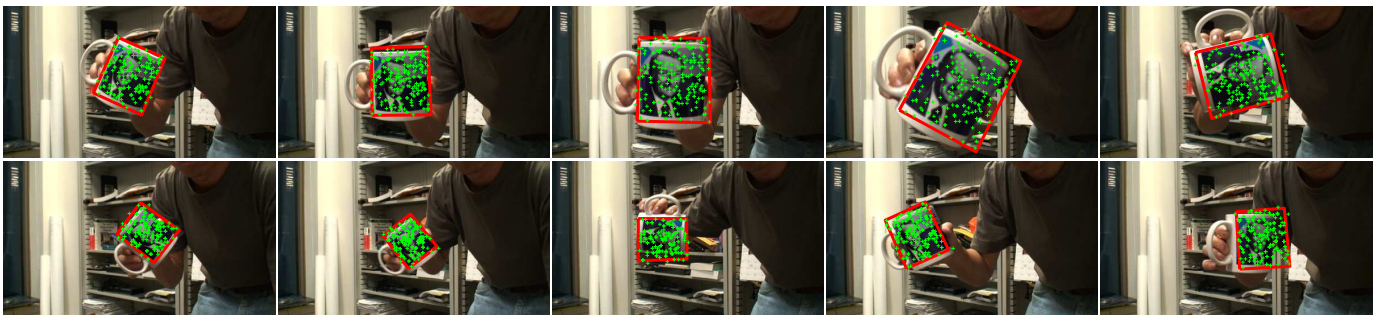
Model Fitting



이 과제에서 우리는 물체 감지를 위해 RANSAC를 사용합니다. 이미지를 사용하여 코드를 테스트 할 수 있습니다. 코드 및 이미지 처리 결과가 포함 된 보고서를 작성해야합니다. 보고서를 PDF로 제출하십시오. Matlab 코드는 별도로 칠판에 제출해야합니다. 이 과제는 2011 년 10 월 22 일에 만료 예정입니다.

I. Object Localization Using RANSAC [60 points]

A. SIFT Feature

클래스에서 논의한 방법을 기반으로 이미지의 객체를 감지하는 RANSAC 메소드를 구현합니다. SIFT 피쳐 검출기는 과제와 함께 첨부됩니다. Matlab 함수 sift는 SIFT 실행 파일을 래핑하여 매개 변수 조작을 단순화합니다. Matlab 코드와 바이너리 실행 파일을 모두 작업 디렉토리에 복사하십시오. 이 코드는 Windows와 Mac 모두에서 작동해야합니다. SIFT에 대한 자세한 내용은 David Lowe의 SIFT 데모 웹 페이지를 확인하십시오.
http://people.cs.ubc.ca/~lowe/keypoints/. SIFTPP 웹 페이지는
http://www.vlfeat.org/~vedaldi/assets/siftpp/sift.html에 있습니다. SIFTPP는 데이비드 로우 (David Lowe)의 SIFT를 구현 한 것으로 소스 코드가 첨부되어 있습니다.

The interface of `sift` is

```
[image, descriptors, locs] = sift(imageFile)
%
% This function reads an image and returns its SIFT keypoints.
%   Input parameters:
%     imageFile: the file name for the image.
%
%   Returned:
%     image: the image array in double format
%     descriptors: a K-by-128 matrix, where each row gives an invariant
%         descriptor for one of the K keypoints.  The descriptor is a vector
%         of 128 values normalized to unit length.
%     locs: K-by-4 matrix, in which each row has the 4 values for a
%         keypoint location (row, column, scale, orientation).  The
%         orientation is in radians.
%   Credits: this code is adapted from David Lowe's SIFT demo and wraps SIFTPP and
%            siftWin32
```

SIFT 기능 위치에는 하위 픽셀 정밀도가 있습니다.
blackboard나 코스 웹 사이트에서 테스트 이미지를 다운로드 할 수도 있습니다.
데이터 세트에서 하나의 이미지를 템플릿으로 선택하고 다른 이미지에서 선택한 객체를 감지 할 수 있습니다.

B. RANSAC for Object Matching

다음 작업을 완료해야합니다.
1) 템플릿 이미지를 열고 관심 분야를 선택하십시오.
영역에서 SIFT 특징점, SIFT 특징 벡터 및 선택된 경계점을 찾습니다. Matlab 스크립트는 다음과 같습니다.

```
im = imread(template_name);
```

```
im = im2double(im);
imwrite(im, 'tmp.pgm'); %sift only accepts pgm
imshow(im);

[mask, xc, yc] = roipoly(im);
mask = im2double(mask);
save mask.mat mask

% Compute the sift features in the template image
[image1, des1, loc] = sift('tmp.pgm');

% Select sift in the region of interest
z = [];
ff = [];
for n = 1 : size(loc,1)
    r = floor(loc(n,2));
    c = floor(loc(n,1));
    if r > 0 & c > 0 & mask(r, c) == 1
        z = [z; floor(loc(n,1:2))];
        ff = [ff; des1(n,:)];
    end
end
hold on;
plot(z(:,1), z(:,2), '+');

p_src = unique(z, 'rows'); % the sift points in ROI
[ism, id] = ismember(p_src, z, 'rows');
feature1 = ff(id,:); % the sift features in ROI
```

스크립트가 끝나면 p_src에서 위치를 선택하고 feature1에서 기능을 선택합니다.
2) 대상 이미지를 열고 SIFT 기능을 계산하고 각 템플릿 타겟 피처 쌍의 유사성을 비교합니다.
유사한 쌍을 찾는 데 경험적 방법(추론)을 사용합니다. 경험적 방법(추론)에서는 최상의 일치 비용이 두 번째로 가장 일치하는 값의 일정 비율 미만인 일치하는 쌍만 유지합니다.

```
im = imread(target_image);
imwrite(im, 'tmp.pgm');
[image2, feature2, loc2] = sift('tmp.pgm');

%Compute the similarity of the template with the target sift points
feature2 = feature2';
dotprods = feature1 * feature2;        % Computes vector of dot products
cc = acos(dotprods)/pi*2;  % the matching costs

p_des = loc2(:,1:2); % the target point locations

% Find some promising point-point pairs
dx = [];
dy = [];
for n = 1 : size(p_src,1)
    [vals,indx] = sort(cc(n,:));
    if (vals(1) < 0.85*vals(2))
       dx = [dx; p_src(n,1), p_src(n,2), p_des(indx(1),1)];
       dy = [dy; p_src(n,1), p_src(n,2), p_des(indx(1),2)];
    end
end
```

위의 절차에서 dx는 처음 두 개의 열에 소스 포인트 x와 y를 포함하고 세 번째 열에는 대상 x를 포함합니다. dy는 첫 번째 두 열과 세 번째 열의 대상 y에서 소스 점 x와 y를 포함합니다.
3) RANSAC을 사용하여 템플릿 포인트를 타겟 포인트와 정렬합니다. dx와 dy를 사용하여 3 개의 템플릿 - 대상 점 쌍을 무작위로 선택해야합니다. 다음 함수를 사용하여 3 개의 다른 난수를 생성 할 수 있습니다.

```
function r3 = rand3(N)
% rand3 randomly generates 3 different numbers in 1 to n.
% r3 contains the the 3 random numbers
   t = 1 : N;
   for n = 1 : 3
       r = floor(rand(1,1)*(N-n+1)) + n;
       tmp = t(n);
       t(n) = t(r);
       t(r) = tmp;
   end
   r3 = t(1:3);
```

r3에 3 개의 난수가 있다고 가정하면, 템플릿에서 타겟으로의 변환은 Matlab 함수를 사용하여 계산할 수 있습니다

```
t = cp2tform(in_points, out_points, 'similarity');
```

여기서 in_points = [dx (r3,1 : 2)] 및 out_points = [dx (r3,3) dy (r3,3)]입니다.
그런 다음 템플릿 포인트를 대상 이미지에 투사 할 수 있습니다 (예 :

```
p_proj = [p_src, ones(size(p_src,1), 1)] * t.tdata.T;
```

p_proj의 처음 두 열은 투영 된 점의 x 및 y 좌표입니다. cp2tform 함수에서 'affine'을 사용할 수도 있습니다.
각 템플릿 점을 대상 이미지에 매핑 한 후 가장 가까운 대상 점과 비교하고 전체 일치 비용을 유지합니다.
N, .e.g에 대해이 절차를 반복하십시오. 100 번, 최소 비용으로 하나를 선택하십시오.
템플릿을 대상 이미지로 투영하고 대상 객체를 찾습니다.

4) 코드를 테스트하십시오. 자신의 테스트 이미지를 사용하는 경우 제출에도 이미지를 포함시켜야합니다.

II. Skeleton Code

The complete skeleton code is as follow.

```
% Read the template image and select a region of interest
template_name = './images/template.jpg';

im = imread(template_name);
im = im2double(im);
imwrite(im, 'tmp.pgm');
imshow(im);

[mask, xc, yc] = roipoly(im);
mask = im2double(mask);
save mask.mat mask

% Compute the sift features in the template image
[image1, des1, loc] = sift('tmp.pgm');

% Select sift in the region of interest
z = [];
ff = [];
for n = 1 : size(loc,1)
    r = floor(loc(n,2));
    c = floor(loc(n,1));
    if r > 0 & c > 0 & mask(r, c) == 1
        z = [z; floor(loc(n,1:2))];
        ff = [ff; des1(n,:)];
    end
end
hold on;
plot(z(:,1), z(:,2), '+');

p_src = unique(z, 'rows'); % the sift points in ROI
[ism, id] = ismember(p_src, z, 'rows');
```

```
feature1 = ff(id,:); % the sift features in ROI

%------A big loop ---------%
for videon = 1 : 40
     videon
     fname2 = sprintf('./images/target%d.jpg', videon);
     im = imread(fname2);
     imwrite(im, 'tmp.pgm');
     [image2, feature2, loc2] = sift('tmp.pgm');

     %Compute the similarity of the template with the target sift points
     feature2 = feature2';
     dotprods = feature1 * feature2;        % Computes vector of dot products
     cc = acos(dotprods)/pi*2;  % the matching costs

     p_des = loc2(:,1:2); % the target point locations

     % Find some promising point-point pairs
     dx = [];
     dy = [];
     for n = 1 : size(p_src,1)
         [vals,indx] = sort(cc(n,:));
         if (vals(1) < 0.85*vals(2))
            dx = [dx; p_src(n,1), p_src(n,2), p_des(indx(1),1)];
            dy = [dy; p_src(n,1), p_src(n,2), p_des(indx(1),2)];
         end
     end


     %--- ransac ---------------%

         Here is your ransac code

     %--------------------------%

         Show the results

     %--------------------------%
end
```