

طراحی و بررسی پارامترهای تشفیص چهره

در متلب

دانشجو: پرستو مالکی

استاد راهنما: استاد نوشین بهزادپور

این برنامه نشان میدهد چگونه به وسیله ی الگوریتم KLT صورت را در وبکم تشخیص داده و دنبال کنیم.

الگوریتم KLT

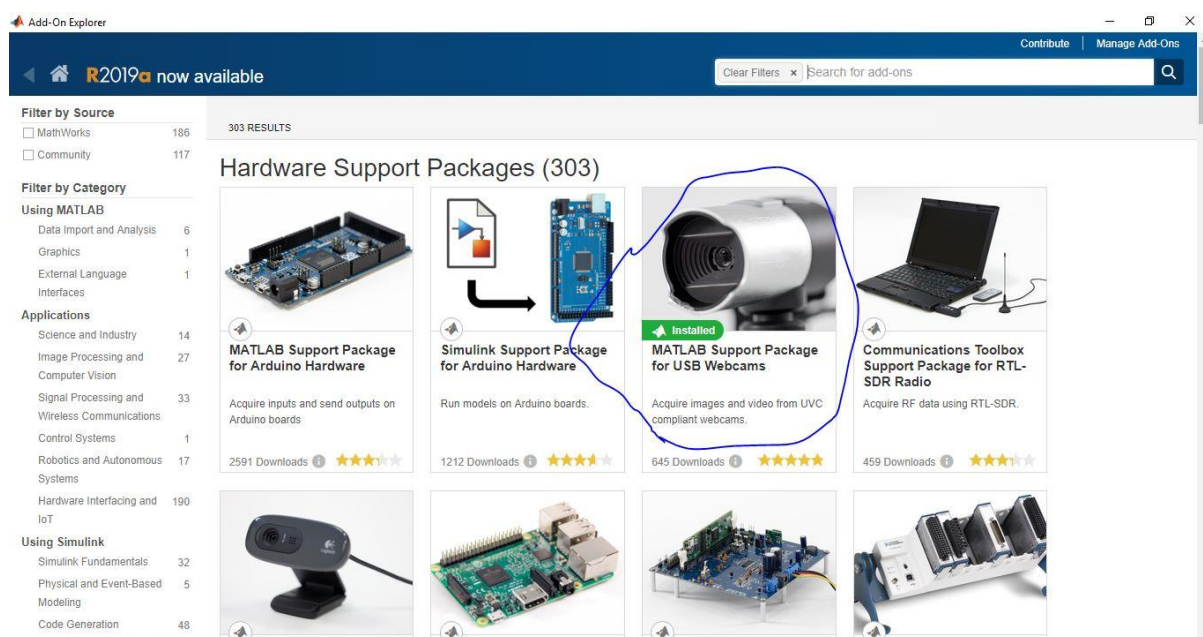
Kanade–Lucas–Tomasi feature tracker

در دید کامپیوتری (computer vision) الگوریتم KLT یک روش برای خارج کردن ویژگی های خاص است که سریع تر و ارزان تر نسبت به روش های پیشین خارج کردن ویژگی است.

بررسی اجمالی

تشخیص اشیا و دنبال کردن آن ها در دید کامپیوتری بسیار مهم است و استفاده های زیادی در برنامه های تشخیص کار، ایمنی خودرو و نظارت دارد. در این پروژه ما یک سیستم ساده برای تشخیص دنبال کردن صورت توسط وبکم ایجاد می کنیم. متلب توسط یک پکیج اضافه (که روش نصب آن ساده است) از وبکم پشتیبانی می کند.

برای نصب پکیج وبکم ابتدا دستور supportPackageInstaller را در قسمت کامند متلب تایپ کنید، سپس در پنجره ی باز شده پکیج زیر را انتخاب کنید



سیستم تشخیص صورت به دو صورت و در دو حالت کار می کند: تشخیص و تعقیب. در حالت تشخیص ما از vision.CascadeObjectDetector برای تشخیص صورت در فریم استفاده می کنیم، اگر

درست تشخیص داده شد سپس گوشه های صورت تشخیص داده می شود و بعد با vision.PointTracker به حالت دنبال کردن صورت وارد می شویم.

در حالت تعقیب به وسیله ی point tracker نقطه ها را دنبال می کنیم. در مرحله تعقیب نقطه ها بعضی از آن ها از دست می رود. اگر تعداد نقاط زیر یک آستانه سقوط کند، به این معنی است که صورت دیگر ردیابی نمی شود و باید دوباره به حالت تشخیص برگشت.

توضیح خط به خط برنامه و مراحل

ایجاد اشیاء برای تشخیص چهره، نقاط ردیابی، دستیابی و نمایش فریم های ویدئویی:

```
% ساخت شیء تشخیص چهره.  
faceDetector = vision.CascadeObjectDetector();  
  
% دنبال کننده ی نقاط.  
pointTracker = vision.PointTracker('MaxBidirectionalError', 2);  
  
% شیء دوربین.  
cam = webcam();  
  
% ضبط یک قاب برای اندازه آن.  
videoFrame = snapshot(cam);  
frameSize = size(videoFrame);  
  
% شیء پخش ویدئو را ایجاد کنید.  
videoPlayer = vision.VideoPlayer('Position', [100 100 [frameSize(2),  
frameSize(1)]+30]);
```

تشخیص و ردیابی:

فریم های ویدیویی را از وب کم در یک حلقه برای شناسایی و ردیابی یک چهره ضبط و پردازش کنید. حلقه برای ۱۰۰۰ فریم اجرا خواهد شد یا تا زمانی که پنجره پخش ویدئو بسته شود.

```
runLoop = true;
numPts = 0;
frameCount = 0;

while runLoop && frameCount < 400

    % فریم بعدی را بگیرید.
    videoFrame = snapshot(cam);
    videoFrameGray = rgb2gray(videoFrame);
    frameCount = frameCount + 1;

    if numPts < 10
        % حالت تشخیص.
        bbox = faceDetector.step(videoFrameGray);

        if ~isempty(bbox)
            % نقاط گوشه ای را در داخل منطقه شناسایی پیدا کنید.
            points = detectMinEigenFeatures(videoFrameGray, 'ROI', bbox(1, :));

            % دوباره ردیابی نقطه.
            xyPoints = points.Location;
            numPts = size(xyPoints,1);
            release(pointTracker);
            initialize(pointTracker, xyPoints, videoFrameGray);

            % یک کپی از نقاط را ذخیره کنید.
            oldPoints = xyPoints;

            bboxPoints = bbox2points(bbox(1, :));

            % [x1 y1 x2 y2 x3 y3 x4 y4] تبدیل گوشه های جعبه به
            % insertShape فرمت مورد نیاز توسط
            bboxPolygon = reshape(bboxPoints', 1, []);

            % یک جعبه محدود را در اطراف چهره شناسایی نمایش دهید.
            videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

            % نمایش گوشه های مشخص شده.
            videoFrame = insertMarker(videoFrame, xyPoints, '+', 'Color', 'white');
        end
    end
end
```

```

else
    % حالت ردیابی.
    [xyPoints, isFound] = step(pointTracker, videoFrameGray);
    visiblePoints = xyPoints(isFound, :);
    oldInliers = oldPoints(isFound, :);

    numPts = size(visiblePoints, 1);

    if numPts >= 10
        % برآورد تحول هندسی بین نقاط قدیمی
        % و نقاط جدید.
        [xform, oldInliers, visiblePoints] = estimateGeometricTransform(...
            oldInliers, visiblePoints, 'similarity', 'MaxDistance', 4);

        % درخواست تحول به جعبه.
        bboxPoints = transformPointsForward(xform, bboxPoints);

        % [x1 y1 x2 y2 x3 y3 x4 y4] تبدیل گوشه های جعبه به
        % insertShape فرمت مورد نیاز توسط.
        bboxPolygon = reshape(bboxPoints', 1, []);

        % یک جعبه محدود در اطراف چهره ای که ردیابی شده است نمایش داده می شود.
        videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon, 'LineWidth', 3);

        % نمایش نقاط ردیابی.
        videoFrame = insertMarker(videoFrame, visiblePoints, '+', 'Color', 'white');

        % تنظیم مجدد نقاط.
        oldPoints = visiblePoints;
        setPoints(pointTracker, oldPoints);
    end

end

step(videoPlayer, videoFrame);

% بررسی کنید که پنجره پخش ویدئو بسته شده است.
runLoop = isOpen(videoPlayer);
end

% پاک کردن.
clear cam;
release(videoPlayer);
release(pointTracker);
release(faceDetector);

```