



Anggota 1 : **Andreas Pascalis Tristan (121140017)**
Anggota 2 : **Adriel Gideon Benedict Sirait (121140196)**
Anggota 3 : **Natanael Argajova (121140205)**
Mata Kuliah: **Teknologi Multimedia (IF 4021)**

Tugas Ke: **UAS/Tubes**
Tanggal: **12-20-2024**

Timewarp Filter Menggunakan OpenCV

1 Pendahuluan

Penggunaan teknologi yang kian meningkat tiap harinya mendorong manusia untuk terus mencoba untuk menemukan hal baru. Salah satu bentuk teknologi itu ialah manipulasi video yang dapat memproses sebuah video sehingga menghasilkan sebuah output baru. Aplikasi seperti Instagram dan Tik Tok memberikan fitur berupa filter yang memberikan user kebebasan untuk memanipulasi bentuk video sesuai fungsi filter. Dalam kesempatan kali ini kami mencoba membuat salah satu filter tersebut, yang bernama timewarp filter. Menggunakan Python dan OpenCV user dapat menggunakan webcam untuk memberikan efek distorsi pada video sehingga menampilkan hasil baru berupa gambar yang telah diproses sedemikian rupa sehingga terlihat seperti gabungan gambar yang diobrak abrik [1].

2 Alat dan Cara Kerja

Filter ini merupakan bentuk proyek implementasi Timewarp filter menggunakan OpenCV yang berjalan menggunakan media **Python** sebagai bahasa pemrograman utama dan library **OpenCV** untuk memproses masukan berupa video. Perangkat yang digunakan berupa laptop dan media webcam untuk input video.

Langkah awal jalannya sistem berawal dari membuka webcam user, kemudian akan ada garis hijau penanda yang turun secara perlahan sambil memproses setiap piksel yang dilalui. Ketika kita melakukan sebuah aksi ketika garis melewati bagian itu akan terdistorsi layaknya membuat video stop motion yang merekam frame per frame namun dalam kasus ini per pikses video. Hasil akhir akan seperti [video berikut ini](#).

3 Penjelasan Kode Program

3.1 Import Library

```
1  
2 import cv2  
3
```

Kode 3.1: Library yang Digunakan

Penjelasan:

- *importcv2*: library yang digunakan untuk memproses gambar/video secara *real time*, termasuk mendeteksi wajah ataupun objek lainnya.

3.2 Fungsi Utama

```

1 threshold = 2 # Jumlah iterasi pixel persatuan waktu. makin tinggi maka garis distorsi akan
2   semakin cepat.
3 is_upside_down = False # fungsi ini digunakan untuk membalik pixel vertikal
4
5 ret, initial_frame = cap.read()
6 if not ret:
7     print("Error: Frame tidak dapat dibaca.")
8     cap.release()
9     return # di cek keberhasilannya semisal gagal maka pesan di atas akan keluar.
10
11 warped_frame = initial_frame.copy() # salinan dari frame yang berhasil disimpan
12 temp_frame = initial_frame.copy() # kemudian disimpan sementara di sini.
13 current_row = 0 # indeks awal
14

```

Kode 3.2: Fungsi Utama

Penjelasan:

- *threshold* = 2: Menentukan jumlah iterasi per satuan waktu. Semakin tinggi nilainya, semakin cepat garis distorsi bergerak. Ini adalah parameter inti yang memengaruhi efek visual.
- *is_upside_down* = False: Variabel boolean yang menentukan apakah filter akan membalikkan frame secara vertikal.
- *ret*: Nilai boolean (True/False) yang menunjukkan apakah pembacaan frame berhasil atau tidak.
- *initial_frame*: frame pertama yang berhasil dibaca dari video.
- *cap.read()*: Membaca frame dari video yang sedang diproses.
- *cap.release()*: Melepaskan (release) objek *cap* yang mewakili video atau kamera, untuk memastikan tidak ada resource yang tetap terkunci.
- *warped_frame* = *initial_frame.copy()*: Membuat salinan dari frame pertama (*initial_frame*). Salinan ini akan digunakan untuk menerapkan efek distorsi tanpa merusak frame aslinya.
- *temp_frame* = *initial_frame.copy()*: Salinan sementara dari frame awal. Ini digunakan untuk menyimpan data sementara saat efek diterapkan pada frame utama.
- *current_row* = 0: Inisialisasi indeks baris (*current_row*) di frame ke 0. Indeks ini menentukan dari baris mana distorsi dimulai, dan akan bertambah seiring waktu saat filter diterapkan.

3.3 Pemrosesan Frame Real Time

```

1 while True:
2     ret, frame = cap.read()
3     if not ret:
4         print("Gagal membaca frame dari kamera.")
5         break
6
7     frame = cv2.flip(frame, int(not is_upside_down))
8
9     # fungsi ini digunakan untuk menggabungkan bagian frame yang baru dan lama kedalam efek
10    timewarp

```

```

11     temp_frame[current_row:current_row + threshold] = frame[current_row:current_row + threshold
12 ]
13     warped_frame[:current_row] = temp_frame[:current_row]
14     warped_frame[current_row:] = frame[current_row:]
15
16     # fungsi untuk membuat garis hijau penanda distorsi
17     cv2.line(warped_frame, (0, current_row), (warped_frame.shape[1], current_row), (0, 255, 0),
18             thickness=2)
19
20     current_row += threshold
21
22     # tampilkan video
23     cv2.imshow('Timewarp Filter', warped_frame)
24
25     # memberhentikan proses semisal tombol esc ditekan (27 adalah ASCII untuk esc)
26     if cv2.waitKey(33) == 27 or current_row >= frame.shape[0]:
27         break

```

Kode 3.3: Pemrosesan Frame Real Time

Penjelasan:

- *cap*: Objek ini digunakan untuk menangkap frame dari kamera secara real-time.
- *ret*: Menyimpan informasi apakah frame berhasil dibaca dari kamera.
- *frame*: Frame saat ini yang diambil dari kamera dan diproses dalam setiap iterasi.
- *is_upside_down*: Mengontrol apakah frame perlu dibalik (flip) atau tidak.
- *temp_frame*: Buffer sementara untuk menyimpan bagian dari frame baru yang digunakan untuk efek timewarp.
- *warp_frame*: Frame yang telah dimodifikasi dengan efek timewarp. Ini adalah hasil akhir yang ditampilkan ke layar.
- *current_row*: Baris saat ini pada frame yang sedang diproses untuk menerapkan efek timewarp.
- *threshold*: Menentukan ukuran (dalam piksel) dari bagian frame yang diproses dalam setiap iterasi.

3.4 Penyimpanan Output

```

1     output_filename = 'timewarp_output.png'
2     cv2.imwrite(output_filename, warped_frame)
3     print(f"Timewarp filter berhasil disimpan sebagai '{output_filename}'.")
4
5

```

Kode 3.4: Penyimpanan Output

Penjelasan:

- *output_filename*: Fungsi untuk menyimpan hasil akhir dalam format nama *timewarp_output.png*.
- *cv2.imwrite(output_filename, warped_frame)* : Menyimpan hasil akhir menggunakan parameter *output_filename* untuk Nama file tempat gambar akan disimpan dan *warped_frame* Frame (citra) yang akan disimpan.

3.5 Cleanup

```
1 cap.release()
2 cv2.destroyAllWindows()
3
4
```

Kode 3.5: Cleanup

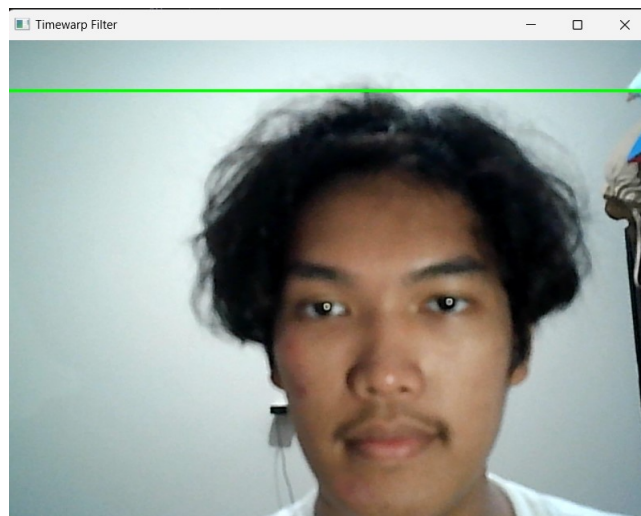
Penjelasan:

- *cap.release()*: Melepaskan objek VideoCapture yang terhubung ke video file atau kamera. Jika *cap* adalah objek hasil dari *cv2.VideoCapture()*, maka fungsi ini memastikan bahwa koneksi ke sumber video/kamera dihentikan dan resource dikembalikan.
- *cv2.destroyAllWindows()*: Menutup semua jendela yang dibuka oleh *cv2.imshow()*. Jendela yang menampilkan gambar atau video akan langsung ditutup dan program kembali ke terminal/console.

4 Hasil dan Pembahasan

4.1 Hasil

Hasil akhir program dapat berhasil menjalankan program timewarp filter untuk menghasilkan gambar akhir dimana video webcam realtime yang awalnya normal menjadi terdistorsi yang kemudian disimpan dalam file png.



Gambar 4.1: Keadaan Awal Ketika Program Berjalan

Ketika program dijalankan maka akan terlihat garis hijau yang turun perlahan. User dapat menggunakan berbagai gerakan atau objek bantuan yang dapat menambah pengalaman pengguna untuk membuat hasil yang unik.



Gambar 4.2: Hasil Akhir Setelah Program Selesai Berjalan

Gambar di atas merupakan bentuk hasil gambar yang telah berhasil di timewarp. gambar yang telah dilewati oleh garis hijau sebelumnya mengalami iterasi ulang sehingga setiap piksel tampak seperti berada pada bagian waktu yang berbeda.

4.2 Pembahasan

Menggunakan bantuan Python dan OpenCV kita dapat membuat sebuah filter sederhana seperti timewarp filter yang mampu memanipulasi hasil sebuah video menjadi gambar distorsi dalam berbagai macam bentuk. Program berjalan dengan mengiterasi ulang tiap frame yang dilalui lalu membalik tampilan frame serta merekam ulang tiap frame sehingga hasil akhir berupa gambar yang terlihat seperti gabungan berbagai macam gambar yang mengalami deformasi.

5 Langkah Instalasi

5.1 Cara Instalasi

Clone repository dari Github

```
1  
2 git clone https://github.com/AndreasPascalis017/IF4021_timewarp_filter.git  
3
```

Kode 5.1: Clone Repository

Install requirements.txt

```
1  
2 pip install -r requirement.txt  
3
```

Kode 5.2: Install Dependency

References

- [1] J. Minichino and J. Howse, “Learning opencv3 computer vision with python,” <https://aiidecoe.com/images/Learning%20openCV3%20computer%20vision%20with%20Python.pdf>, 15 Sep. 2015, accessed: 2024-12-22.