

## Εργασία Πολυμέσων (Εαρινό 2024–2025)

Ονοματεπώνυμο: Στυλιανός Μύαρης

A.M.: Π22114

### Μέρος Α – Πρόβλεψη με τον απλό τρόπο

Κάθε επόμενο καρέ είναι ίδιο με το προηγούμενο, δηλαδή:

$$I(n+1)=I(n)$$

Για να δούμε τις αλλαγές, υπολογίζουμε τα διαφορικά καρέ (residuals):

$$D(n+1)=| I(n+1)-I(n) |$$

Δηλαδή πήρα τα καρέ του βίντεο, τα μετέτρεψα σε grayscale (**`cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`**), και αφαίρεσα το καθένα από το προηγούμενο (**`diff = gray - prev`**). Όπου δεν υπάρχει αλλαγή, η εικόνα είναι μαύρη, αλλιώς τα πίξελ γίνονται άσπρα (**`cv2.absdiff(cur, prev)`**).

Οι αλλαγές απεικονίζονται στα διαγράμματα, τα οποία μετρούν το ποσοστό των πίξελς που αλλάζουν σε κάθε καρέ (**`np.sum(D > T)`**) και στο διάγραμμα MAD – Mean Absolute Difference (**`np.mean(np.abs(cur - prev))`**). Μαζί με 3 φωτογραφίες που φαίνονται οι διαφορές (μπορείτε να αλλάξετε τον αριθμό μέχρι και 115 που είναι τα συνολικά καρέ του βίντεο) βρίσκονται σε έναν φάκελο που δημιουργεί ο κώδικας.

### Μέρος Β – Κωδικοποίηση Huffman

Από τα καρέ, μάζεψα όλες τις τιμές των pixels και έφτιαξα το ιστόγραμμα συχνοτήτων (**`Counter(abs_flat.tolist())`**).

- Η τεράστια πλειοψηφία ήταν 0 (καμία αλλαγή).
- Λίγο παραπάνω είναι μικρότερα νούμερα (1, 2, 3...) – που σημαίνουν μικρές αλλαγές.
- Οι μεγάλες τιμές σπάνια εμφανίζονται.

Τι γίνεται με την κωδικοποίηση Huffman (**`build_huffman_code(freq_abs)`**):

- Δίνει μικρούς κωδικούς (λίγα bits) στα συχνά σύμβολα, όπως το 0.
- Δίνει μεγάλους κωδικούς στα σπάνια.

Αποτελέσματα (υπολογισμένα με **`bit_length_sum(freq_abs, code_abs)`**):

- Αρχικό μέγεθος : 230.850 KB

- Με Huffman: 84.903 KB
- Λόγος συμπίεσης (CR):  $\times 2.72$

Το αρχείο έγινε σχεδόν 3 φορές μικρότερο.

### Μέρος Γ – Αντιστάθμιση κίνησης

Χρησιμοποίησα block-based motion compensation:

- Χώρισα το καρέ σε blocks 16×16 (*for by in range(0, H - BLOCK + 1, BLOCK)*).
- Για κάθε block στο επόμενο καρέ, έψαξα στο προηγούμενο για να βρω το πιο ταιριαστό block μέσα σε ένα παράθυρο  $\pm 8$  pixels (*cv2.matchTemplate(win, T, cv2.TM\_SQDIFF)*).
- Έτσι πήρα το “διάνυσμα κίνησης” (dx,dy) για κάθε block (*dx\_map[...] = dx, dy\_map[...] = dy*).

Με αυτά έφτιαξα το καρέ πρόβλεψης (*pred[...] = prev[by+dy:..., bx+dx:...]*). Το αποτέλεσμα είναι πολύ πιο κοντά στο πραγματικό, οπότε τα residuals είχαν περισσότερα μηδενικά (*cv2.absdiff(cur, pred)*).

Στη συνέχεια:

- Κωδικοποίησα τα residuals με Huffman (*build\_huffman\_code(freq\_res)*).
- Κωδικοποίησα τα διανύσματα κίνησης (dx,dy) επίσης με Huffman (*build\_huffman\_code(freq\_dx)* και *build\_huffman\_code(freq\_dy)*), μιας και τα περισσότερα είναι 0.

Αποτελέσματα:

- Residual bits (Huffman): 57.413 KB
- Motion vectors: 682 KB
- Σύνολο: 58.095 KB
- Baseline (raw residuals 8bpp): 230.850 KB
- Λόγος συμπίεσης (CR):  $\times 3.97$

Η συμπίεση βελτιώθηκε σε σχέση με το Μέρος Β (από 2.72 σε 3.97).

Ο κώδικας είναι χωρισμένος σε 3 αρχεία (ένα για κάθε ερώτημα) και εξοπλισμένος με επαρκή σχόλια. Παρακάτω παραθέτω παράδειγμα εκτέλεσης των αρχείων

## Ερώτημα 1

### Τερματικό:

```
%runfile C:/Users/user/.spyder-py3/mymulti.py --wdir
```

```
[ WARN:0@1.790] global cap_gstreamer.cpp:2824 cv::handleMessage OpenCV | GStreamer warning: your GStreamer installation is missing a required plugin: Audio Video Interleave (AVI) demuxer
```

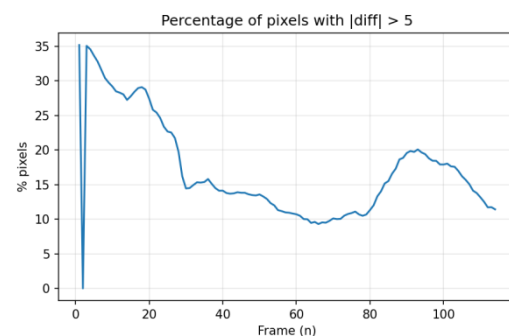
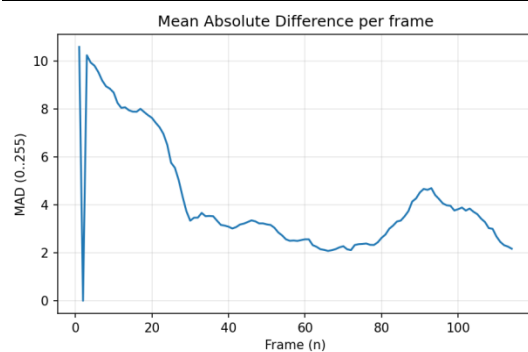
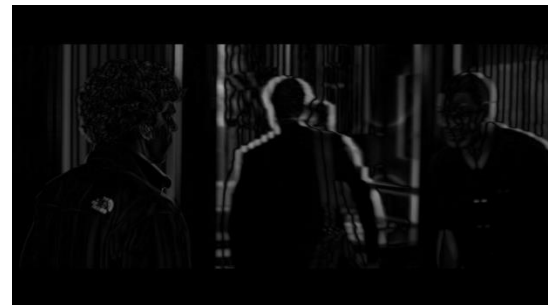
```
[ WARN:0@1.800] global cap_gstreamer.cpp:2840 cv::handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module uridecodebin0 reported: Your GStreamer installation is missing a plug-in.
```

```
[ WARN:0@1.806] global cap_gstreamer.cpp:1698 cv::GStreamerCapture::open OpenCV | GStreamer warning: unable to start pipeline
```

```
[ WARN:0@1.807] global cap_gstreamer.cpp:1173 cv::GStreamerCapture::isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
```

Photos and plots' directory: C:\Users\user\.spyder-py3\diff\_outputs

Φάκελος εικόνων (περιέχει 3 τυχαίες εικόνες και τα διαγράμματα):



## Ερώτημα 2 (χωρίς προβολή του Huffman)

%runfile C:/Users/user/.spyder-py3/mymulti2.py --wdir

Frames: 115 total, diff-frames: 114, size: 1920x1080, pixels: 236390400

[ABS] Raw: 230850.00 KB | Compressed: 84903.36 KB | CR=2.72x

[SIGNED] Raw(9b): 259706.25 KB | Compressed: 98142.45 KB | CR=2.65x

Do you want to see the Huffman codes? (y/n): n

End.

## Ερώτημα 3 (χωρίς προβολή του Huffman)

%runfile C:/Users/user/.spyder-py3/mymulti3.py --wdir

Frames total: 115, diff-frames: 114, size: 1920x1080

Residual bits (Huffman): 57413.37 KB

MV bits (huffman): 681.63 KB

Total bitstream: 58095.00 KB

Baseline (raw residuals 8bpp): 230850.00 KB

Compression Ratio (CR): 3.97x

Do you want to see the Huffman codes? (y/n): n

End.