

Freeze

Kikiola Akanbi
Oluwadare Bankole
Harena Yemane
Zena Aduwo

Dr. Danda Rawat

April 15, 2021



Problem Statement

With an increase in virtual learning due to Covid-19, teachers need to ensure students are gaining the full learning experience and paying attention without interrupting their live teaching to address distracted individuals, allowing teachers to benefit from higher in-class engagement, stronger test scores, and better peer-to-teacher relationships.

Design Requirements

Product Specification

- Motion Detection
- Present/ Absent System
- 9 students on virtual learning platform (Zoom)

Constraints

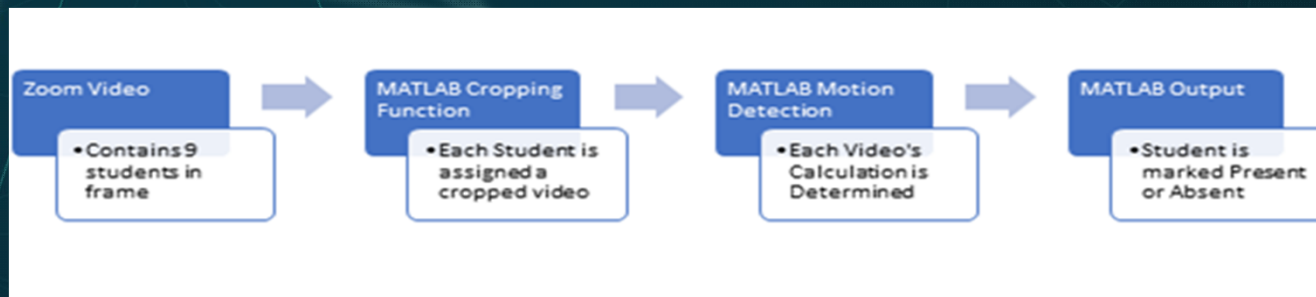
- No cost; Zoom free subscription
- Product completed and ready for testing by 4/10/21
- Student Privacy

Rules and Regulations

- Right to Privacy as outlined in the United States Bill Of Rights

Solution Design (Dare)

- Input is Zoom pre-recorded video with 9 pinned students
- Each student is cropped and assigned a respective video
- Each student's video is passed through the motion detection function
- Student is marked absent or present.



Sprint 1

Sprint: Identifying code for each function

Increment: Utilize computer vision toolbox in MATLAB for facial feature detection and motion detection and test functionality for each

```
nFrames = 1;
count = 0;
counter = 0;
while (nFrames<x)    % Process for the first 100 frames.
    %readFrame = frameRGB;
    % Acquire single frame from imaging device.
    frameRGB = read(vidDevice,nFrames);
    %frameRGB = vidDevice();

    % Compute the optical flow for that particular frame.
    flow = estimateFlow(opticFlow,rgb2gray(frameRGB));

    imshow(frameRGB)
    hold on
    plot(flow,'DecimationFactor',[10 10],'ScaleFactor',100)
    pause(.000001)
    hold off
```

```
oldPoints = points;
while hasframe(videoReader)
    % get the next frame
    videoFrame = readFrame(videoReader);

    % Track the points. Note that some points may be lost.
    [points, isFound] = step(pointTracker, videoFrame);
    visiblePoints = points(isFound, :);
    oldInliers = oldPoints(isFound, :);

    if size(visiblePoints, 1) >= 2 % need at least 2 points
        % Estimate the geometric transformation between the old points
        % and the new points and eliminate outliers
        [xform, inlierIdx] = estimateGeometricTransform2D(...
            oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);
        oldInliers = oldInliers(inlierIdx, :);
        visiblePoints = visiblePoints(inlierIdx, :);

        % Apply the transformation to the bounding box points
        bboxPoints = transformPointsForward(xform, bboxPoints);

        % Insert a bounding box around the object being tracked
        bboxPolygon = reshape(bboxPoints', 1, []);
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, ...
            'LineWidth', 2);

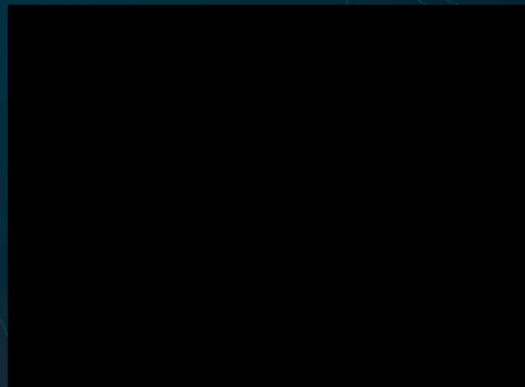
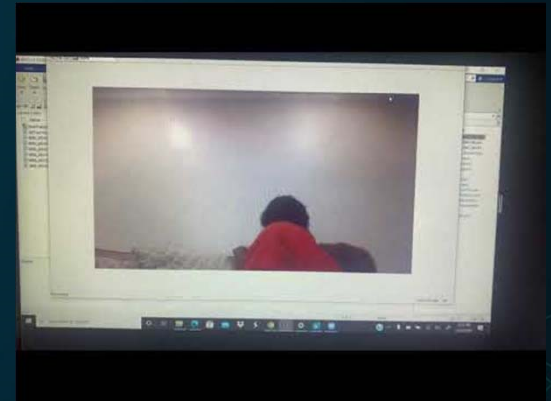
        % Display tracked points
        videoFrame = insertMarker(videoFrame, visiblePoints, 's', ...
            'Color', 'white');

        % Reset the points
        oldPoints = visiblePoints;
        setPoints(pointTracker, oldPoints);
        step(videoPlayer, videoFrame);
    end
```

Sprint 2

Sprint: Combine all 3 separate codes for 1 student

Increment: Write a code for 1 student on pre-recorded video.



Sprint 3

Sprint: Implement absent and present function

Increment: Write a code that outputs present/absent depending on video input and time constraints

```
yy = sum(flow.Magnitude(:) == 0);  
  
if yy == 921600  
    counter = counter + 1;  
end  
  
if flow.Magnitude(:) == 0  
    counter = counter + 1;  
end  
  
nFrames = nFrames + 1;  
  
end  
r = nFrames*0.5;  
if counter >= r  
    disp("Absent")  
else  
    disp("Present")  
end
```

Conclusion

Final Product Features:

- User records a Zoom session using the application's internal features
- User utilizes the cropping program to convert the larger video into a smaller video with a single student
- User runs the Motion Detection Program to determine the student's participation during the session

