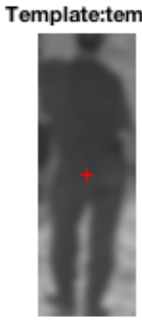
```
%============================== findTemplate ==============================
%
%  A script to find the location of a template object within a given image
%  when some approxiamte location is known.
%
%============================== findTemplate ==============================


%==[1] Perform a little smoothing of the image to aid the gradient descent.
%               It is good to also smooth the template a little bit.  The image
%               should be smoothed noticeably, while the template just a bit.
%

% Choose template and image
I = I1;
t = temp2;
pos = [640,840];                                        % Initial guess of position.

tI = imgaussfilt(t,2);  % Filter template a little bit.
fI = imgaussfilt(I,5);  % Filter image some.

tnsq = sqrt(sumsqr(tI));                          % Compute "magnitude" of the template.
                                                                        % Useful to incorporate into c.

c = 1/(30*tnsq);                                                        % Step adjustment factor.

nsteps = 200;                                           % Number of steps in the gradient descent.

I_name = 'I1';
t_name = 'temp2';
fprintf('Image: %s\n',I_name);
fprintf('Template: %s\n',t_name);
fprintf('Parameters:\n');
fprintf('(c=%f,nsteps=%d,pos=[%d,%d])\n',c,nsteps,pos(1),pos(2));

%==[2] Plot the template and the image.

figure(3);
  imshow(tI);
  title(sprintf('Template:%s', t_name));
  hold on;
  plot( size(tI,2)/2, size(tI,1)/2, 'r+');
  hold off;
  % Placed a plus at the template center.

figure(2);
  imagesc(I);
  colormap('Gray');
  title(sprintf('Image:%s',I_name));
  hold on;
  plot(pos(1), pos(2), 'go');

for ii=1:nsteps
  dpos = gradTempMatch(double(tI), double(fI), pos);
  pos = pos - c.*dpos';
  plot(pos(1), pos(2), 'r.');
  drawnow;
end

plot(pos(1), pos(2), 'rx');
```

```
fprintf('Final Point:[%f,%f]\n',pos(1),pos(2));
```

```
Image: I1
Template: temp2
Parameters:
(c=0.000024,nsteps=200,pos=[640,840])
Final Point:[586.983502,672.585061]
```



Published with MATLAB® R2016b