I. Image Acquisition:

a)

```
>> webcam

ans =

  webcam with properties:

                    Name: 'HP TrueVision HD Camera'
              Resolution: '640x360'
     AvailableResolutions: {1x5 cell}
                Exposure: -6
              Brightness: 128
           WhiteBalance: 4000
                     Hue: 0
           ExposureMode: 'auto'
    BacklightCompensation: 1
                   Gamma: 120
                Contrast: 32
              Saturation: 64
         WhiteBalanceMode: 'auto'
                    Gain: 4
               Sharpness: 2
```

b)

```
>> webcamlist

ans =

    'HP TrueVision HD Camera'
```

c)

```
>> imaqhwinfo()

ans =

    InstalledAdaptors: {'winvideo'}
       MATLABVersion: '9.0 (R2016a)'
```

```
        ToolboxName: 'Image Acquisition Toolbox'
     ToolboxVersion: '5.0 (R2016a)'
```

d)

```
>> x=videoinput('winvideo', 1);
>> imaqhwinfo(x)

ans =

            AdaptorName: 'winvideo'
             DeviceName: 'HP TrueVision HD Camera'
              MaxHeight: 720
               MaxWidth: 1280
         NativeDataType: 'uint8'
           TotalSources: 1
  VendorDriverDescription: 'Windows WDM Compatible…'
    VendorDriverVersion: 'DirectX 9.0'
```

e)

```
>> preview(x)
```

II. Motion Detection and Event Triggering:

```
function [] = official()
% Create a cascade detector object.
faceDetector = vision.CascadeObjectDetector();

videoFileReader = imaq.VideoDevice('winvideo', 1, 'MJPG_640x480','ROI',[1 1 640 480]);
videoFrame      = step(videoFileReader);
bbox            = step(faceDetector, videoFrame);
while(size(bbox,1)<1)
  videoFrame= step(videoFileReader);
   bbox= step(faceDetector, videoFrame);
end

x = bbox(1, 1); y = bbox(1, 2); w = bbox(1, 3); h = bbox(1, 4);
bboxPolygon = [x, y, x+w, y, x+w, y+h, x, y+h];

videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon);

figure(1); imshow(videoFrame); title('Detected face');
```

```
% Detect feature points in the face region.
points = detectMinEigenFeatures(rgb2gray(videoFrame), 'ROI', bbox);

% Display the detected points.
figure('name','detected'), imshow(videoFrame), hold on, title('Detected features');
plot(points);

pointTracker = vision.PointTracker('MaxBidirectionalError', 2);

% video frame.
points = points.Location;
initialize(pointTracker, points, videoFrame);
videoPlayer  = vision.VideoPlayer('Position',...
    [100 100 [size(videoFrame, 2), size(videoFrame, 1)]+30]);

% transformation between the points in the previous and the current frames
oldPoints = points;
newperson=1;

for taoefaeffadef =1:1800

    videoFrame = step(videoFileReader);

    % 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
    if newperson==1
    matchPic = imcropPolygon(bboxPolygon,videoFrame);
    matchPic = cutPic(matchPic);
    imwrite(matchPic,'test2.bmp','bmp');
    number = libCheck(load_database(3),imread('test2.bmp'))
    if number<11
       disp('hello liuliu')
    elseif number<21
          disp('hello zhouyuchun')
    elseif number<31
              disp('hello zhouchi')
    end

    newperson=0;

    end
```

```
        [xform, oldInliers, visiblePoints] = estimateGeometricTransform(...
            oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);


        [bboxPolygon(1:2:end), bboxPolygon(2:2:end)] ...
          = transformPointsForward(xform, bboxPolygon(1:2:end), bboxPolygon(2:2:end));

        % Insert a bounding box around the object being tracked
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon);

        videoFrame = insertMarker(videoFrame, visiblePoints, '+', ...
            'Color', 'red');

        oldPoints = visiblePoints;
        setPoints(pointTracker, oldPoints);
    else
        release(pointTracker);
        pointTracker = vision.PointTracker('MaxBidirectionalError', 2);

        % Detect feature points in the face region.
        points = detectMinEigenFeatures(rgb2gray(videoFrame), 'ROI', bbox(0,:));
        points = points.Location;
        initialize(pointTracker, points, videoFrame);
        oldPoints = points;

        bbox= step(faceDetector, videoFrame);
        while(size(bbox,1)<1)
            videoFrame = step(videoFileReader);
            bbox= step(faceDetector, videoFrame);
            step(videoPlayer, videoFrame);
        end

        x = bbox(1, 1); y = bbox(1, 2); w = bbox(1, 3); h = bbox(1, 4);
        bboxPolygon = [x, y, x+w, y, x+w, y+h, x, y+h];


        % Draw the returned bounding box around the detected face.
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon);

        newperson=1;

    end
    step(videoPlayer, videoFrame);

end
```

```
% Clean up
release(videoFileReader);
release(videoPlayer);
release(pointTracker);
end
```