

```

function v = LK_alg(I1, I2, lambda, mask, ...
    v_initial, num_iterations)
%LK_ALG Runs the Lucas Kanade iterative algorithm for calc. optical flow
% Parameters
% -----
% I1 - the first frame of an image
% I2 - the second frame of an image (same size as I1)
% lambda - the noise variance to prior variance ratio (scalar)
% mask - area of the image to sum upon (same size as I1)
% v_initial - initial guess for the velocity (2d vector)
% num_iterations - ... y'know
% Returns
% -----
% v - the computed velocity (2d vector)

At = zeros(size(I1,1), size(I1,2), 4);
Bt = zeros(size(I1,1), size(I1,2), 2);
v = v_initial;
for i = 1:num_iterations
    [I2w, warpMask] = warp(I2, v);
    newMask = mask .* warpMask;
    [Ix, Iy, It] = ImageDerivatives(I1, I2w);
    Ix = Ix .* newMask;
    Iy = Iy .* newMask;
    It = It .* newMask;
    At(:, :, 1) = Ix.^2;
    At(:, :, 2) = Ix .* Iy;
    At(:, :, 3) = At(:, :, 2);
    At(:, :, 4) = Iy.^2;
    Bt(:, :, 1) = Ix .* It;
    Bt(:, :, 2) = Iy .* It;
    A = reshape(sum(sum(At, 1), 2), 2, 2) + eye(2).*lambda;
    B = -reshape(sum(sum(Bt, 1), 2), 2, 1);
    v = v + A \ B;
end

end

```