CS 6320, 3D Computer Vision
Spring 2015, Prof. Guido Gerig

Assignment 4: Optical Flow / Structured Light

Out:	Mon 04-06-2015
Due:	Mon 04-20-2015 midnight (theoretical and practical parts, submission to CANVAS)
TA:	Padmashree Teeka
Office hours	TA MEB 3115: Tue 3-5pm, Thu 11am to 1pm, Instructor Mo/Wed 3pm-5pm

Required Readings: Handout Klette Chapter 9: Structured Light
Material on motion and optic flow (Horn: Tutorial)
Slides to these chapters provided on course web-page

I. Theoretical Problems

1 Structured Light

1.1 2D Triangulation

Section 9.1.1. (see handout course web-page) described the 2D triangulation using two known angles α and β (see Fig. 1 below). Now, consider the calculation of the angle β. Let f be the effective focal length of the camera. The point P is projected onto a point p in the image plane Z=-f of the camera which has the X-coordinate x. The optical center is O (compare Figure 1). Derive the formula to determine the angle β from the position of the projected scene point p in the image plane.

1.2 3D Triangulation

As described in Section 9.1.1 in the Klette pdf (see handout course web-page) and also in our slides, the angle γ is not taken into account when the range data is calculated (3D case). How can γ be used to verify the reconstruction result? (see Fig. 9.2 Klette p. 351 , Fig. 2 below).

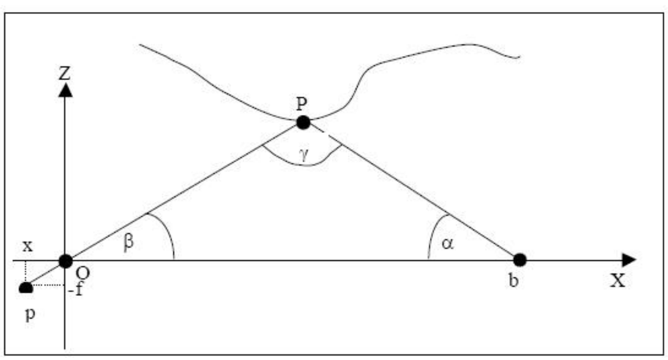

Figure 1: Illustration of the light spot projection technique in three dimensions.

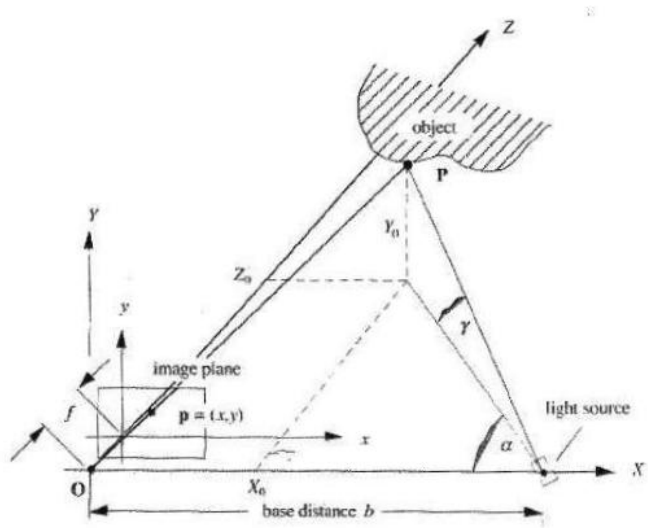

Figure 2: Fig. 9.7 Klette Chapter 9.

2 Motion and Optical Flow

- Sketch local displacement fields as seen in the image plane (a) for the translation of a rectangle lying and moving within a translation plane which is slanted to the image plane, and (b) for a rotation of a rectangle about one of its corner points where the rotation plane is parallel to the image plane. Assume the rectangle having a regular grid of points to show a dense field of motion vectors.
- Sketch the displacement fields for both situations (a) and (b) as measured by local displacement image processing operators that would only see the border of the rectangle. (Remember that velocity vectors can only be measured orthogonal to the boundary if estimated locally).
- Compare the two. Would you have the vector field of local velocities available, would you be able the true motion of the rectangle?
- Sketch (, draw, describe) that the aperture problem can be solved if a corner is visible through the aperture (means the small window where you calculate flow).

II. Practical Problem

3 Optical Flow

Optical flow is a technique to determine motion in a series of image data. We are given a sequence of images showing a real life situation of a driver (soure R. Klette, Auckland). Our goal is to implement an optical flow method that can measure the motion of objects.
Implement an optical flow method to determine the motion between consecutive images. The methodology is based on the basic assumption of brightness constancy $\nabla E.v + \frac{\partial E}{\partial t} = 0$. Remember further that we cannot calculate a solution for the velocity vector at every pixel but need to include a pixel neighborhood. Note that this is a local estimate, where we can primarily

toy2.png toy3.png toy4.png
toy5.png toy6.png toy7.png

Figure 3: Video sequence (courtesy of Anastasios Roussos, Queen Mary University of London, UK).

only measure the normal flow (i.e. flow parallel to the image gradient and thus perpendicular to boundaries.).

1. Choose two consecutive images from the video sequence.
2. Apply smoothing to the images (remember that optical flow assumes smooth object boundaries, i.e. boundaries with larger smoothness than the spatial shift). See note below regarding image smoothing.
3. Calculate the temporal gradient image $\frac{\partial E}{\partial t}$ via the difference of the blurred versions of the two consecutive frames.
4. Calculate the spatial derivatives $E_x = \frac{\partial E}{\partial x}$ and $E_y = \frac{\partial E}{\partial y}$. See note below regarding image differentiation.
5. Calculate the time gradient by the difference between consecutive frames, simply subtracting the two frames as I(x,t+1)-I(x,t).
6. Display the original image and the spatial and time gradients (see Fig. 4 for an example.
7. Display the resulting flow vectors as a 2D image. Maybe you find a way to overlay these vectors and the gray level image.
8. Eventuall apply your program to another consecutive image pair of the 6 image video sequence and compare the results.
9. Look at the velocities of the toy cars and discuss your solution.
10. **Improved estimate of time gradient:** Choose 2x2 pixel neighborhoods for a local estimate of velocities using the following solution strategy:

$$\nabla E.v + \frac{\partial E}{\partial t} = 0$$

now you have 4 measurements from 2x2 pixels to solve for v:

$$\begin{aligned} Av + b &= 0 \\ A^T A v &= -A^T b \\ v &= -(A^T A)^{-1} A^T b \\ C &= A^T A \end{aligned}$$

The columns of A are the x and y components of the gradient ∇E and b is a column vector of the t gradient component of E, E_t.

This calculation must be performed at each (x,y) in the image with the columns of A and b extracted within a neighborhood of size 2x2 (or NxN if you want to extend the smoothness range).

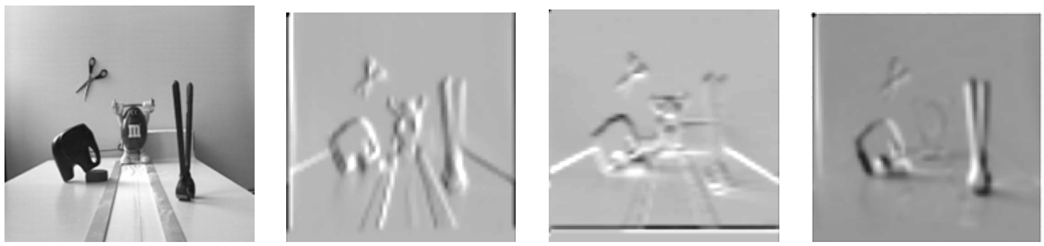

Figure 4: Example of derivatives used to solve optical flow. From left to right: Original image, derivative in x, derivative in y, derivative over time frames.

Image smoothing: This is for non-imaging students: Digital images can be smoothed via spatial filtering, ideally by Gaussian filters. In principle, this is a process that replaces each pixel by the average of the pixel neighborhood, e.g. a 3x3 or 5x5 window, or by a weighted average of these neighborhood pixels. In 2D, one can implement a 2D window filter, sequentially go through the image, and replace each pixel by the averaging. As an alternative, we can use the property of separability and run a horizontal filter followed by a vertical filter. E.g., you can use a symmetric 1-D filter [1,2,1]/4 or [1,3,4,3,1]/12 (the division accounts for the fact that the resulting value has to be divided by the sum of the weights since you don't want to change the average image brightness). Notice that you cannot filter the border, so that there is a 1 pixel border for a filter with 3 elements and a 2 pixel border for the 5-element filter, best is to set these unfiltered pixels to 0.

We will provide a Matlab image filtering module.

Image derivatives: This is for non-imaging students: *After smoothing*, the simplest way to get the components of the image gradient is a [-1,0,1]/2 filter that is applied horizontally and vertically, respectively (since the spatial length is 2 pixels, the result of this difference needs to be divided by 2). I.e., you replace each pixel by the difference between the right and the left neighbor divided by 2, which results in a derivative image that has positive and negative values. Notice that you cannot filter the whole image but create a nonfiltered border of 1 pixel width. The horizontal derivative gives you E_x, and the vertical derivative E_y. Please also note that the derivative operation creates a signed image, i.e. results that contain negative and positive values. You therefore need to choose the type "double" for all the differentiation and optical flow calculations. For display of images, we usually convert double back to integer and move the into a positive value range.

4 Bonus: Regularization: Horn and Schunck or Lucas and Kanade Method

This is a bonus question for students who would be willing to do extra work and want to earn extra credits.
Given the limitations of the local estimates above, you might implement the Horn and Schunck or Lucas and Kanade algorithms which provide a smoothing term to regularize the flow field (see details in slides Optimal-Flow-I and in the respective Horn-Schunck papers on canvas).

- Implement the Horn-Schunck algorithm.
- Compare the flow fields with the ones obtained with the simplified approach.