



Πανεπιστήμιο Πειραιώς
University of Piraeus

Εργασία μαθήματος «Συστήματα Πολυμέσων»
Εαρινό εξάμηνο ακαδημαϊκού έτους 2022-2023

Μέλη:

Κούκη Τζανέτο, Π20097

Σπυροπούλου Κανέλλα, Π20179

Φραγκογιάννης Διονύσιος, Π20198

Θέμα 1:

a.

```
import cv2

# Encoder
input_video_path =
'C:/Users/kuqit/PycharmProjects/Systems/video/input_video.mp4' #
Ορίζουμε τη διαδρομή του εισερχόμενου βίντεο
output_video_path =
'C:/Users/kuqit/PycharmProjects/Systems/video/output_video.mp4' #
Ορίζουμε τη διαδρομή του εξερχόμενου βίντεο

cap = cv2.VideoCapture(input_video_path) # Δημιουργούμε ένα
αντικείμενο για το διάβασμα του εισερχόμενου βίντεο
fps = cap.get(cv2.CAP_PROP_FPS) # Παίρνουμε τα καρέ ανά δευτερόλεπτο
του βίντεο
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) # Παίρνουμε το πλάτος
των frame του βίντεο
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) # Παίρνουμε το ύψος
των frame του βίντεο

# Δημιουργούμε ένα αντικείμενο VideoWriter για το εξερχόμενο βίντεο
fourcc = cv2.VideoWriter_fourcc(*"MJPG")
out = cv2.VideoWriter(output_video_path, fourcc, fps, (width, height),
isColor=True)

ret, frame1 = cap.read()
out.write(frame1) # Γράφουμε το πρώτο I-frame

while True:
    ret, frame = cap.read()
    if not ret:
        break

    if frame.shape != frame1.shape:
        frame = cv2.resize(frame, (frame1.shape[1], frame1.shape[0]))

    # Υπολογίζουμε και εμφανίζουμε το frame του σφάλματος
    error_frame = cv2.absdiff(frame, frame1)
    cv2.imshow("Error Frame", error_frame)
    cv2.waitKey(1)

    # Κωδικοποιούμε το frame του σφάλματος (δεν είναι απαραίτητο για
    απώλειας ασυμπίεστης συμπίεσης)

    # Ενημερώνουμε το προηγούμενο frame
    frame1 = frame
```

```

        # Γράφουμε το τρέχον frame στο εξερχόμενο βίντεο
        out.write(frame)

cap.release()
out.release()
cv2.destroyAllWindows()

# Decoder
encoded_video_path =
'C:/Users/kuqit/PycharmProjects/Systems/video/output_video.mp4' #
Ορίζουμε τη διαδρομή του κωδικοποιημένου βίντεο
decoded_output_path =
'C:/Users/kuqit/PycharmProjects/Systems/video/decoded_video.mp4' #
Ορίζουμε τη διαδρομή του αποκωδικοποιημένου βίντεο

cap = cv2.VideoCapture(encoded_video_path) # Δημιουργούμε ένα
αντικείμενο για το διάβασμα του κωδικοποιημένου βίντεο
fps = cap.get(cv2.CAP_PROP_FPS) # Παίρνουμε τα καρέ ανά δευτερόλεπτο
του βίντεο
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) # Παίρνουμε το πλάτος
των frame του βίντεο
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) # Παίρνουμε το ύψος
των frame του βίντεο

# Δημιουργούμε ένα αντικείμενο VideoWriter για το αποκωδικοποιημένο
βίντεο
fourcc = cv2.VideoWriter_fourcc(*"MJPG")
out = cv2.VideoWriter(decoded_output_path, fourcc, fps, (width,
height), isColor=False)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Αποκωδικοποιούμε το frame (δεν είναι απαραίτητο για απώλειας
    ασυμπίεστης συμπίεσης)

    # Εμφανίζουμε το αποκωδικοποιημένο frame
    cv2.imshow("Decoded Frame", frame)
    cv2.waitKey(1)

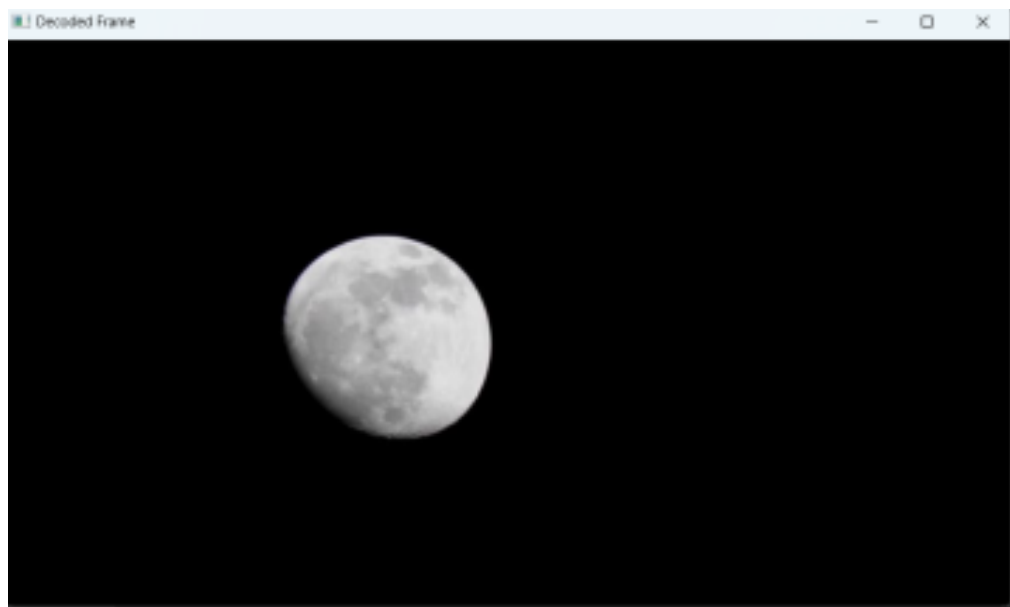
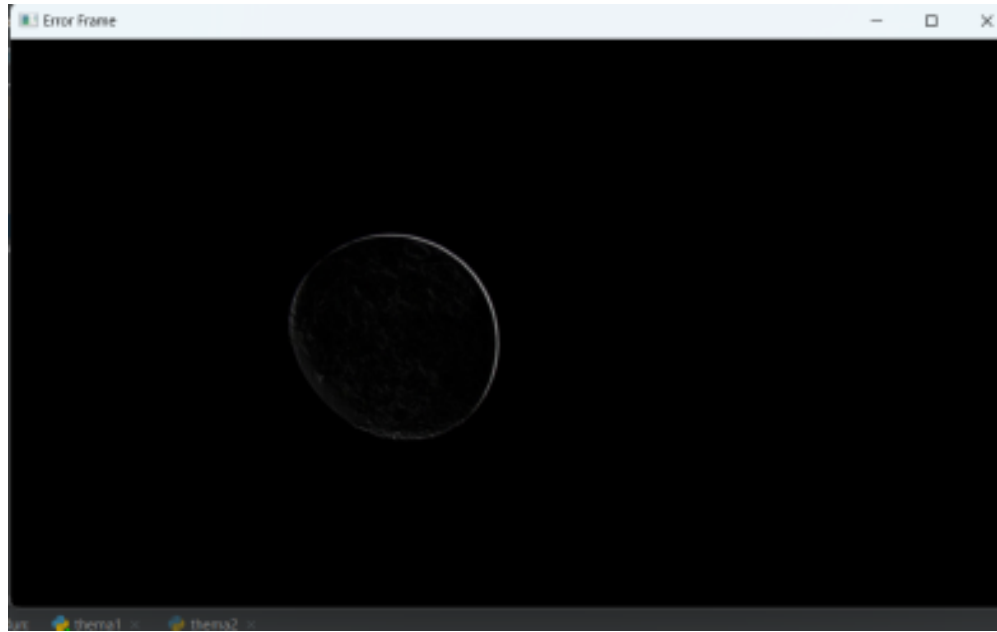
    # Γράφουμε το αποκωδικοποιημένο frame στο εξερχόμενο βίντεο
    out.write(frame)

cap.release()
out.release()

```

```
cv2.destroyAllWindows()
```

Output:



Θέμα 2:

```
import cv2
import numpy as np

# Διαδρομή προς το αρχείο εισόδου και εξόδου του βίντεο
input_video_path =
'C:/Users/kuqit/PycharmProjects/Systems/video/input_video.mp4'
output_video_path =
'C:/Users/kuqit/PycharmProjects/Systems/video/output_video.mp4'

# Άνοιγμα του αρχείου εισόδου βίντεο
cap = cv2.VideoCapture(input_video_path)

# Λήψη των καρέ ανά δευτερόλεπτο (fps), πλάτους και ύψους του βίντεο
fps = cap.get(cv2.CAP_PROP_FPS)
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

# Ορισμός του κωδικοποιητή βίντεο και δημιουργία αντικειμένου
VideoWriter για το εξαγωγή βίντεο
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(output_video_path, fourcc, fps, (width, height))

# Ανάγνωση του πρώτου καρέ του βίντεο
ret, frame = cap.read()

# Επιλογή περιοχής ενδιαφέροντος (ROI) για την παρακολούθηση
roi = cv2.selectROI(frame, False)
cv2.destroyAllWindows()

# Εξαγωγή του χρώματος της ROI για παρακολούθηση
roi_frame = frame[roi[1]:roi[1] + roi[3], roi[0]:roi[0] + roi[2]]

# Έναρξη επεξεργασίας των καρέ του βίντεο
while True:
    # Ανάγνωση του επόμενου καρέ από το βίντεο
    ret, frame = cap.read()

    # Εάν δεν υπάρχουν περισσότερα καρέ, διακοπή της επανάληψης
    if not ret:
        break

    # Μετατροπή του καρέ στον χώρο χρωμάτων HSV για ανίχνευση του
    χρώματος
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    hsv_roi_frame = cv2.cvtColor(roi_frame, cv2.COLOR_BGR2HSV)
```

```

    # Ορισμός του εύρους χρώματος για την παρακολούθηση (γκρι
    αποχρώσεις)
    lower_grey = np.array([0, 0, 50]) # Κάτω όριο γκρι απόχρωσης
    upper_grey = np.array([179, 50, 200]) # Άνω όριο γκρι απόχρωσης

    # Δημιουργία μάσκας για την ανίχνευση του γκρι χρώματος εντός του
    καθορισμένου εύρους
    mask = cv2.inRange(hsv_frame, lower_grey, upper_grey)

    # Ανεύρεση των περιγράμμάτων του εντοπισμένου γκρι χρώματος στη
    μάσκα
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)

    if len(contours) > 0:
        # Εύρεση του μεγαλύτερου περιγράμματος (του παρακολουθούμενου
        αντικειμένου)
        largest_contour = max(contours, key=cv2.contourArea)

        # Λήψη του περιβλήματος περιοχής του μεγαλύτερου περιγράμματος
        x, y, w, h = cv2.boundingRect(largest_contour)

        # Ενημέρωση της θέσης του πλαισίου ROI
        roi = (x, y, w, h)
        roi_frame = frame[y:y + h, x:x + w]

        # Διαγραφή της περιοχής ενδιαφέροντος χρησιμοποιώντας την
        ενημερωμένη θέση
        mask = np.zeros_like(frame)
        mask[roi[1]:roi[1] + roi[3], roi[0]:roi[0] + roi[2]] = 255

        # Εφαρμογή μορφολογικών πράξεων για την κλείσιμο των κενών στη
        μάσκα
        kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (25, 25))
        mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
        mask = cv2.GaussianBlur(mask, (35, 35), 0)

        # Αναπλήρωση του εντοπισμένου αντικειμένου χρησιμοποιώντας τη μάσκα
        frame = cv2.inpaint(frame, mask[:, :, 0], 3, cv2.INPAINT_NS)

        # Γράψιμο του τροποποιημένου καρέ στο αρχείο εξόδου
        out.write(frame)

        # Εμφάνιση του τροποποιημένου καρέ του βίντεο
        cv2.imshow('Motified Video', frame)

        # Εάν πατηθεί το πλήκτρο 'q', έξοδος από την επανάληψη
        if cv2.waitKey(1) & 0xFF == ord('q'):

```

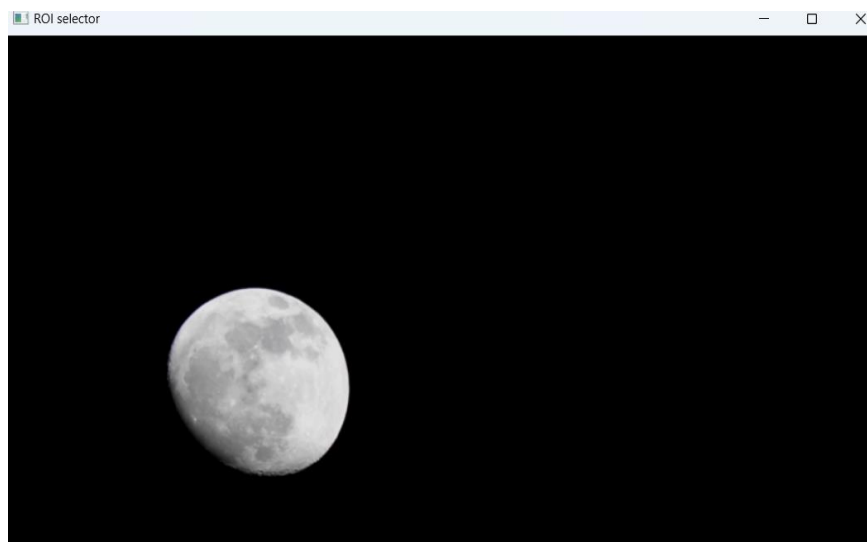
```
break

# Απελευθέρωση του αντικειμένου ανάγνωσης και εγγραφής βίντεο
cap.release()
out.release()

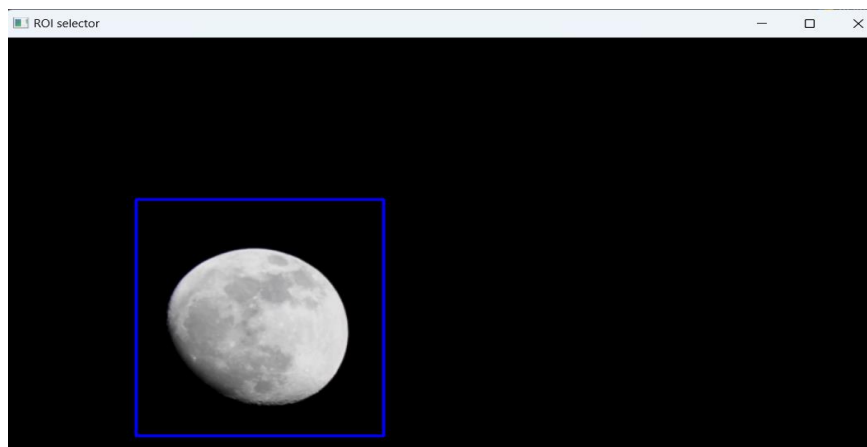
# Κλείσιμο όλων των ανοιχτών παραθύρων
cv2.destroyAllWindows()
```

Output:

Αρχικό βίντεο



Επιλογή του αντικειμένου που επιθυμούμε να αφαιρέσουμε



Τελικό βίντεο

