

### **When does it perform well/poorly?**

The tracker performs well for slow, deliberate movements.

This is likely due to noise caused by motion blur on fast movement in the frame giving noisy gradients.

The tracker is also less prone to drifting when tracking smaller areas. The main reason is likely since a larger area has many more pixels, there is much more noisy pixel data.

With large iterations (more than 2 or 3) and a step-size of 1, the tracker overshoots the tracked area if the area had small movement. Similarly, small iteration values with a small (1-5) stepsize yielded trackers that would lose the tracked object if the movement was not slow.

Larger stepsizes, more than 5, would make the tracker move slowly, but the effect can be reduced with increased

maximum iterations for the calculation. However, once we cross a step-size of about 14, the tracker drifts significantly,

and does not follow the motion of the tracked object well at any maximum iterations.

The tracker does not work well if the environment is not well lit,

as the camera iso increases and becomes more light sensitive. This causes individual pixels to become noisy and the tracker erroneously detects that as motion.

If the image is too bright, similar issues arise where the correct motion is not detected.

If there is a shadow in the image and the tracked object goes between dark and light areas of the image, the tracker accumulates errors.

If the tracked object is a single colour, the tracker performs significantly better than with textured areas.

### **What type of image processing operation might improve performance?**

Applying a smoothing filter, such as a gaussian blur, can mitigate errors from noisy pixels.

Since it averages movement with surrounding pixels, it smooths out the errors from noisy pixels.

**What would be the advantages/disadvantages of using other warps (for example affine warps)?**

The weaknesses with the translational x-y tracker comes with its inability to detect rotation or scaling.

So if the camera is not directly overhead, it does not detect the perspective changes of the object such as stretching,

instead it simply detects it as motion. When rotating an object, the tracker does not detect that at all, and it

simply allows error accumulation in the tracker.

Using an affine warp would allow the tracker to deal with stretching and rotation on top of the x-y directional motion.

However, this would come at a computational cost, as the number of parameters we need to deal with increases from a 2x2 matrix,

to a 2x3 matrix corresponding to adding rotation and scaling.

(Code submission in the next page)

Code:

```
%ims = image_cap(100,0);
%videotracker(1,2,ims);
%livetracker(8,2)
pyramidTracker(1,1)
function [ img_seq ] = image_cap( n_frames , time)
% Takes a sequence of images over some time
interval and returns a cell of
% images converted to grayscale
% Detailed explanation goes here
img_seq = cell(n_frames, 1);
%img_seq = {};
for i = 1:n_frames
    [success, im] = mexMTF2('get_frame');
    img_seq{i}=rgb2gray(im);
    imshow(im);
    pause(time);
end
end

function livetracker(stepsize, maxiters)

    [~, im] = mexMTF2('get_frame');
    im1 = rgb2gray(im);
    dim1 = double(im1);
    imshow(im);
    rect = getrect();

    rectangle('Position',rect,'LineWidth',2,'LineStyle'
, '--')
    lowx = rect(1);
    lowy = rect(2);
    w = rect(3);
    h = rect(4);
    [imlen, imwidth] = size(im1);

    Xi = 1:imwidth;
```

```
Yi = 1:imlen;
[X, Y] = meshgrid(Xi, Yi);

while 1
    [~, im] = mexMTF2('get_frame');
    im2 = rgb2gray(im);
    dim2 = double(im2);
    imdiff = dim2 - dim1; % temporal gradient
    %imdiff(abs(imdiff) < 60) = 0;
    [dX, dY] = gradient(dim2); %spatial
gradient

    for i=1:maxiters

        XQ = lowx:stepsize:lowx+w;
        YQ = lowy:stepsize:lowy+h;
        [Xq, Yq] = meshgrid(XQ, YQ);

        %Xi(lowx:stepsize:lowx+w), Yi(lowy:stepsize:lowy+h))
        ;

        VqX = interp2(X, Y, dX, Xq, Yq);
        VqY = interp2(X, Y, dY, Xq, Yq);
        Vqt = interp2(X, Y, imdiff, Xq, Yq);
        VqX = reshape(VqX, [], 1);
        VqY = reshape(VqY, [], 1);
        Vqt = reshape(Vqt, [], 1);

        dI = [VqX VqY];

        V = (dI'*dI)\( dI'*Vqt);

        lowx = lowx - V(1);
        lowy = lowy - V(2);

        if norm(V,2) < 0.001
            break
    end
end
```

```
        end
    end
    hold on;
    imshow(im2)
    rectangle('Position',[lowx lowy w
h], 'LineWidth',2, 'LineStyle','--')
    drawnow;
    hold off;
    %im1 = im2;
    dim1 = dim2;
end

end

function videotracker(stepsize, maxiters,
imageCells)
    n_frames = length(imageCells);
    if n_frames == 0
        error('Video cells is of size zero');
    end
    im1 = imageCells{1};
    dim1 = double(im1);
    imshow(im1);
    rect = getrect();

    rectangle('Position',rect, 'LineWidth',2, 'LineStyle'
, '--')
    lowx = rect(1);
    lowy = rect(2);
    w = rect(3);
    h = rect(4);
    [imlen, imwidth] = size(im1);

    Xi = 1:imwidth;
    Yi = 1:imlen;
    [X, Y] = meshgrid(Xi, Yi);
```

```
for i=2:n_frames

    im2 = imageCells{i};
    dim2 = double(im2);
    imdiff = dim2 - dim1; % temporal gradient
    %imdiff(abs(imdiff) < 60) = 0;
    [dX, dY] = gradient(dim2); %spatial
gradient

    for i=1:maxiters

        XQ = lowx:stepsize:lowx+w;
        YQ = lowy:stepsize:lowy+h;
        [Xq, Yq] = meshgrid(XQ, YQ);

        %Xi(lowx:stepsize:lowx+w),Yi(lowy:stepsize:lowy+h))
        ;

        VqX = interp2(X,Y,dX,Xq,Yq);
        VqY = interp2(X,Y,dY,Xq,Yq);
        Vqt = interp2(X,Y,imdiff,Xq,Yq);
        VqX = reshape(VqX, [], 1);
        VqY = reshape(VqY, [], 1);
        Vqt = reshape(Vqt, [], 1);

        dI = [VqX VqY];

        V = (dI'*dI)\( dI'*Vqt);

        lowx = lowx - V(1);
        lowy = lowy - V(2);

        if norm(V,2) < 0.01
            break
        end
    end
end
hold on;
```

```
        imshow(im2)
        rectangle('Position',[lowx lowy w
h], 'LineWidth',2, 'LineStyle','--')
        drawnow;
        hold off;
        %im1 = im2;
        dim1 = dim2;
    end

end

function pyramidTracker(stepsize, maxiters)

[~, im] = mexMTF2('get_frame');
    im1 = rgb2gray(im);
    dim1 = double(im1);
    imshow(im);
    rect = getrect();

    rectangle('Position',rect, 'LineWidth',2, 'LineStyle'
, '--')
        lowx = rect(1);
        lowy = rect(2);
        w = rect(3);
        h = rect(4);
        [imlen, imwidth] = size(im1);

        Xi = 1:imwidth;
        Yi = 1:imlen;
        [X, Y] = meshgrid(Xi, Yi);
        [X2, Y2] = meshgrid(Xi(1:imwidth/2),
Yi(1:imlen/2));
        [X3, Y3] = meshgrid(Xi(1:imwidth/4),
Yi(1:imlen/4));
        while 1
            [~, im] = mexMTF2('get_frame');
            im2 = rgb2gray(im);
```

```
dim2 = double(im2);
imdifff = dim2 - dim1; % temporal gradient
imdifff2 = impyramid(imdifff, 'reduce');
imdifff3 = impyramid(imdifff2, 'reduce');
%imdifff(abs(imdifff) < 60) = 0;
[dX, dY] = gradient(dim2); %spatial
gradient
rdim2 = impyramid(dim2, 'reduce');
[dX2, dY2] = gradient(rdim2); %spatial
gradient
rrdim2 = impyramid(rdim2, 'reduce');
[dX3, dY3] = gradient(rrdim2); %spatial
gradient
for i=1:maxiters
    XQ = lowx:stepsize:lowx+w;
    YQ = lowy:stepsize:lowy+h;
    [Xq, Yq] = meshgrid(XQ, YQ);

    XQ2 = lowx/2:stepsize:(lowx+w)/2;
    YQ2 = lowy/2:stepsize:(lowy+h)/2;
    [Xq2, Yq2] = meshgrid(XQ2, YQ2);

    XQ3 = lowx/4:stepsize:(lowx+w)/4;
    YQ3 = lowy/4:stepsize:(lowy+h)/4;
    [Xq3, Yq3] = meshgrid(XQ3, YQ3);

    %Xi(lowx:stepsize:lowx+w),Yi(lowy:stepsize:lowy+h))
    ;

    VqX = interp2(X,Y,dX,Xq,Yq);
    VqY = interp2(X,Y,dY,Xq,Yq);
    Vqt = interp2(X,Y,imdifff,Xq,Yq);
    VqX = reshape(VqX, [], 1);
    VqY = reshape(VqY, [], 1);
    Vqt = reshape(Vqt, [], 1);
    dI = [VqX VqY];
    V = (dI'*dI)\( dI'*Vqt);
```



```
lowx = lowx - V(1);
lowy = lowy - V(2);

VqX = interp2(X2,Y2,dX2,Xq2,Yq2);
VqY = interp2(X2,Y2,dY2,Xq2,Yq2);
Vqt = interp2(X2,Y2,imdiff2,Xq2,Yq2);
VqX = reshape(VqX, [], 1);
VqY = reshape(VqY, [], 1);
Vqt = reshape(Vqt, [], 1);
dI = [VqX VqY];
V = (dI'*dI)\( dI'*Vqt);
lowx = lowx - V(1);
lowy = lowy - V(2);

VqX = interp2(X3,Y3,dX3,Xq3,Yq3);
VqY = interp2(X3,Y3,dY3,Xq3,Yq3);
Vqt = interp2(X3,Y3,imdiff3,Xq3,Yq3);
VqX = reshape(VqX, [], 1);
VqY = reshape(VqY, [], 1);
Vqt = reshape(Vqt, [], 1);
dI = [VqX VqY];
V = (dI'*dI)\( dI'*Vqt);
lowx = lowx - V(1);
lowy = lowy - V(2);

if norm(V,2) < 0.001
    break
end
end
disp("here")
hold on;
imshow(im2)
rectangle('Position',[lowx lowy w
h], 'LineWidth',2, 'LineStyle','--')
drawnow;
hold off;
```

Imtihan Ahmed

Cmput 428

Lab 2

```
        %im1 = im2;  
        dim1 = dim2;  
    end  
  
end
```