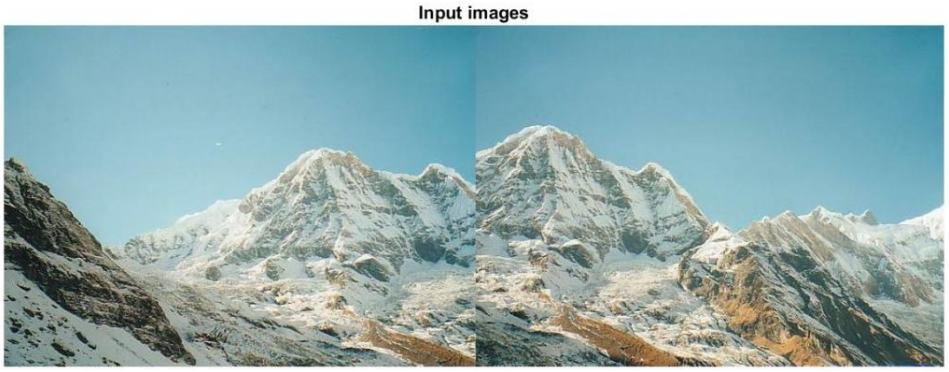
Table of Contents

Image Stitching - Stéphane Maillot - June 1st

```
close all
% name = 'results\mountains';
```

Import pictures

```
Im1 = cell(1,4);
Im2 = cell(1,4);
Im1{1} = imread('demoimages\beach1.jpg');
Im1{2} = imread('demoimages\jungfrau1.jpg');
Im1{3} = imread('demoimages\mountains1.jpg');
Im2{1} = imread('demoimages\beach2.jpg');
Im2{2} = imread('demoimages\jungfrau2.jpg');
Im2{3} = imread('demoimages\mountains2.jpg');
for k=1:3

    I1 = single(rgb2gray(Im1{k}))/255;
    I2 = single(rgb2gray(Im2{k}))/255;

%     f = figure('Position', [100, -100, 500, 1500])
%     subplot(511)
    figure
    imshow([Im1{k}(1:min(size(Im1{k}, 1), size(Im2{k}, 1)), :, :)
 Im2{k}(1:min(size(Im1{k}, 1), size(Im2{k}, 1)), :, :)])
    title('Input images')
```



Warning: Image is too big to fit on screen; displaying at 67%

Features detection

```
points1 = detectSURFFeatures(I1);
points2 = detectSURFFeatures(I2);

[features1,valid_points1] = extractFeatures(I1,points1);
[features2,valid_points2] = extractFeatures(I2,points2);
```

Features matching

```
distRatio = 0.8;
matches = zeros(1,length(features1));
```

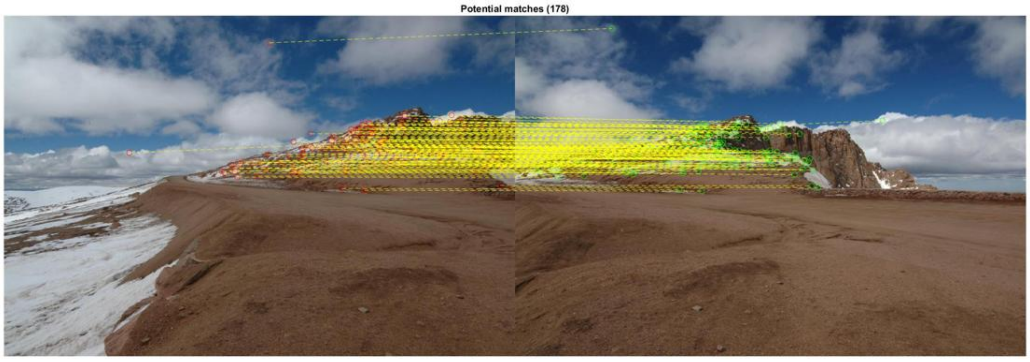
Find valid feature points index

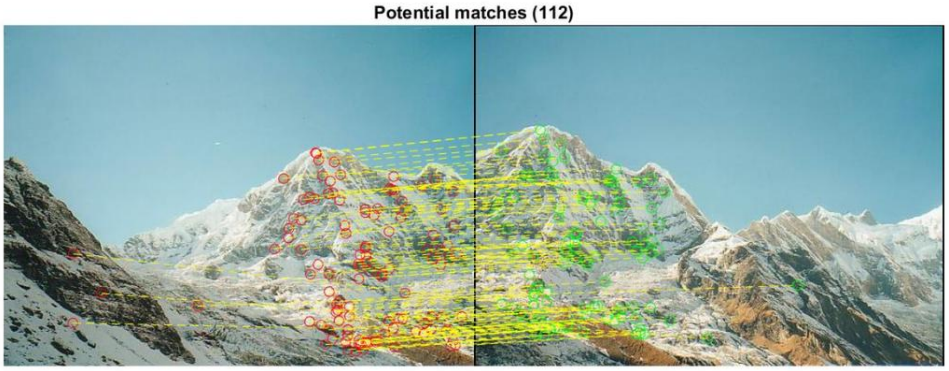
```
for i=1:length(features1)

    % I found this angle technique on internet and it works better
than
    % distances for these SURF features
    dotprods = features1(i,:) * features2';
```

```
        [angle,index] = sort(acos(dotprods));
        if (angle(1) < distRatio * angle(2))
           matches(i) = index(1);
        end

%         dist = sqrt(sum(features2 - features1(i,:), 2));
%         [dist, index] = sort(dist);
%         if dist(1) < distRatio * dist(2)
%             matches(i) = index(1);
%         end
    end
```

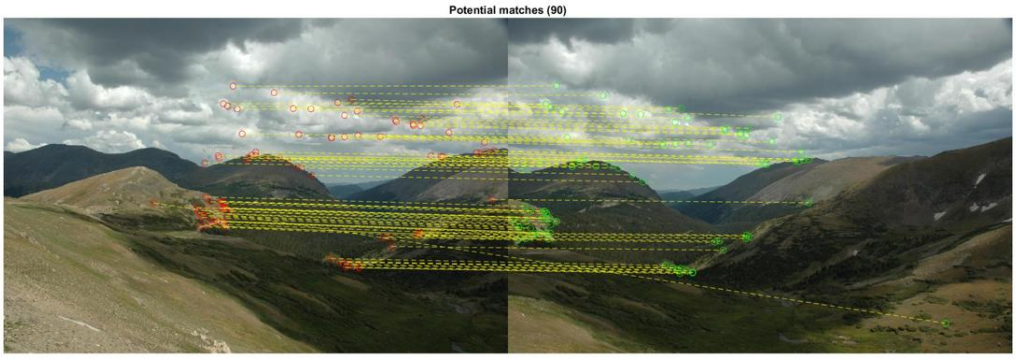
Extract corresponding points

```
    matched1 = [];
    matched2 = [];
    for i=1:length(matches)
        if matches(i) > 0
            matched1 = [matched1 ; valid_points1.Location(i,:)];
            matched2 = [matched2 ;
 valid_points2.Location(matches(i),:)];
        end
    end

    matched = matchFeatures(features1, features2);
 matched1 = valid_points1(matched(:,1),:);
    matched1 = matched1.Location;
    matched2 = valid_points2(matched(:,2),:);
    matched2 = matched2.Location;

    % matlab automatic feature matching
    indexPairs = matchFeatures(features1,features2);
    matched1 = valid_points1(indexPairs(:,1),:);
    matched1 = matched1.Location;
    matched2 = valid_points2(indexPairs(:,2),:);
    matched2 = matched2.Location;

%     subplot(512)
    figure

 showMatchedFeatures(Im1{k},Im2{k},matched1,matched2,'montage','PlotOptions',
{'ro','go','y--'});
    title(strcat('Potential matches (',
 num2str(length(matched1)), ')'))
```





Warning: Image is too big to fit on screen; displaying at 67%



RANSAC

```
n = length(matched1);
nPts = 4;
iter = 172;
minInliersRatio = 0.5;
minDist = 3;
concensus = round(minInliersRatio*n);
```

```
    n_inliers = zeros(1,iter);
    H_array = cell(1,iter);

    for i = 1:iter
```

choose random points

```
        randIndex =  randperm(n);
        randIndex = randIndex(1:nPts);
```

compute the corresponding homography

```
        h = DLT(matched1(randIndex,:),matched2(randIndex,:));
```

compute distances

```
        proj = h*[matched1' ; ones(1,length(matched1))];
        proj = proj(1:2,:)./repmat(proj(3,:),2,1);
        dist = sum((matched2'-proj).^2,1);
```

count inliers

```
        inlier1 = find(dist < minDist);
        n_inliers(i) = length(inlier1);
```

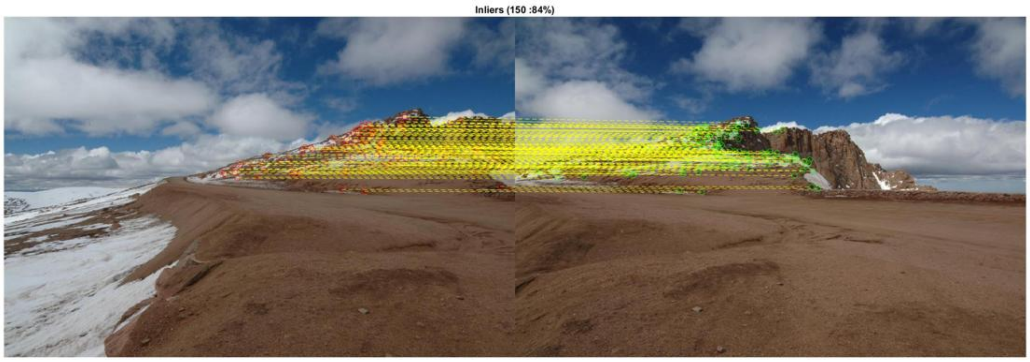
save this solution if there is enough inliers

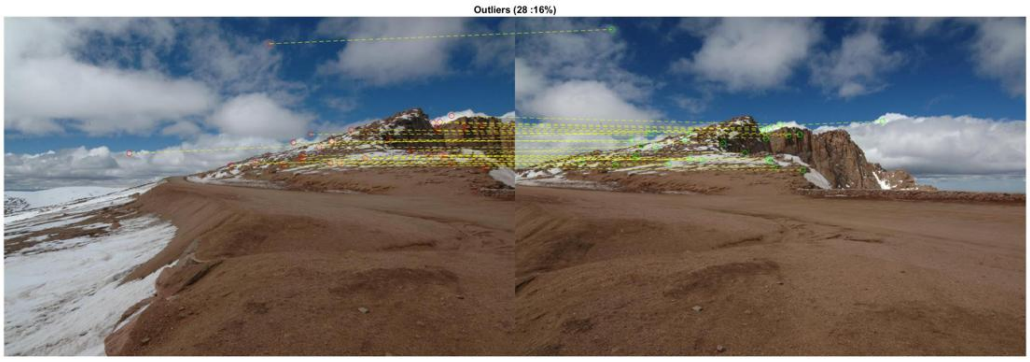
```
        if n_inliers(i) > concensus
            H_array{i} = DLT(matched1(inlier1,:),matched2(inlier1,:));
        end

    end
```

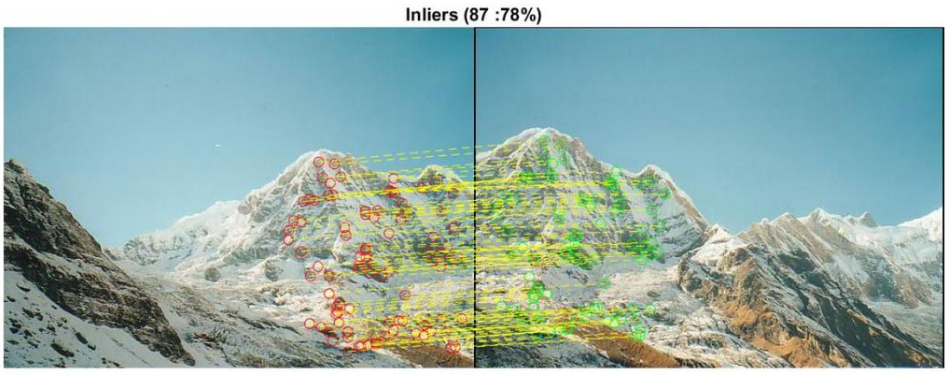
keep the best solution

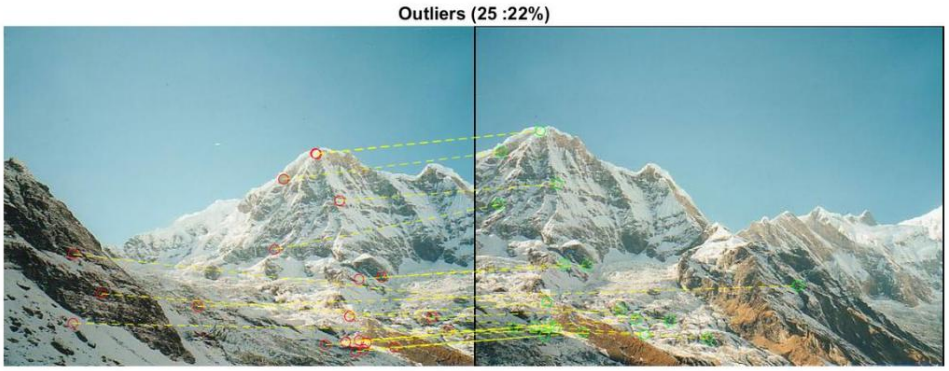
```
    [val,index] = max(n_inliers);
    h = H_array{index};
    dist = calcDist(h,matched1,matched2);
    inliers = find(dist < minDist);
    outliers = find(dist >= minDist);
    matched1_in = matched1(inliers, :);
    matched2_in = matched2(inliers, :);
    matched1_out = matched1(outliers, :);
    matched2_out = matched2(outliers, :);

%     subplot(513)
    figure

 showMatchedFeatures(Im1{k},Im2{k},matched1_in,matched2_in,'montage','PlotOptions'
{'ro','go','y--'});
    title(strcat('Inliers (', num2str(n_inliers(index)), ' : ',
 num2str(round(100*n_inliers(index)/length(matched1))), '%)'))

%     subplot(514)
```

```
    figure

 showMatchedFeatures(Im1{k},Im2{k},matched1_out,matched2_out,'montage','PlotOption
{'ro','go','y--'});
    title(strcat('Outliers (', num2str(length(matched1) -
 n_inliers(index)), ' : ', num2str(round(100*(1-n_inliers(index)/
length(matched1)))), '%)'))
```

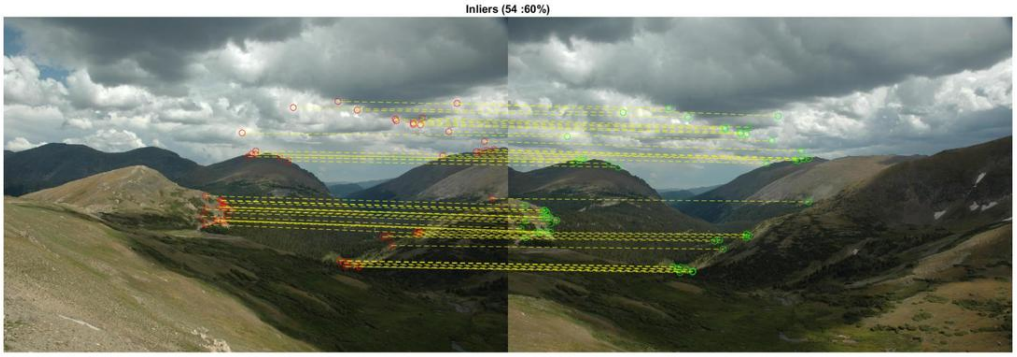


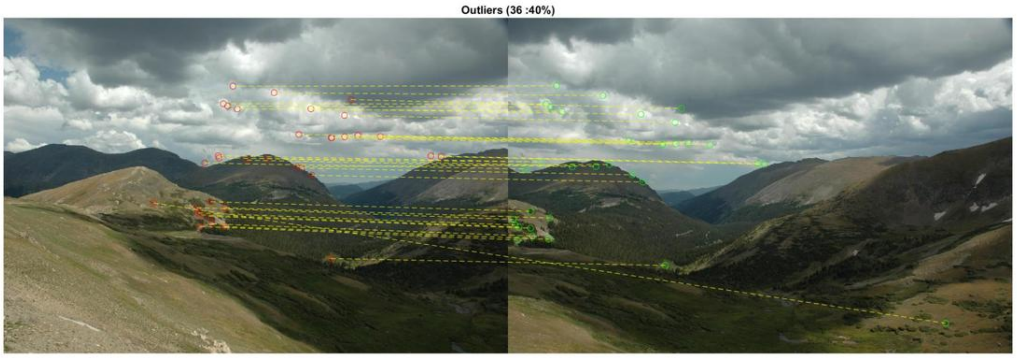







```
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Image is too big to fit on screen; displaying at 67%
```

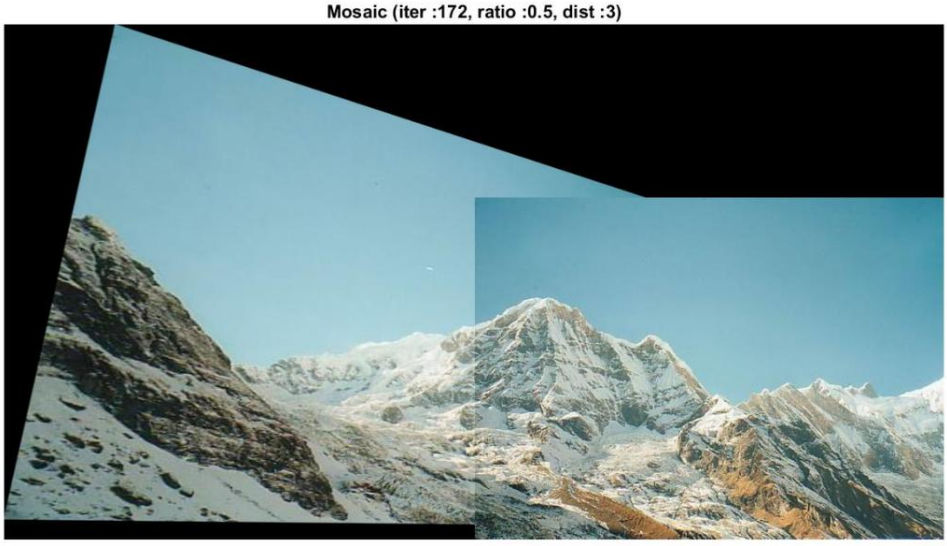


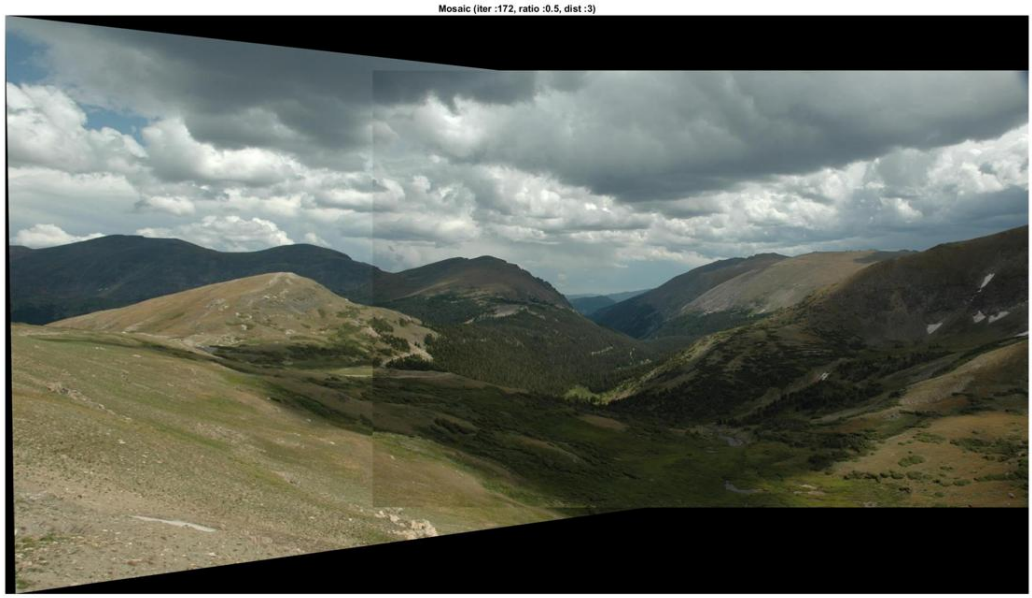


Mosaic

```
%     subplot(515)
    figure
    imgout=make_mosaic(Im2{k},Im1{k},h);
    imshow(imgout)
    title(strcat('Mosaic (', 'iter : ', num2str(iter), ', ratio : ',
 num2str(minInliersRatio), ', dist : ', num2str(minDist), ')'))
```

```
%     saveas(f, strcat(name, '_fig.jpg'));
%     imwrite(imgout, strcat(name, '_mosaic.jpg'));
```






```
end
```

Published with MATLAB® R2017a