώστε το δομικό στοιχείο να είναι κεντραρισμένο σε κάθε pixel της εικόνας.



Μορφολογικοί μετασχηματισμοί

Δίδεται ένα παράθυρο B και μια δυαδική εικόνα I

- **Διαστολή (dilate):** μεγέθυνση αντικειμένου (OR)
  - Αυξάνει τη λευκή περιοχή του αντικειμένου



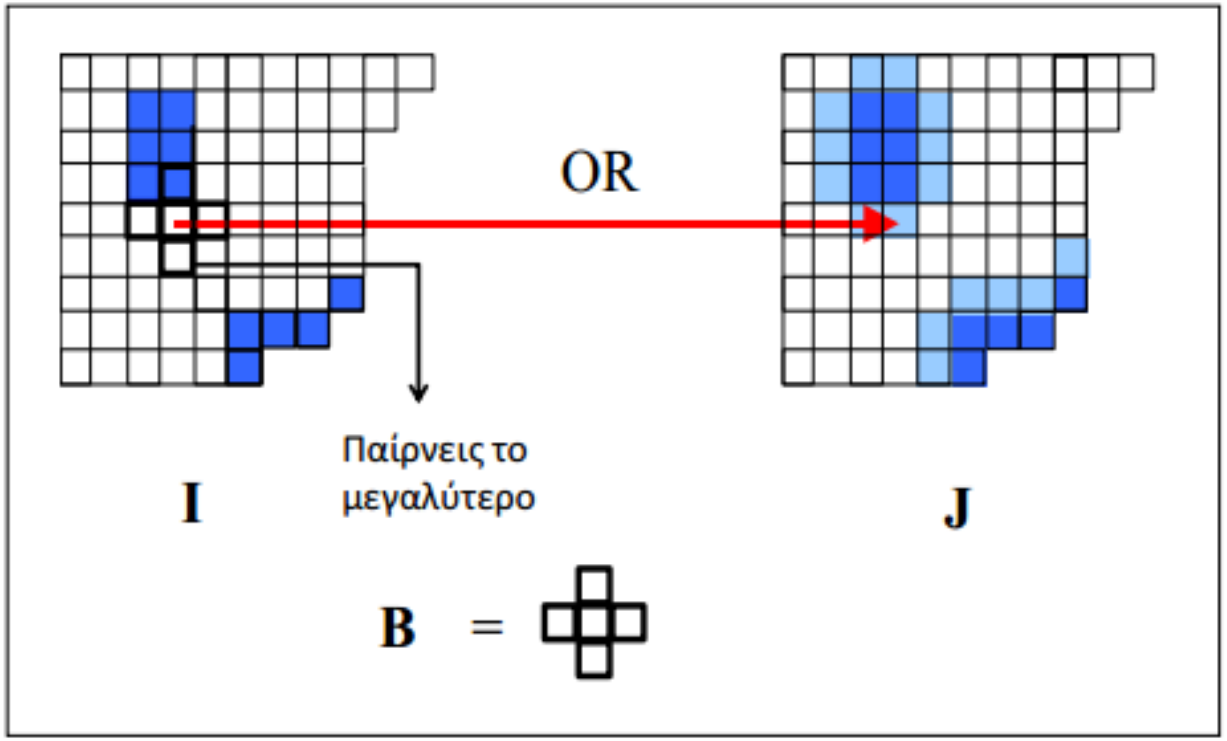
- **Συστολή (erode):** συστολή αντικειμένου (AND)
  - Είναι χρήσιμο για την αφαίρεση μικρών λευκών θορύβων, ξεχωρίζει δύο συνδεδεμένα αντικείμενα κλπ. Μειώνει τη λευκή περιοχή του αντικειμένου.

- **Μορφολογικό opening** είναι μια συστολή (erosion) ακολουθούμενη από μια διαστολή (dilation) με το ίδιο δομικό στοιχείο (structuring element).
  - Είναι χρήσιμο για το κλείσιμο μικρών οπών μέσα σε αντικείμενα ή μικρών μαύρων σημείων πάνω σε αντικείμενο

- **Μορφολογικό closing** είναι μια διαστολή (dilation) ακολουθούμενη από μια συστολή (erosion) με το ίδιο δομικό στοιχείο (structuring element).
  - Είναι χρήσιμο για την απομάκρυνση θορύβου

Μορφολογικοί μετασχηματισμοί

Διαστολή (Dilation) – παράδειγμα στη θεωρία

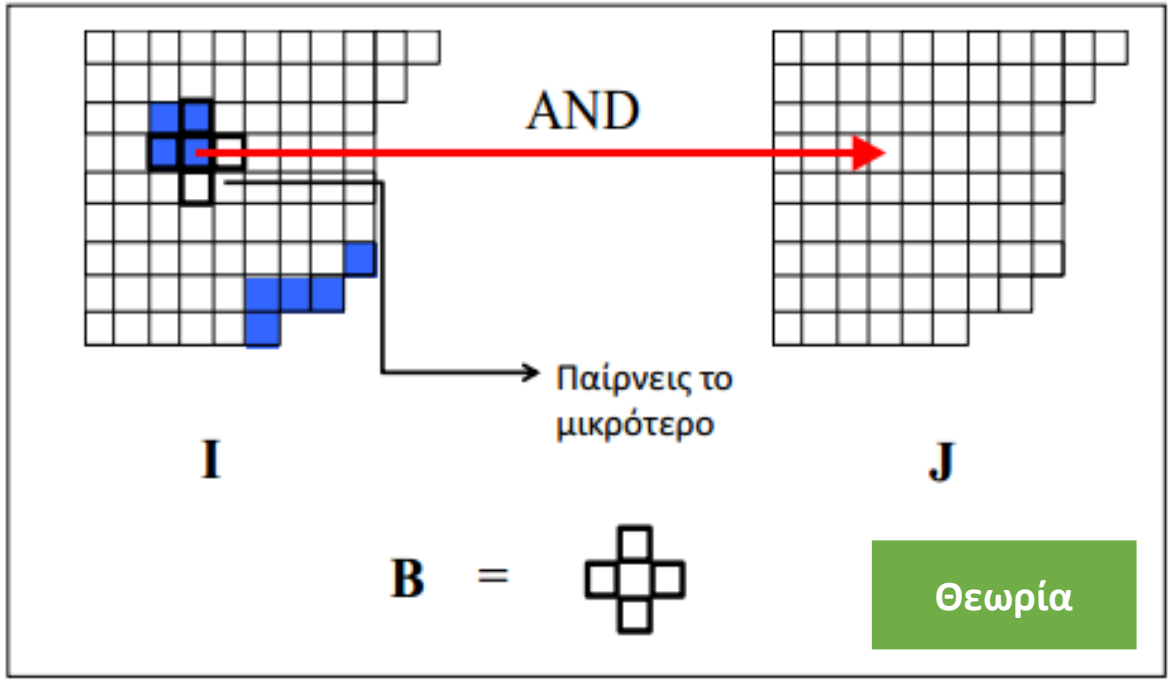


= λογικό “1”
= λογικό “0”
= λογικό “1” που άλλαξε λόγω διαστολής

Στη θεωρία το 0 είναι το λευκό και το 1 το μαύρο. Δηλαδή τα αντικείμενα είναι μαύρα. Στο OpenCV το 0 είναι μαύρο και το 255 το λευκό. Δηλαδή τα αντικείμενα είναι λευκά

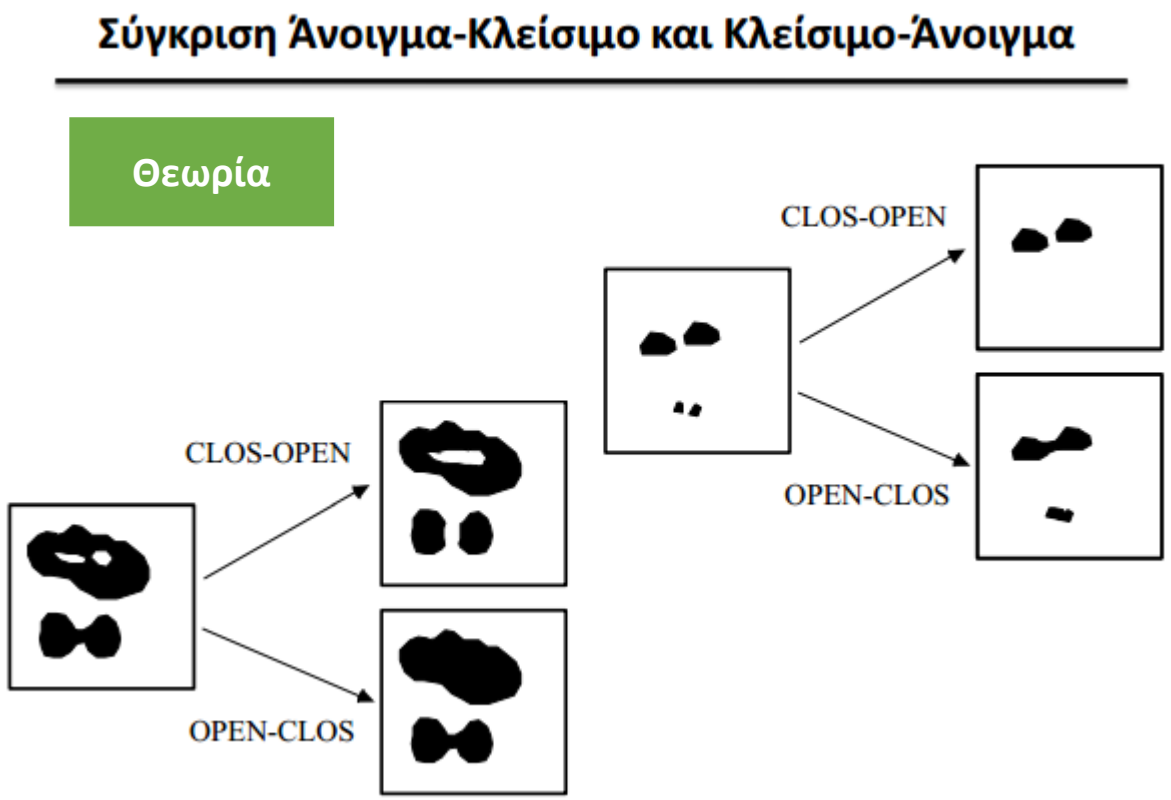
Μορφολογικοί μετασχηματισμοί

Συστολή – παράδειγμα στη θεωρία



Μορφολογικοί μετασχηματισμοί

- Άνοιγμα(Opening) = Συστολή(erosion) + Διαστολή(dilation)
- Κλείσιμο(Closing) = Διαστολή(dilation) + Συστολή(erosion)



Δομικά στοιχεία στο OpenCV

- Δημιουργία δομικού στοιχείου: cv2.getStructuringElement()
- Επιλογή δομικού στοιχείου από:
  - Rectangle: cv2.MORPH_RECT
  - Ellipse: cv2.MORPH_ELLIPSE
  - Cross: cv2.MORPH_CROSS
- Καθορισμός διαστάσεων

Παραδείγματα δημιουργίας δοκιμών στοιχείων

```
# Rectangular Kernel
>>> cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]], dtype=uint8)

# Elliptical Kernel
>>> cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(5,5))
array([[0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0]], dtype=uint8)

# Cross-shaped Kernel
>>> cv2.getStructuringElement(cv2.MORPH_CROSS,(5,5))
array([[0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0]], dtype=uint8)
```

Μορφολογικοί μετασχηματισμοί στο OpenCV

- Συστολή: cv2.erode()
- Διαστολή: cv2.dilate()
- Άλλοι μορφολογικοί μετασχηματισμοί: cv2.morphologyEx()
  - Opening: cv2.MORPH_OPEN
  - Closing: cv2.MORPH_CLOSE

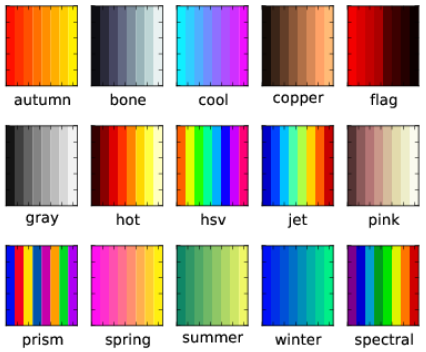
Συνήθως εκτελούνται σε δυαδικές εικόνες

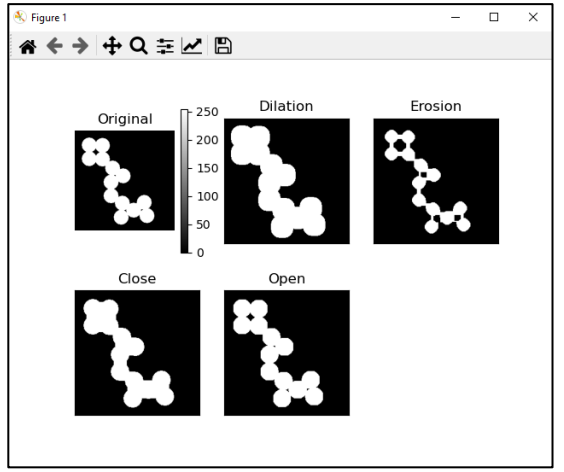
Παράδειγμα στην Python

```
# Import libraries
import cv2
from matplotlib import pyplot as plt
# Read the image as greyscale
img = cv2.imread('circles.png',0)
# Define a structuring element (kernel)
kernel = cv2.getStructuringElement(cv2.MORPH_RECT,(9,9))
# Apply morphological transformations to the image
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
erosion = cv2.erode(img,kernel,iterations = 1)
dilation = cv2.dilate(img,kernel,iterations = 1)
# Show images in subplots
plt.subplot(231), plt.imshow(img,'gray'), plt.title('Original')
plt.axis('off')
plt.colorbar() # show the color bar next to the subplot
plt.subplot(232), plt.imshow(dilation,'gray'), plt.title('Dilation')
plt.xticks([]), plt.yticks([])  # to hide tick values on X and Y axis
plt.subplot(233), plt.imshow(erosion,'gray'), plt.title('Erosion')
plt.xticks([]), plt.yticks([])
plt.subplot(234), plt.imshow(closing,'gray'), plt.title('Close')
plt.xticks([]), plt.yticks([])
plt.subplot(235), plt.imshow(opening,'gray'), plt.title('Open')
plt.xticks([]), plt.yticks([])
plt.show()
```

ή plt.axis('off')

color map





Διαδική εικόνα στο OpenCV

- Μετατροπή εικόνας σε grayscale (μαυρόασπρη):
  - img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
- Μετατρέπει τη μαυρόασπρη εικόνα σε δυαδική με βάση ένα κατώφλι (threshold) T:
  - cv2.threshold(img_gray, T, maxValue, cv2.THRESH_BINARY)

$$\texttt{dst}(x,y) = \begin{cases} \texttt{maxValue} & \text{if } \texttt{src}(x,y) > T \\ 0 & \text{otherwise} \end{cases}$$

Οι τιμές φωτεινοτήτων πάνω από το κατώφλι T γίνονται ίσες με 255 (λευκό χρώμα) και οι τιμές φωτεινοτήτων κάτω από το κατώφλι T γίνονται ίσες με 0 (μαύρο χρώμα). Στο OpenCV/python εάν ένα στοιχείο της εικόνας έχει τιμή ίση με το κατώφλι, τότε η νέα τιμή θα είναι 0 και όχι maxValue.

Παράδειγμα στην Python

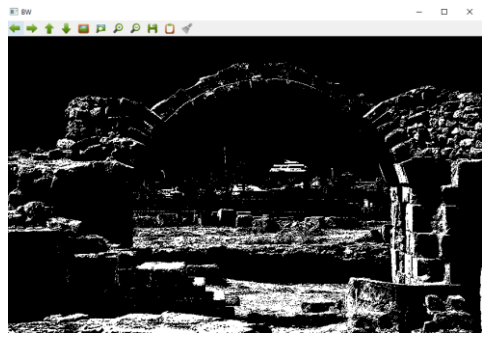
```
# Import libraries
import cv2
from google.colab.patches import cv2_imshow
# Read image
img = cv2.imread('cyprus.jpg')

# Convert image to grayscale
img_gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
# Convert image to binary using a threshold
(thresh, img_bw) = cv2.threshold(img_gray, 200, 255, cv2.THRESH_BINARY)

# Show images
print('Image'); cv2_imshow(img)
print('Grayscale'); cv2_imshow(img_gray)
print('Binary'); cv2_imshow(img_bw)
```

maxValue

Δοκιμάστε να αλλάξετε το κατώφλι από 200 με οποιαδήποτε άλλη τιμή μεταξύ 0-255 και σχολιάστε το αποτέλεσμα



Εμφάνιση έγχρωμης εικόνας χρησιμοποιώντας τη βιβλιοθήκη matplotlib

```
# Import libraries
import cv2
from matplotlib import pyplot as plt
# Read image
img = cv2.imread('cyprus.jpg’)

# Show images in subplots
plt.subplot(121),plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB)),
plt.title('Original RGB')
plt.axis(‘off’)
plt.subplot(122),plt.imshow(im), plt.title('Original BGR')
plt.axis(‘off’)
plt.show()
```

Η OpenCV αναπαριστά τις έγχρωμες εικόνες ως πολυδιάστατους πίνακες NumPy, αλλά με αντίστροφη σειρά καναλιών.
Αυτό σημαίνει ότι οι εικόνες αποθηκεύονται στη σειρά **BGR** αντί για **RGB**.