Tehničko veleučilište u Zagrebu

Obrada videozapisa

Seminar iz kolegija  
Napredne tehnike programiranja web servisa

|  |  |
| --- | --- |
| Nositelj kolegija: Alen Šimec | Autor seminara: Stjepan Salopek |

U Zagrebu, 11.1.2024.

Tablica sadržaja

[Sažetak 2](#_Toc155967583)

[Uvod 3](#_Toc155967584)

[Postavljanje tehnologija 3](#_Toc155967585)

[Postavljanje okruženja 3](#_Toc155967586)

[Rad sa videozapisima 4](#_Toc155967587)

[Učitavanje i spremanje videozapisa 4](#_Toc155967588)

[Metode spajanja više videozapisa u jedan 6](#_Toc155967589)

[Vizualna obrada i prilagođeni filtri 7](#_Toc155967590)

[Rad sa vremenskim zapisima 8](#_Toc155967591)

[Dodavanje tekstualnog sadržaja 9](#_Toc155967592)

[Vizualno-vremenska obrada 9](#_Toc155967593)

[Komentar na rad sa zvukom u videozapisima 10](#_Toc155967594)

[Zaključak 11](#_Toc155967595)

[Tablica programskih isječaka 12](#_Toc155967596)

[Tablica slika 13](#_Toc155967597)

[Izvori 14](#_Toc155967598)

Sažetak

Ovaj seminarski rad za cilj ima pružiti jednostavan i koristan pregled obrade videozapisa. Točnije, obrada se proučava na razini programskog kôda koji se izvršava u pozadini mnogih aplikacija.

U uvodu su navedena česta područja u kojima se može pronaći obrada videozapisa. Poslije uvoda naveden je i obrazložen odabir tehnologija, nakon čega slijede najčešći načini obrade. Primjeri se sastoje od opisa, programskog kôda i slika koje prikazuju tijek ili rezultat obrade. Iako je kôd pokrenut kraćem videozapisu radi bržeg izvršavanja kôda, na istom principu se mogu obrađivati svake vrste isječaka, filmova i serija.

Radi jednostavnosti, seminarski rad ne obrađuje temu rada sa zvukom, ali će se radi cjelovitosti prokomentirati.

Uvod

U današnjem vremenu, obrada videozapisa važan je čimbenik u mnogim područjima djelovanja te se informacije šire pretežito vizualnim putem. Osim industrija koje izravno rade sa videozapisima poput filmske ili televizijske industrije, obrada vizualnih podataka iznimno je bitna za svako kvalitetno poduzeće. Razlog tome je broj korisnika na društvenim mrežama i raznim drugim platformama koje služe za marketing proizvoda ili usluga.

Vrlo česti oblici videozapisa su kratke snimke i GIF zapisi. Objave u tom obliku su česta pojava te sve popularne društvene mreže veliku važnost pridodaju omogućavanju korisnicima jednostavno uređivanje i objavljivanje kratkih videozapisa. Primjer ovakvog načina poslovanja su dijeljenje priča (eng. *stories*) na Instagramu i Facebooku.

Postavljanje tehnologija

Prilikom odabira i postavljanja okruženja, potrebno je naći alat za prikaz videozapisa te programski jezik i biblioteku koja vrši obradu. Odabrani programski jezik je Python koji je poznat po jednostavnoj implementaciji i velikoj količini biblioteka koje služe za obradu vizualnih podataka, što je i razlog zašto se često odabire u domeni strojnog učenja. Sve korištene tehnologije otvorenog su kôda.

Za svrhe rada korištena je biblioteka MoviePy koja objedinjuje mnoge metode obrade videozapisa, a u pozadini koristi druge biblioteke popularne u radu sa matricama i slikama. Osim toga, da bi se sav kôd mogao izvršiti, potrebno je zasebno instalirati alate OpenCV i ImageMagick koji MoviePy biblioteci pružaju dodatne mogućnosti.

Za prikaz videozapisa često se koriste biblioteke i alati za izradu grafičkog sučelja kao što su Tkinter, PyQt, Kivy i wxPython. Međutim, s obzirom na to da je svrha seminara na što jednostavniji način približiti obradu videozapisa, niti jedna od navedenih mogućnosti nije odabrana, nego se videozapisi pregledavaju ugrađenom preview funkcijom.

Postavljanje okruženja

Primjeri su rađeni u operacijskom sustavu Windows. Za svrhe rada je na platformi GitHub izrađen repozitorij naziva py\_movie\_demo, čija se poveznica nalazi na kraju seminara. Za pisanje i pokretanje kôda, kao i za rad sa repozitorijem, korišten je Microsoftov alat Visual Studio Code.

Rad sa videozapisima

Na početku datoteke programskog jezika Python, potrebno je uključiti nekoliko biblioteka i postaviti određene varijable, što je vidljivo u sljedećem programskom isječku.

Biblioteka sys koristi se za bolju komunikaciju sa operacijskim sustavom. Primjer korištenja naveden je u dijelu Učitavanje i spremanje videozapisa.

Biblioteka math korištena je za kompleksnije matematičke operacije i konstante.

Objekt moviepy.config uključen je da bi se odredila i spremila putanja do ImageMagick biblioteke, što je prikazano u zadnja dva retka programskog isječka.

Objekt moviepy.editor glavni je izvor za rad sa videozapisima, a pruža mnoge mogućnosti učitavanja, obrade i spremanja.

import sys

import math

import moviepy.config as moviepyconf

import moviepy.editor as editor

moviepyconf.IMAGEMAGICK\_BINARY =  
 'C:\\putanja\_do\_datoteke\\magick.exe'

Programski isječak 1 Postavljanje okruženja programskog kôda

Učitavanje i spremanje videozapisa

Programski isječak na kraju trenutnog dijela seminara prikazuje uobičajen slijed naredbi prilikom obrade videozapisa. U početku je korištena Pythonova with..as sintaksa koja će originalni videozapis otključati i omogućiti drugim programima da ga koriste dok se skripta u potpunosti ne izvrši.

Samo učitavanje videozapisa vrši se pozivanjem konstruktora klase VideoFileClip koji vraća objekt sa raznim funkcijama.

Nakon toga može uslijediti jedna od metoda obrade koja će se kasnije spomenuti, a koja zatim vraća zaseban videozapis, ostavljajući izvorni netaknutim.

Zastavica write\_vf\_flag služi za dodatne mogućnosti prilikom pokretanja skripte. Točnije, ukoliko se u naredbenom retku prilikom pokretanja programskog kôda pošalje varijabla  
--write\_videofile=true, obrađeni videozapis će se spremiti. U protivnome, skripta će ga prikazati bez spremanja na disk.

With editor.VideoFileClip('video.mp4') as original:

  edited\_clip = FUNKCIJA\_OBRADE(original)

  write\_vf\_flag = len([

    arg for arg in sys.argv

    if arg.lower() == '--write\_videofile=true'

  ]) > 0

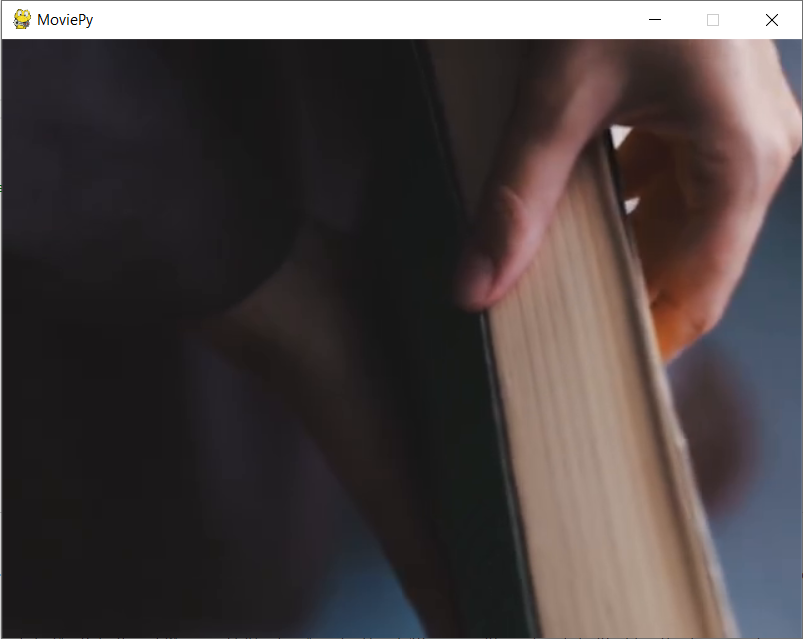
  if write\_vf\_flag:

    edited\_clip.write\_videofile('output.mp4')

  else:

    edited\_clip.preview()

Programski isječak 2 Učitavanje i spremanje videozapisa



Slika 1 Prikaz videozapisa MoviePy bibliotekom

Metode spajanja više videozapisa u jedan

Postoje tri česte metode spajanja više videozapisa u jednu snimku, a to su sljedeće funkcije i klase: concatenate\_videoclips, clips\_array te CompositeVideoClip. Programski isječak u nastavku prikazuje primjere uporabe sve tri metode, a ispod njega nalazi se i dijagram koji prikazuje novonastali videozapis na vremenskoj crti.

Prva metoda spaja videozapise na način da prima kao argument niz videozapisa, pokrećući jedan iza drugog jedan, redom kojim su postavljeni u nizu.

Druga metoda kao argument prima niz nizova videozapisa, gdje svaki od njih predstavlja jedan red u mreži videozapisa. Spaja ih na način da će sve videozapise pustiti istovremeno. U slučaju da ne traju svi videozapisi jednako dugo, na mjesto kraćega će nakon njegovog završetka doći crni ekran.

Treća metoda najkompleksnija je, no i najkorisnija. Kao i prethodna, pušta sve videozapise istovremeno. Međutim, videozapisi se neće prikazati u mreži, nego jedan na drugome. To znači da, ukoliko se želi prikazati videozapis na dnu, svi koji se nalaze iznad njega moraju biti pomaknuti ili djelomično prozirni. Na ovaj način omogućava se rad sa kompleksnim filterima.

conc\_vid = editor.concatenate\_videoclips([video\_1, video\_2])

grid\_vid = editor.clips\_array([

  [video\_1, video\_2],  # Row 1

  [video\_3, video\_4]   # Row 2

])

composite\_vid = editor.CompositeVideoClip([

  movie\_vid,   # Main video of composition

  special\_effect\_vid   # Special effect played on top of movie\_vid

.crossfadein(0.5)

.set\_start(2.1)

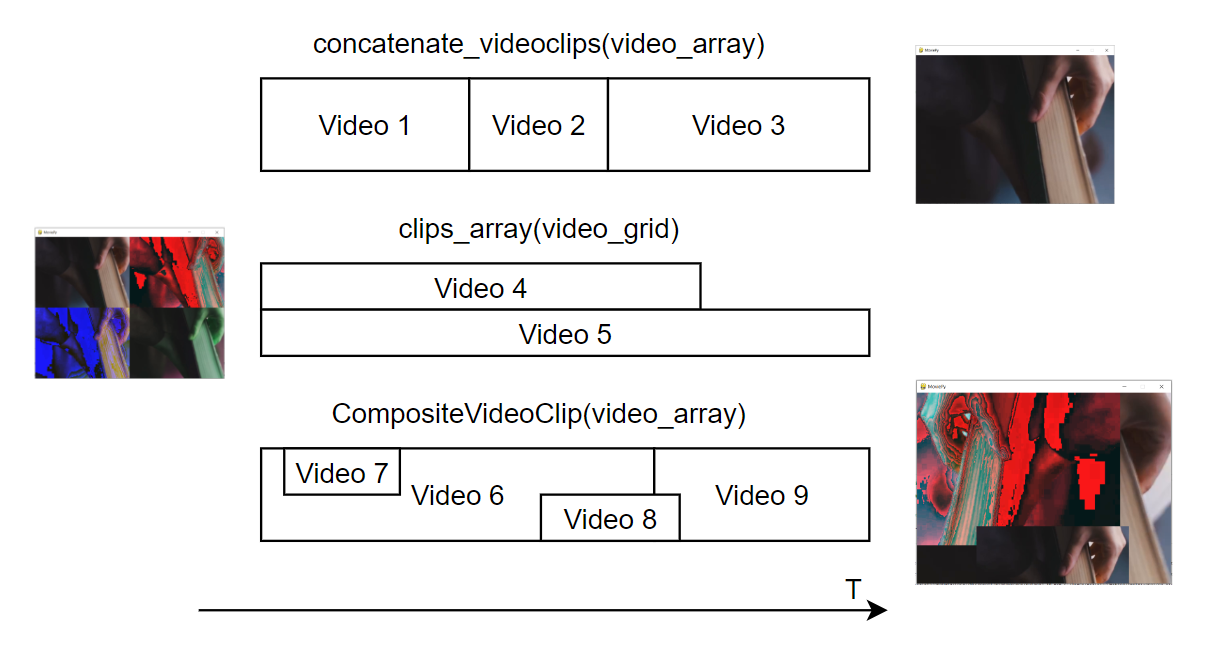
.set\_duration(1),

  movie\_titles\_vid   # Movie subtitles

.set\_position(('center', 'bottom')),

])

Programski isječak 3 Načini prikazivanja više videozapisa istovremeno



Slika 2 Grafički prikaz metoda spajanja više videozapisa u jedan

Vizualna obrada i prilagođeni filtri

Sljedeći primjer pokazuje česti način vizualne obrade. U prvom dijelu je pomoću funkcije fl\_image izvršena obrada svake sličice videozapisa. Točnije, na svakoj sličici RGB vrijednosti su pomnožene sa brojevima 1, 1 i 10, što je stvorilo videozapis sa prenaglašenom plavom bojom.

Drugi dio primjera prikazuje ručno stvaranje funkcije vizualne obrade. Funkcija change\_color\_channels mijenja vrijednosti kanala RGB u BGR. U stvarnosti stvorena funkcija može biti puno kompleksnija. Obrada se vrši prosljeđivanjem funkcijskog objekta metodi fx. Ovi primjeri mogu se koristiti za dodavanje određenih filtara na videozapis, iako je za to bolje koristiti kompozitne videozapise koji se obrađuju u jednom od narednih dijelova seminara.

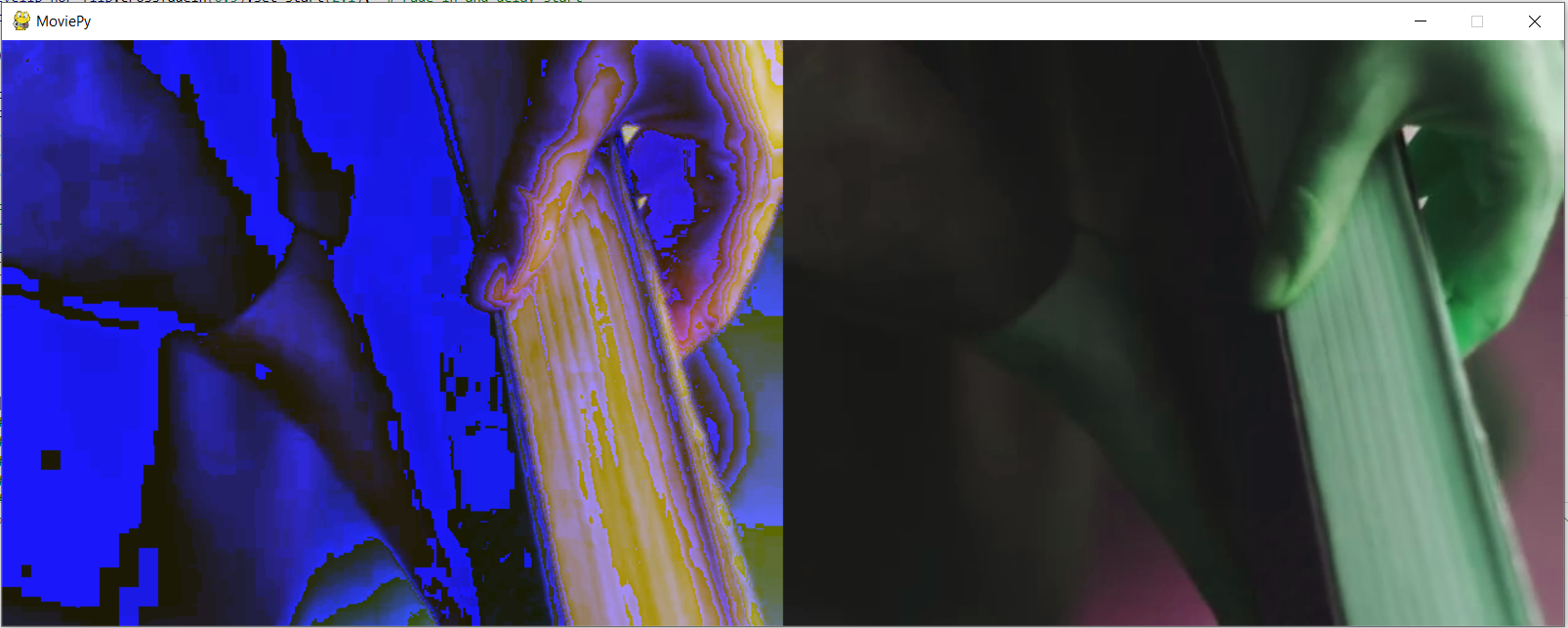
blue\_vid = original.fl\_image(lambda frame: frame \* [1, 1, 10])

def change\_color\_channels(clip):

  return clip.fl\_image(lambda frame: frame[..., [2, 0, 1]])

custom\_fx\_vid = original.fx(change\_color\_channels)

Programski isječak 4 Uređivanje vizualnih karakteristika videozapisa



Slika 3 Prikaz vizualno uređenih videozapisa

Rad sa vremenskim zapisima

Na sličan način se radi i u vremenskom aspektu. Funkcija fl\_time omogućava manipuliranje vremenskom crtom videozapisa. Primjerice, pomoću nje se videozapis može ubrzavati i usporavati, ali i uređivati na kompleksnije načine.

U sljedećem programskom kôdu, napisana je funkcija koja uzima prve dvije sekunde videozapisa te ih periodički prikazuje, unaprijed pa unazad, frekvencijom od jednog perioda po sekundi. Slučaj uporabe ovog učinka bili bi GIFovi koji se često koriste na društvenim mrežama.

Obrada se vrši na način da se u lambda funkciji promijeni parametar t koji označuje vrijeme sličice videozapisa. Promjenom parametra, promijeniti će se i sličica. Primjerice, način na koji možemo ubrzati videozapis dva puta jest da navedena lambda funkcija vraća vrijednost 2\*t. U sljedećem isječku korištena je kompleksnija funkcija koja će za svaki t vratiti vrijednost  
1 + sinus, odnosno vrijednosti od 0 do 2. Time se ponavljaju prve dvije sekunde videozapisa. Vrijednost 2\*PI\*t koja je poslana u sinusnu funkciju određuje frekvenciju prikazivanja petlje videozapisa.

time\_fx\_vid = original

  .fl\_time(lambda t: 1 + math.sin(2 \* math.pi \* t))

  .set\_duration(original.duration)

Programski isječak 5 Obrada vremenske crte videozapisa

Dodavanje tekstualnog sadržaja

Da bi se dodao tekst na videozapis, potrebno je napraviti zasebnu snimku koristeći klasu TextClip te odrediti stil teksta. Nakon toga, izvorna snimka spaja se sa tekstom koristeći klasu CompositeVideoClip koja se pojasnila u ranijim dijelovima seminara.

Sljedeći programski isječak na snimku dodaje tekst „Hello world!“ veličine 70 piksela i bijele boje, postavljajući ga u centar izvornog videozapisa tijekom prve tri sekunde.

clip\_title = editor.TextClip("Hello world!", fontsize=70, color="white")

  .set\_position('center')

  .set\_duration(3)

final\_clip = editor.CompositeVideoClip([original\_clip, clip\_title])

Programski isječak 6 Dodavanje teksta na videozapis

Vizualno-vremenska obrada

Postoji mogućnost izrade i efekta koji će istovremeno uzimati u obzir okvir, ali i vrijeme. Sljedeći programski isječak prikazuje implementaciju pomicanja videozapisa.

Programski kôd radi na način da se uzima okvir u vremenu t pomoću metode get\_frame. Zatim se stvara pomični okvir visine 150 piksela čije se koordinate računaju uzimajući u obzir navedenu visinu, vrijeme t i postavljenu brzinu pomicanja.

def scroll\_effect(get\_frame, t):

  frame = get\_frame(t)

scroll\_speed = 50

frame\_height = 150  
 y1\_coord = int(t \* scroll\_speed)   
 y2\_coord = int(t \* scroll\_speed) + frame\_height

  frame\_region = frame[y1\_coord:y2\_coord, :]

  return frame\_region

scrolling\_vid = original.fl(scroll\_effect)

Programski isječak 7 Primjer vizualno-vremenske obrade videozapisa

Za demonstraciju je stvoren kompozitni video uz pomoć izvornog videozapisa i klase ColorClip koja omogućava stvaranje videozapisa proizvoljnih karakteristika. Kompozitnom videu postavljena je crna pozadina, a preko nje je izvorni videozapis na lijevoj polovici te uređeni na desnoj. Na slici ispod sljedećeg programskog isječka, vidljivo je da je visina uređenog videozapisa manja, ali i da nije prikazan vrh videozapisa, nego dio videozapisa nakon pomicanja.

blackscreen = e.ColorClip(

  size=(original\_clip.w \* 2, original\_clip.h),

  color=(0,0,0)

).set\_duration(original\_clip.duration)

e.CompositeVideoClip([

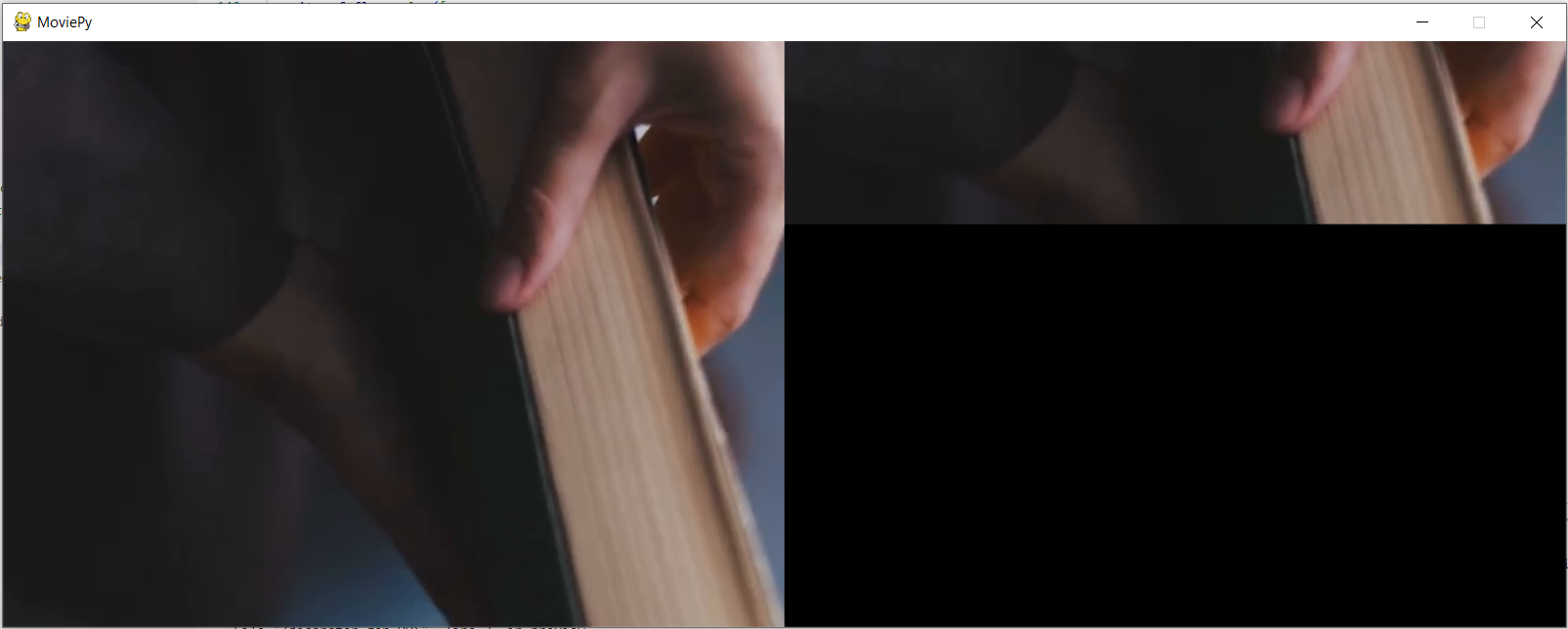
  blackscreen,

  original\_clip.set\_position((0, 0)),

  scrolling\_vid.set\_position((scrolling\_vid.w, 0))

]).preview()

Programski isječak 8 Priprema demonstracije vizualno-vremenske obrade



Programski isječak 9 Demonstracija vizualno-vremenske obrade videozapisa

Komentar na rad sa zvukom u videozapisima

S obzirom na to da cilj seminara nije bio pokriti sve moguće načine obrade videozapisa, nego pružiti jednostavan i koristan uvod, izostavljeni su načini obrade i spajanja zvučnih i vizualnih podataka.

Međutim, obrada zvuka prati slične principe, a biblioteka MoviePy pruža sučelje pomoću kojega je moguće sve spojiti u smislenu cjelinu.

Zaključak

U seminaru su na jednostavan način prikazane česte metode obrade videozapisa koje mogu poslužiti kao uvod u područje. U početku su navedeni razlozi važnosti teme, nakon čega je tema proširena obrazloženjem odabira tehnologija i primjerima obrade, zajedno sa programskim kôdom i odgovarajućim slikama.

Seminar je naveo i demonstrirao načine učitavanja i spremanja videozapisa MoviePy bibliotekom, metode prikazivanja više zapisa istovremeno, vizualnu obradu ugrađenim i ručno stvorenim filtrima, manipuliranje vremenskom crtom videozapisa, dodavanjem tekstualnog sadržaja. Osim toga, ostavljen je prostor za daljnje proširenje znanja preko uređivanja zvučnih zapisa.

Poveznicom u izvorima na kraju seminara moguće je pristupiti GitHub repozitoriju gdje se nalazi sav izvorni kôd seminarskog rada.

Tablica programskih isječaka

[Programski isječak 1 Postavljanje okruženja programskog kôda 4](#_Toc155966775)

[Programski isječak 2 Učitavanje i spremanje videozapisa 5](#_Toc155966776)

[Programski isječak 3 Načini prikazivanja više videozapisa istovremeno 6](#_Toc155966777)

[Programski isječak 4 Uređivanje vizualnih karakteristika videozapisa 7](#_Toc155966778)

[Programski isječak 5 Obrada vremenske crte videozapisa 8](#_Toc155966779)

[Programski isječak 6 Dodavanje teksta na videozapis 9](#_Toc155966780)

[Programski isječak 7 Primjer vizualno-vremenske obrade videozapisa 9](#_Toc155966781)

[Programski isječak 8 Priprema demonstracije vizualno-vremenske obrade 10](#_Toc155966782)

[Programski isječak 9 Demonstracija vizualno-vremenske obrade videozapisa 10](#_Toc155966783)

Tablica slika

[Slika 1 Prikaz videozapisa MoviePy bibliotekom 5](#_Toc155966834)

[Slika 2 Grafički prikaz metoda spajanja više videozapisa u jedan 7](#_Toc155966835)

[Slika 3 Prikaz vizualno uređenih videozapisa 8](#_Toc155966836)

Izvori

Poveznica na repozitorij: <https://github.com/shtef21/py_video_demo>

MoviePy biblioteka: <https://pypi.org/project/moviepy/>

Dokumentacija MoviePy biblioteke: <https://zulko.github.io/moviepy/>

OpenCV biblioteka: <https://opencv.org/>

ImageMagick biblioteka: <https://imagemagick.org/index.php>