Praktikum 1 Visi Komputer dan Pengolahan Citra

Firnanda Pristiana Nurmaida (112200008 / S2 TE)

Program 1 Kamera Minoru

```
import numpy as np
import cv2

left_camera = cv2.VideoCapture(4)
right_camera = cv2.VideoCapture(6)

# id cam l = 4
# id cam r = 6

# Check if the cameras opened successfully
if not left_camera.isOpened() or not right_camera.isOpened():
    print("Error: Unable to open one or both cameras")
    exit()

stereo = cv2.StereoSGBM_create(numDisparities=3, blockSize=15)
```

Membuka kamera sebelah kanan dan kiri

Apakah kamera telah berhasil terbuka?

Membuat objek stereo SGBM

```
while True:
    ret1, left_frame = left_camera.read()
    ret2, right_frame = right_camera.read()

    if not (ret1 and ret2):
        break

    left_gray = cv2.cvtColor(left_frame, cv2.COLOR_BGR2GRAY)
    right_gray = cv2.cvtColor(right_frame, cv2.COLOR_BGR2GRAY)

    try:

        disparity = stereo.compute(left_gray, right_gray) + 0.0000001
    except cv2.error as e:
        print(f"Error computing disparity: {e}")
        continue
```

Mengambil frame dari kedua kamera

Konversi ke abu-abu

Menghitung disparit

```
    depth = 1.0 / disparity
```

Menghitung jarak ke objek, Faktor konversi yang bergantung pada konfigurasi kamera

```
    cv2.imshow('Camera Laptop', left_frame)
    cv2.imshow('Camera HP', right_frame)
    cv2.imshow('Depth Map', depth)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

left_camera.release()
right_camera.release()
cv2.destroyAllWindows()
```

Menampilkan citra kedua kamera dan citra kedalaman

Program Minoru Dilengkapi Disparity Map

```
import cv2
import numpy as np

# Open the left and right cameras (modify indices if needed)
left_cap = cv2.VideoCapture(4)
right_cap = cv2.VideoCapture(6)

# Check if the cameras opened successfully
if not left_cap.isOpened() or not right_cap.isOpened():
    print("Error: Could not open one or both cameras.")
    exit()

# Set up stereo block matching parameters (adjust as needed)
stereo = cv2.StereoBM_create(numDisparities=16, blockSize=5)
```

Membuka kamera sebelah kanan dan kiri

Apakah kamera berhasil dibuka?

Mengatur parameter stereo block matching

```
while True:
    # Read frames from both cameras
    ret1, left_frame = left_cap.read()
    ret2, right_frame = right_cap.read()

    # Check if both frames were read successfully
    if not ret1 or not ret2:
        print("Error: Could not read frames from one or both cameras.")
        break

    # Convert frames to grayscale
    left_gray = cv2.cvtColor(left_frame, cv2.COLOR_BGR2GRAY)
    right_gray = cv2.cvtColor(right_frame, cv2.COLOR_BGR2GRAY)
```

Membaca kamera kanan dan kiri

Apakah kamera berhasil terbaca?

Konversi frame ke grayscale

```
    # Compute disparity map
    disparity = stereo.compute(left_gray, right_gray)

    # Normalize disparity values for better visualization
    disparity = cv2.normalize(disparity, None, 0, 255, cv2.NORM_MINMAX)
    disparity = disparity.astype(np.uint8)

    # Display the disparity map
    cv2.imshow('Disparity Map', disparity)
    cv2.imshow('left frame', left_frame)
    cv2.imshow('right frame', right_frame)

    # Exit the loop if the 'q' key is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release both cameras and close the window
left_cap.release()
right_cap.release()
cv2.destroyAllWindows()
```

Menghitung disparity map

Normalisasi disparitas

Menampilkan disparitas

Menutup loop dengan menekan q