

**Πανεπιστήμιο Πειραιώς  
Τμήμα Πληροφορικής  
Έτος: 2022 - 2023**



**Μάθημα:  
«ΣΥΣΤΗΜΑΤΑ-ΠΟΛΥΜΕΣΩΝ»  
Εργασία: Υπολογιστική Εργασία Μαθήματος  
Εξάμηνο: 6ο**

**Ομάδα εργασίας:**  
Αιμιλιανός Κουρπάς-Δανάς Π20100,  
Αναστάσιος Μελαχροινούδης Π20124

**Ημερομηνία παράδοσης : 05.07.2023**

# Περιεχόμενα

<b>ΘΕΜΑ 1ο:</b>	<b>3</b>
1.1 Εκφώνηση:	3
1.2 Λύση:	3
1.2.1 Αλγοριθμική Περιγραφή:	3
1.2.2 Υλοποίηση Προγράμματος:	4
1.4 Παράδειγμα:	6
<b>ΘΕΜΑ 2ο:</b>	<b>8</b>
2.1 Εκφώνηση:	8
2.2 Λύση:	8
2.2.1 Αλγοριθμική Περιγραφή:	8
2.2.2 Υλοποίηση Προγράμματος:	9
2.3 Το τεχνικό κομμάτι:	10
2.4 Παράδειγμα:	10
<b>ΘΕΜΑ 3ο:</b>	<b>12</b>
3.1 Εκφώνηση:	12
3.2 Λύση:	12
3.2.1 Αλγοριθμική Περιγραφή:	12
3.2.2 Υλοποίηση Προγράμματος:	13
3.4 Παράδειγμα:	14
<b>Βιβλιογραφία Θεμάτων:</b>	<b>15</b>

## Σημείωση

Τα προγράμματα γράφτηκαν στο προγραμματιστικό περιβάλλον VScode.

Στον κώδικα υπάρχουν περιεκτικά σχόλια ουσίας για την καλύτερη επεξήγησή του.

## **ΘΕΜΑ 1ο:**

### **1.1 Εκφώνηση:**

Έστω video της επιλογής σας διάρκειας 5 s – 15 s. Υποθέστε ότι το Frame 1 είναι πάντα 1 frame και ότι τα επόμενα πλαίσια είναι P frames.

Κάθε πλαίσιο P προβλέπεται χωρίς αντιστάθμιση κίνησης από το προηγούμενο πλαίσιο. Υπολογίστε και απεικονίστε την ακολουθία εικόνων σφάλματος και κωδικοποιήστε την χωρίς απώλειες. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

### **1.2 Λύση:**

#### **1.2.1 Αλγοριθμική Περιγραφή:**

Για τις ανάγκες της άσκησης χρησιμοποιήθηκε ένα βίντεο 10 δευτερολέπτων. Το βίντεο ονομάζεται videoThema1i.mp4

Αρχικά, φορτώνουμε το αρχικό Frame του βίντεο. Στη συνέχεια, εκτελούμε τη διαδικασία της αφαίρεσης για να βρούμε τα υπόλοιπα Frames. Αφαιρούμε το τρέχον Frame από το προηγούμενο Frame και κωδικοποιούμε τη διαφορά αυτής της αφαίρεσης με την τεχνική Huffman. Με αυτόν τον τρόπο δημιουργείται το κωδικοποιημένο βίντεο με την ονομασία "encoded.avi". Για να ανακατασκευάσουμε το σήμα, εκτελούμε την αντίστροφη διαδικασία. Αποκωδικοποιούμε κάθε Frame και προσθέτουμε το τρέχον Frame στο προηγούμενο του. Ωστόσο, ο αποκωδικοποιητής παρουσιάζει σφάλματα σε κάθε Frame, τα οποία συσσωρεύονται με κάθε βήμα. Τέλος, καταλήγουμε σε ένα ανακατασκευασμένο βίντεο με θόρυβο και έλλειψη λεπτομερειών.

## **Huffman**

Για να κωδικοποιήσουμε την ακολουθία εικόνων σφάλματος, πρώτα μετατρέπουμε τον δισδιάστατο πίνακα με τις τιμές κάθε εικόνας σφάλματος σε έναν μονοδιάστατο πίνακα. Στη συνέχεια, υπολογίζουμε τη συχνότητα εμφάνισης κάθε συμβόλου στον πίνακα. Έπειτα, χρησιμοποιούμε τον κωδικοποιητή για να καλέσουμε τη συνάρτηση "encode" με όρισμα την εικόνα σφάλματος που θέλουμε να κωδικοποιήσουμε. Η αντίστοιχη διαδικασία εφαρμόζεται κατά την αποκωδικοποίηση του κωδικοποιημένου Frame, αλλά αντί να χρησιμοποιήσουμε τη συνάρτηση "encode", χρησιμοποιούμε τη συνάρτηση "decode" του κωδικοποιητή (codec.decode()).

### 1.2.2 Υλοποίηση Προγράμματος

Γλώσσα Υλοποίησης: Python

Βιβλιοθήκες :

- **collections**: Αυτή η βιβλιοθήκη χρησιμοποιείται για τη δημιουργία ενός μετρητή για τον υπολογισμό της συχνότητας των στοιχείων σε μια ακολουθία.
- **cv2 (OpenCV)**: Αυτή η βιβλιοθήκη παρέχει λειτουργίες υπολογιστικής όρασης και επεξεργασίας εικόνας. Χρησιμοποιείται για την ανάγνωση των Frame του βίντεο, τη μετατροπή των Frame σε γκρι, και την αποθήκευση των βίντεο σε αρχεία.
- **pickle**: Αυτή η βιβλιοθήκη χρησιμοποιείται για την αποθήκευση και τη φόρτωση αντικειμένων Python. Σε αυτόν τον κώδικα, χρησιμοποιείται για την αποθήκευση και φόρτωση του πίνακα κωδικοποίησης Huffman.
- **heapq**: Αυτή η βιβλιοθήκη παρέχει λειτουργίες στοίβας. Χρησιμοποιείται για τη δημιουργία μιας προτεραιότητας ουράς συμβόλων κατά την κωδικοποίηση Huffman.
- **pathlib**: Αυτή η βιβλιοθήκη παρέχει μια αντικειμενοστραφή διεπαφή για την χειρισμό των διαδρομών του αρχείου στο σύστημα αρχείων. Χρησιμοποιείται για την διαχείριση των διαδρομών αρχείων με έναν τρόπο που είναι ανεξάρτητος από την πλατφόρμα.

Κλάση:

- **HuffmanCodec**: υλοποιεί τον αλγόριθμο κωδικοποίησης και αποκωδικοποίησης Huffman. Έχει μεθόδους για τη δημιουργία ενός κωδικοποιητή Huffman από δεδομένα ή συχνότητες, την κωδικοποίηση δεδομένων με τον κωδικοποιητή, την αποκωδικοποίηση δεδομένων με τον κωδικοποιητή, και την αποθήκευση/φόρτωση του πίνακα κωδικοποίησης.

### Συναρτήσεις:

- **encode\_video**: διαβάζει ένα αρχείο βίντεο καρέ-προς-καρέ, μετατρέπει κάθε καρέ σε κλίμακα του γκρι, υπολογίζει τη διαφορά μεταξύ συνεχόμενων καρέ, κωδικοποιεί τις διαφορές χρησιμοποιώντας την κωδικοποίηση Huffman, και γράφει τα κωδικοποιημένα δεδομένα σε ένα αρχείο βίντεο εξόδου.
- **decode\_video**: διαβάζει ένα κωδικοποιημένο αρχείο βίντεο, αποκωδικοποιεί τα δεδομένα χρησιμοποιώντας την αποκωδικοποίηση Huffman, ανακατασκευάζει τα Frame του βίντεο και γράφει το αποκωδικοποιημένο βίντεο σε ένα αρχείο εξόδου.

### Μεταβλητές:

- **input\_path**: το path του αρχείου εισόδου.
- **output**: το path του αρχείου εξόδου.
- **video**: το αντικείμενο VideoCapture για την ανάγνωση του εισερχόμενου βίντεο.
- **success**: ανάγνωση του βίντεο (Boolean).
- **diff**: τη διαφορά μεταξύ του τρέχοντος Frame και του προηγούμενου Frame.
- **symb2freq**: αντιστοίχιση συμβόλου-συχνότητας για τον πίνακα διαφοράς.
- **codec**: το αντικείμενο HuffmanCodec για την κωδικοποίηση και αποκωδικοποίηση.

## 1.3 Το τεχνικό κομμάτι

### Οδηγίες για εκτέλεση του προγράμματος:

- Χρειάζεται να έχουμε εγκατεστημένη την python
- Χρειάζεται να έχουμε εγκατεστημένη την cv2 όπου γίνεται με την εντολή στο terminal `pip install opencv-python`
- Δεν χρειάζεται να κατεβάσουμε τις υπόλοιπες βιβλιοθήκες γιατί περιέχονται στην Python
- Βρισκουμε το terminal στο path που βρίσκεται το Thema1i.py , και γράφετε `python3 Thema1i.py`

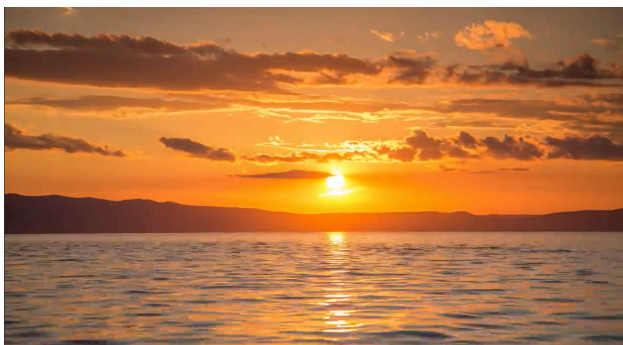
## 1.4 Παράδειγμα

Θα τρέξουμε ένα video το οποίο είναι timelapse ηλιοβασίλεμα

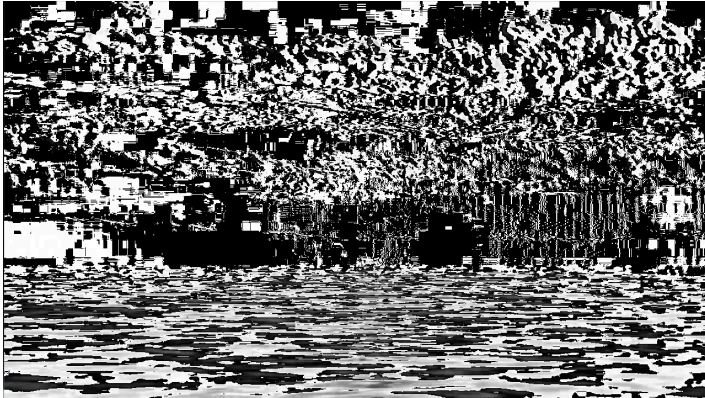
### Terminal:

```
Encoding Started!  
Video released as 'encoding.avi'!  
Video encoding completed!  
Video decoding has started  
Video released as 'decoded.avi'!  
Video decoding completed!
```

### ScreenShot in videoThema1i.mp4:



**ScreenShot in encoded.avi:**



**ScreenShot in decoded.avi:**



**Την βιβλιογραφία για το πρώτο θέμα θα το βρείτε  
στο τέλος του τρίτου θέματος.**

## **ΘΕΜΑ 2ο:**

### **2.1 Εκφώνηση:**

Έστω ασυμπίεστο video της επιλογής σας, διάρκειας 5 s – 15 s.

Υποθέστε ότι το Frame 1 είναι πάντα 1 frame και ότι τα επόμενα πλαίσια είναι P frames.

Υλοποιήστε την τεχνική αντιστάθμισης κίνησης για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε macroblocks 64x64, ακτίνα αναζήτησης  $k=32$  και τεχνική σύγκρισης macroblocks της επιλογής σας. Να επιταχυνθεί η διαδικασία υλοποιώντας ιεραρχική αναζήτηση. Υπολογίστε, αποθηκεύστε και απεικονίστε την ακολουθία εικόνων πρόβλεψης και εικόνων ασφαμάτων. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

### **2.2 Λύση:**

#### **2.2.1 Αλγοριθμική Περιγραφή**

Για να επιλύσουμε την άσκηση, χρησιμοποιούμε ένα βίντεο διάρκειας 10 δευτερολέπτων, το βίντεο ονομάζεται videoThema1\_2.mp4.

Αρχικά, μετατρέπουμε κάθε frame του βίντεο σε αποχρώσεις του γκρι. Έπειτα, βρίσκουμε τα επίπεδα για το προηγούμενο και το τρέχον frame. Στη συνέχεια, διαιρούμε το frame σε macroblocks και υπολογίζουμε την κινητικότητα σε κάθε macroblock. Αφού βρούμε την κινητικότητα στο πιο χαμηλό επίπεδο, ελέγχουμε επίσης την κινητικότητα στα ανώτερα επίπεδα και τα αποθηκεύουμε. Τέλος, αποθηκεύουμε την εικόνα ασφαμάτων. Για την αποκωδικοποίηση, μετατρέπουμε κάθε frame του βίντεο ασφαμάτων σε macroblocks και εισάγουμε το διάνυσμα κίνησης που αντιστοιχεί σε κάθε ένα από αυτά. Έπειτα, μετατρέπουμε το νέο macroblock με τα διανύσματα κίνησης πίσω σε frame και επαναλαμβάνουμε αυτήν τη διαδικασία μέχρι το τέλος του βίντεο



## 2.2.2 Υλοποίηση Προγράμματος

Γλώσσα Υλοποίησης: Python

Βιβλιοθήκες :

- **cv2 (OpenCV)**: χρησιμοποιείται για την ανάγνωση των καρέ του βίντεο, την εμφάνιση εικόνων και τη δημιουργία παραθύρων βίντεο.
- **pickle**: Αυτή η βιβλιοθήκη χρησιμοποιείται για την αποθήκευση και τη φόρτωση αντικειμένων Python
- **tqdm**: Προσφέρει έναν απλό τρόπο για την οπτικοποίηση της προόδου της εργασίας, εμφανίζοντας το εκτιμώμενο χρόνο που απομένει και μια γραμμή προόδου με το ποσοστό ολοκλήρωσης.

Συναρτήσεις:

- 1) **upper\_level**: Παίρνει δύο frames και κάνει αναζήτηση σε αυτά
- 2) **to\_lower\_level**: Μειώνει την ανάλυση των εικόνων και ελέγχει τα macro-blocks
- 3) **hierarchical\_search**: Πραγματοποιεί υποδειγματοληψία σε μια εικόνα
- 4) **has\_motion**: Καθορίζει εάν υπάρχει κίνηση μεταξύ δύο macro-blocks
- 5) **calculate\_sad**: Υπολογίζει το άθροισμα των απόλυτων διαφορών
- 6) **get\_sad**: Συγκρίνει ένα macro-block με τα γειτονικά του macro-blocks
- 7) **create\_image**: Ανακατασκευάζει μια εικόνα από macro-blocks.
- 8) **to\_macro\_blocks**: Διαχωρίζει μια εικόνα σε macro-blocks μεγέθους k.

Σε αυτο το πρόγραμμα έχουμε 3 αρχεία

- encoder1ii.py
- decoder1ii.py
- function.py

## 2.3 Το τεχνικό κομμάτι

- Χρειάζεται να έχουμε εγκατεστημένη την python
- Χρειάζεται να έχουμε εγκατεστημένη την tqdm οπου γινεται με την εντολη στο terminal `pip install tqdm`
- Χρειάζεται να έχουμε εγκατεστημένη την cv2 οπου γινεται με την εντολη στο terminal `pip install opencv-python`
- Βρισκουμε το terminal στο path που βρίσκεται το encode1\_2.py , και γράφετε `python3 encode1_2.py`
- Βρισκουμε το terminal στο path που βρίσκεται το decode1\_2.py , και γράφετε `python3 decode1_2.py`

## 2.4 Παράδειγμα

Terminal:

Encoder

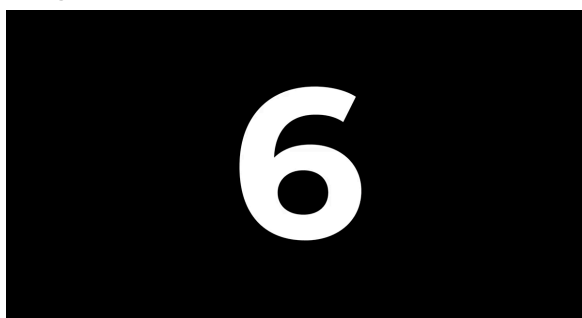
```
Encoding Started!  
Processing frames: 100% | 251/252  
Encoding finished
```

Decoder

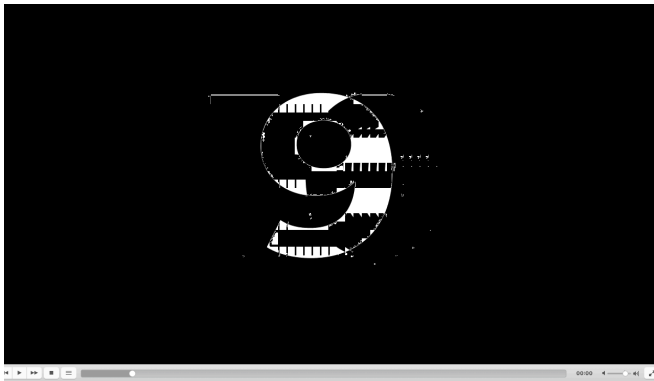
```
Decoding Started!  
Decoding frames: 100% | 252/252  
Decoding finished
```

Video:

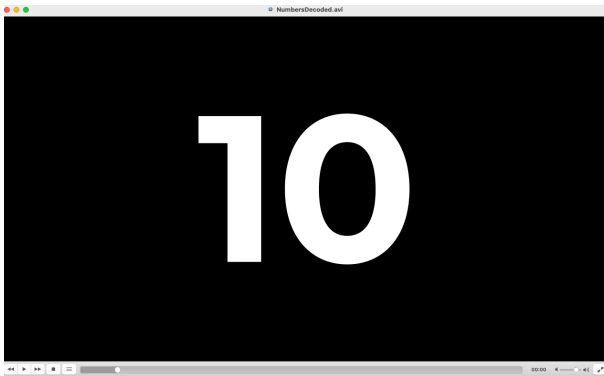
Original :



Encoded:



Decoded:



**Την βιβλιογραφία για το δεύτερο θέμα θα το βρείτε  
στο τέλος του τρίτου θέματος.**

## **ΘΕΜΑ 3ο:**

### **3.1 Εκφώνηση:**

Σε βίντεο της επιλογής σας, διάρκειας 5s – 10s, στο οποίο υπάρχει ήπια κίνηση αντικειμένου και κάμερας, επιλέξτε ένα αντικείμενο και εξαφανίστε το αλγοριθμικά. Δηλαδή, δημιουργήστε και αποθηκεύστε ένα νέο βίντεο στο οποίο δεν θα υπάρχει το αντικείμενο που επιλέξατε. Για τον σκοπό αυτόν, αξιοποιήστε την τεχνική αντιστάθμισης κίνησης. Υλοποιήστε και τεκμηριώστε το σχετικό σύστημα.

### **3.2 Λύση:**

#### **3.2.1 Αλγοριθμική Περιγραφή**

Για τις ανάγκες της άσκησης χρησιμοποιήθηκε ένα βίντεο 7 δευτερολέπτων. Το βίντεο ονομάζεται videoThema2.mp4

Σκοπός της διαδικασίας είναι να απομακρύνουμε την κινούμενη μπάλα στο βίντεο. Αρχικά, επιλέγουμε το πρώτο πλαίσιο του βίντεο επειδή τα διανύσματα κίνησης σε αυτό είναι σχεδόν μηδενικά, λόγω του ότι το παρασκήνιο είναι στατικό. Στη συνέχεια, διαχωρίζουμε το βίντεο σε μακρομπλόκ μεγέθους 64. Κάθε επόμενο πλαίσιο δανείζεται τα μακρομπλόκ του προηγούμενου από τη μέση και κάτω. Αυτό σημαίνει ότι το δεύτερο πλαίσιο παίρνει τα μακρομπλόκ παρασκηνίου του πρώτου, το τρίτο πλαίσιο παίρνει τα μακρομπλόκ παρασκηνίου του δεύτερου, και ούτω καθεξής. Αυτή η διαδικασία οδηγεί στην αφαίρεση των κινούμενων χαρακτήρων από το βίντεο. Έτσι, με τη συγκεκριμένη προσέγγιση, επιτυγχάνουμε τον σκοπό μας να απομακρύνουμε την κινούμενη μπάλα και να διατηρήσουμε μόνο το στατικό παρασκήνιο στο τελικό βίντεο.

### 3.2.2 Υλοποίηση Προγράμματος

Γλώσσα Υλοποίησης: Python

Βιβλιοθήκες :

- **cv2 (OpenCV)**: χρησιμοποιείται για την ανάγνωση των καρέ του βίντεο, την εμφάνιση εικόνων και τη δημιουργία παραθύρων βίντεο.
- **numpy** : χρησιμοποιείται για την επεξεργασία πινάκων και για πράξεις πλαίσιας πάνω στα Frame του βίντεο.

Συναρτήσεις:

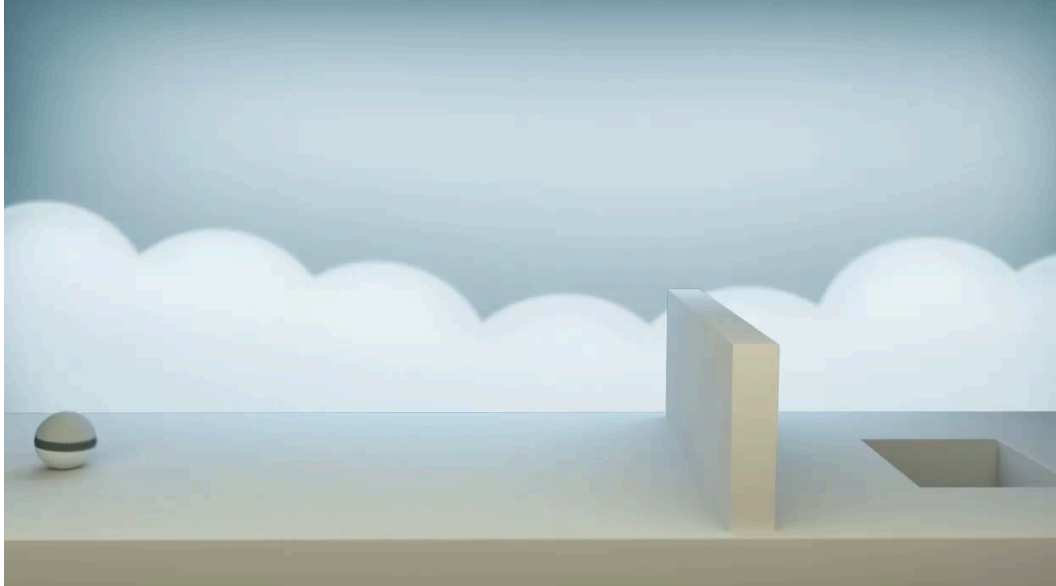
- **split\_frame\_to\_macroblocks**: χωρίζει το κάθε καρέ του βίντεο σε μακρομπλόκ μεγέθους 64x64 pixels. Επιστρέφει μια λίστα με τα μακρομπλόκ.
- **split\_macroblocks\_to\_frame**: συνθέτει τα μακρομπλόκ πίσω στο αρχικό Frame του βίντεο. Επιστρέφει το αρχικό Frame.
- **fit\_the\_size**: προσαρμόζει το μέγεθος μιας διάστασης στο μέγεθος του παραθύρου.

### 3.3 Το τεχνικό κομμάτι

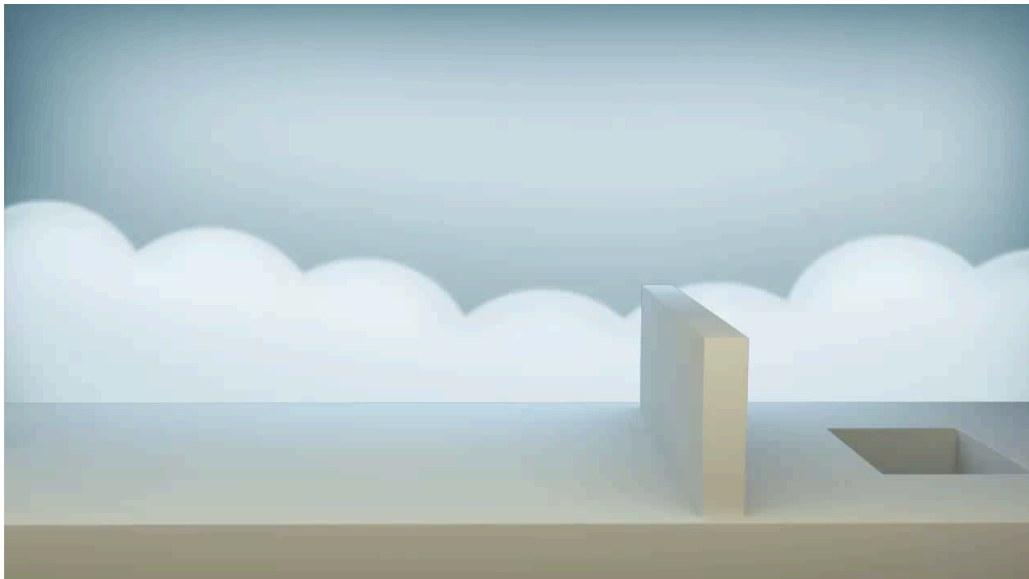
- Χρειάζεται να έχουμε εγκατεστημένη την python
- Χρειάζεται να έχουμε εγκατεστημένη την numpy όπου γίνεται με την εντολή στο terminal **pip install numpy**
- Χρειάζεται να έχουμε εγκατεστημένη την cv2 όπου γίνεται με την εντολή στο terminal **pip install opencv-python**
- Βρισκουμε το terminal στο path που βρίσκεται το Thema2.py , και γράφετε **python3 Thema2.py**

### 3.4 Παράδειγμα

Screenshot before the removal:



Screenshot after:



## **Βιβλιογραφία Θεμάτων:**

### Θέμα 1ο:

- Συστήματα Πολυμέσων - Αλγόριθμοι, Πρότυπα και Εφαρμογές Parag Havalдар, Gerard Medioni
- OpenCv Documentation:  
[https://docs.opencv.org/3.4/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html)
- collectionsDocumentation:  
<https://docs.python.org/3/library/collections.html>
- pickle Documentation:  
<https://docs.python.org/3/library/pickle.html>
- heapq Documentation:  
<https://docs.python.org/3/library/heapq.html>
- pathlib Documentation:  
<https://docs.python.org/3/library/pathlib.html>

### Θέμα 2ο:

- Συστήματα Πολυμέσων - Αλγόριθμοι, Πρότυπα και Εφαρμογές Parag Havalдар, Gerard Medioni
- OpenCv Documentation:  
[https://docs.opencv.org/3.4/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html)
- tqdm Documentation :  
<https://tqdm.github.io/>

### Θέμα 3ο:

- Numpy Documentation : <https://numpy.org/doc>
- OpenCv Documentation:  
[https://docs.opencv.org/3.4/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html)
- Συστήματα Πολυμέσων - Αλγόριθμοι, Πρότυπα και Εφαρμογές Parag Havalдар, Gerard Medioni