Motion Capture and Future Interaction Technology Research

MoCap with hand tracking

Speaker: Fu-Song Hsu

What's Covered

- Setup hand tracking module
- Two-Hand Tracking
- Hand Pose Recognition (class work)
- Interactive Game

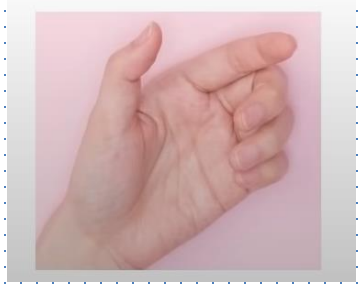
MediaPipe

Hand Tracking

Palm Detection

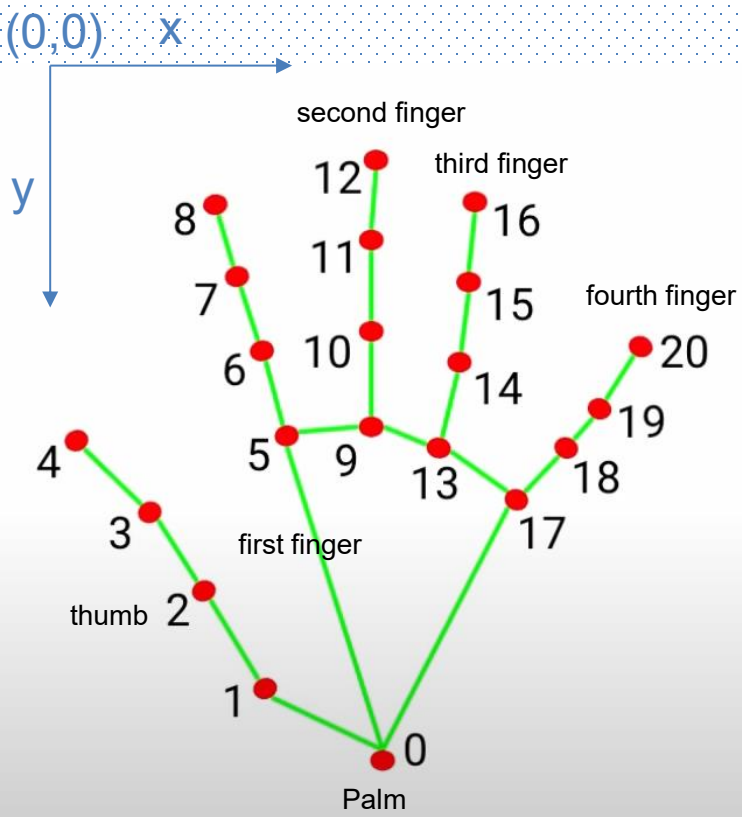
complete image

Hand Landmarks



cropped image

Hand Landmarks



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Source: Media Pipe Website

Two-Hand Tracking

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [2]:
```
import mediapipe as mp
import cv2
import time
import math
import numpy as np
```

Import libraries

In [3]:
```
mpDraw = mp.solutions.drawing_utils
mpHands = mp.solutions.hands
```

Two-Hand Tracking

In [4]:
```
hands = mpHands.Hands(
    static_image_mode=False,
    #model_complexity=0,
    max_num_hands=2,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
    )
```

create a module for tracking hands

In [5]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_handedness)):
            thisHandType=results.multi_handedness[i].classification[0].label
            thisHand=results.multi_hand_landmarks[i]
            mpDraw.draw_landmarks(frame, thisHand, mpHands.HAND_CONNECTIONS)

            for id, lm in enumerate(thisHand.landmark):
                hx, hy = int(lm.x * w), int(lm.y * h)
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
                if id==0:
                    cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
    etime=time.time()
    fps=round(1/(etime-stime),2)
    cv2.putText(frame,"FPS:" + str(fps),(10,50), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 255), 3)
```

```
    #model_complexity=0,
    max_num_hands=2,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
    )
```

send our webcam image to hand module

In [5]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_handedness)):
            thisHandType=results.multi_handedness[i].classification[0].label
            thisHand=results.multi_hand_landmarks[i]
            mpDraw.draw_landmarks(frame, thisHand, mpHands.HAND_CONNECTIONS)

            for id, lm in enumerate(thisHand.landmark):
                hx, hy = int(lm.x * w), int(lm.y * h)
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
                if id==0:
                    cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
    etime=time.time()
    fps=round(1/(etime-stime),2)
    cv2.putText(frame,"FPS:" + str(fps),(10,50), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 255), 3)
    cv2.imshow('Webcam',frame)
    key=cv2.waitKey(1)
    if key==ord('a'):
        cv2.imwrite('webcam.jpg',frame)
    if key==ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

apply custom colors

get hand label

get hand landmark

draw hand landmarks

get FPS

Hand Pose Recognition

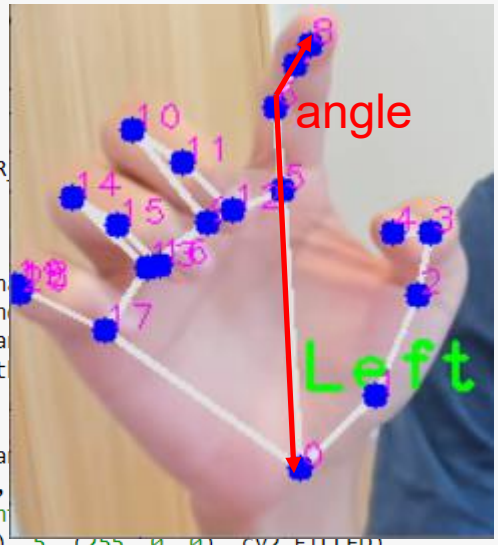
Hand Pose Recognition

In [29]:
```
hands = mpHands.Hands(
    static_image_mode=False,
    model_complexity=0,
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.5
    )
```

In [30]:
```
AngleTH=130
def findAngleF(a,b,c):
    ang = math.degrees(math.atan2(c[2]-b[2], c[1]-b[1]) - math.atan2(a[2]-b[2], a[1]-b[1]))
    print(ang)
    if ang<0 :
      ang=ang+360
    if ang >= 360- ang:
        ang=360-ang
    return round(ang,2)
```

create a function for getting degree

In [31]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_h
            thisHandType=results.multi_han
            thisHand=results.multi_hand_la
            mpDraw.draw_landmarks(frame, t

            thisHandLMList = []
            for id, lm in enumerate(thisHa
                thisHandLMList.append([id,
                hx, hy = int(lm.x * w), in
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
```



For an instance,
a=0
b=6
c=8

Hand Pose Recognition
By calculating angles

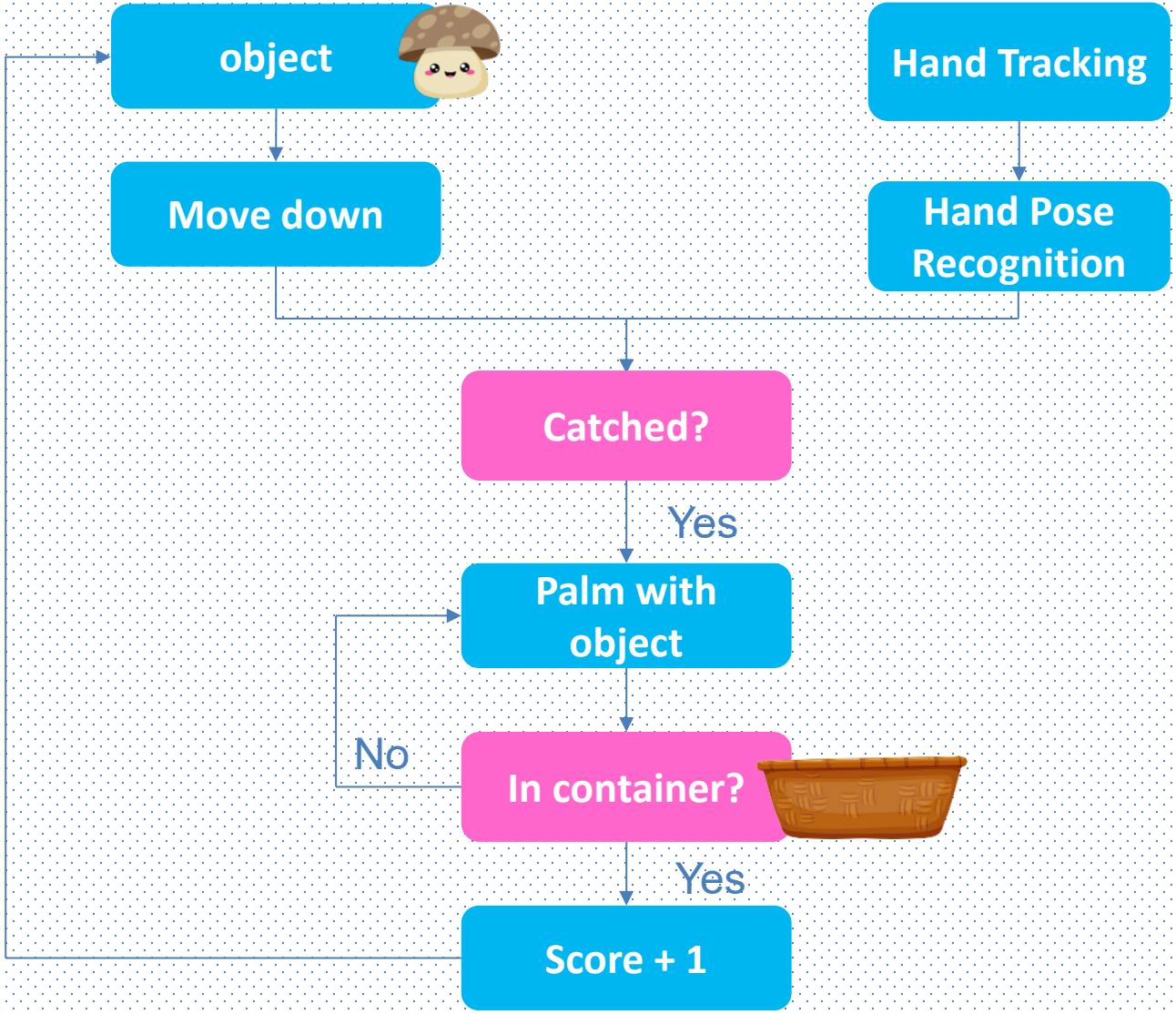
In [31]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_handedness)):
            thisHandType=results.multi_handedness[i].classification[0].label
            thisHand=results.multi_hand_landmarks[i]
            mpDraw.draw_landmarks(frame, thisHand, mpHands.HAND_CONNECTIONS)

            thisHandLMList = []
            for id, lm in enumerate(thisHand.landmark):
                thisHandLMList.append([id, lm.x, lm.y,lm.z])
                hx, hy = int(lm.x * w), int(lm.y * h)
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
                if id==0:
                    cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
            finger=[0,0,0,0,0]
            if (findAngleF(thisHandLMList[0],thisHandLMList[3],thisHandLMList[4])>AngleTH):
                finger[0]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[6],thisHandLMList[8])>AngleTH):
                finger[1]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[10],thisHandLMList[12])>AngleTH):
                finger[2]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[14],thisHandLMList[16])>AngleTH):
                finger[3]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[18],thisHandLMList[20])>AngleTH):
                finger[4]=1
            print(finger)


            text=""
            if (finger==[0,0,0,0,0]):
                text="Zero"
            if (finger==[1,1,1,1,1]):
                text="Hi"
```

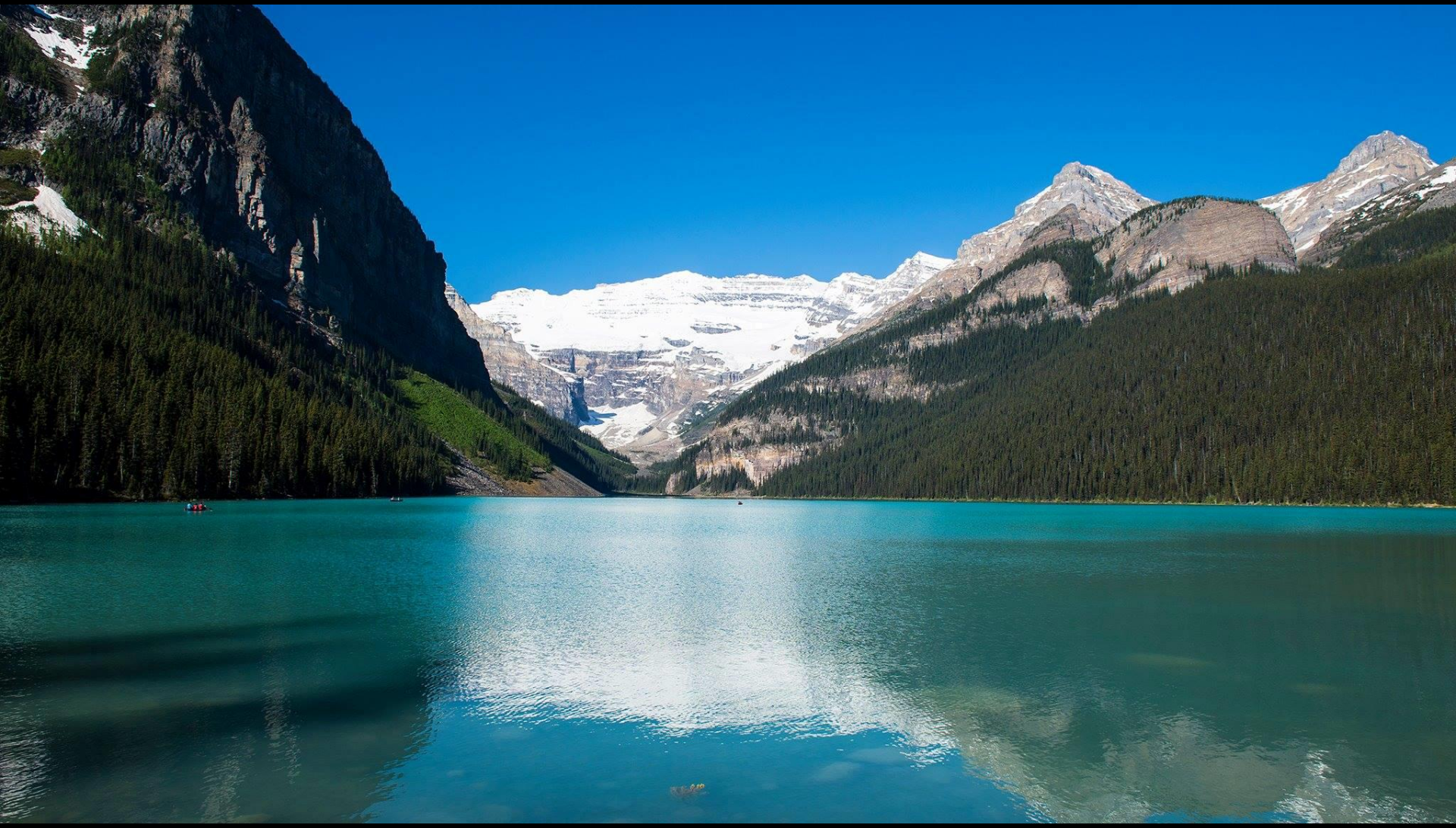
Class work

Hand Gesture Recognizer

Five hand gestures: victory, hi, spider-man and so on

Interactive Game


object
Hand Tracking
Move down
Hand Pose Recognition
Catched?
Yes
Palm with object
No
In container?
Yes
Score + 1



Q&A