

CS 6968, 3D Computer Vision

Fall 2009, Prof. Guido Gerig

Assignment 3: Structured Light / Motion

Out: Sat Nov-7-2009
Due: Tuesday Nov-24-2009
TA: Evrard Ohou
Office hours: Tue/Thu 2:00pm to 3:00pm

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

I. Theoretical Problems

1 Structured Light

1.1 2D Triangulation

Section 9.1.1. described the 2D triangulation using two known angles α and β . 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). 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 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 below).

1.3 Light Stripes in Image Plane

A light plane is projected into a scene consisting of two polyhedral objects using a projector. The light stripes represented in Fig. 2 are visible in the image. The geometry of the image acquisition system corresponds to the arrangement in Fig. 9.7 (Fig. 3 below) (page 356 Klette Chapter 9 handout) with $M = 512$, $D_0 = 30cm$, $D_z = 60cm$, $b = 20cm$, and $f = 15mm$. Determine the corresponding 3D profile (see equations 9.1 to 9.10), i.e. the 3D coordinates determined in the camera system (optical center). You can then visualize the profile in 3D.

Note that the image size is 512x512 pixels. The figure shows the pictures of the polygon points, and you can read their pixel coordinates.

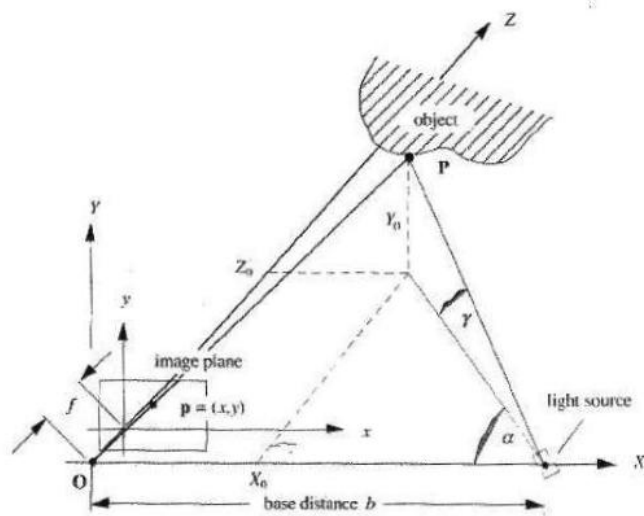


Figure 1: Illustration of the light spot projection technique in three dimensions.

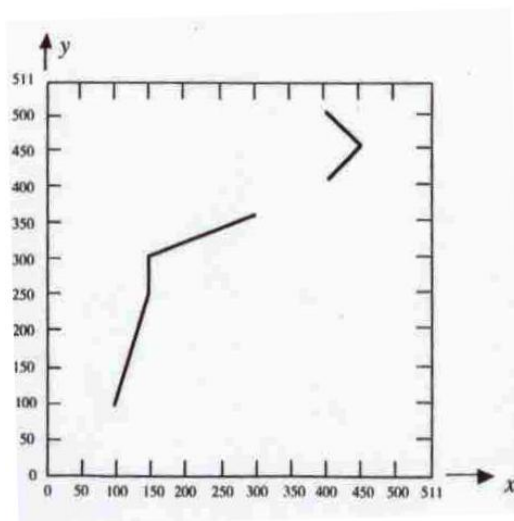


Figure 2: Light Stripes visible in Image Plane.

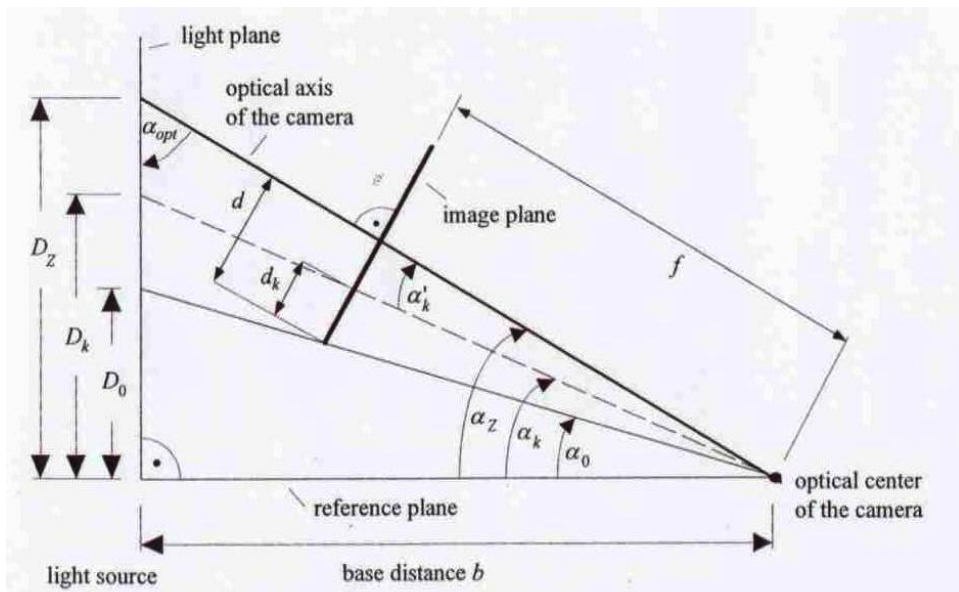


Figure 9.7: Schematic representation of a particular geometry for the light stripe technique which simplifies the calibration significantly.

Figure 3: 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.

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 (source R. Klette, Auckland). Our goal is to implement an optical flow method that can measure the motion of objects.

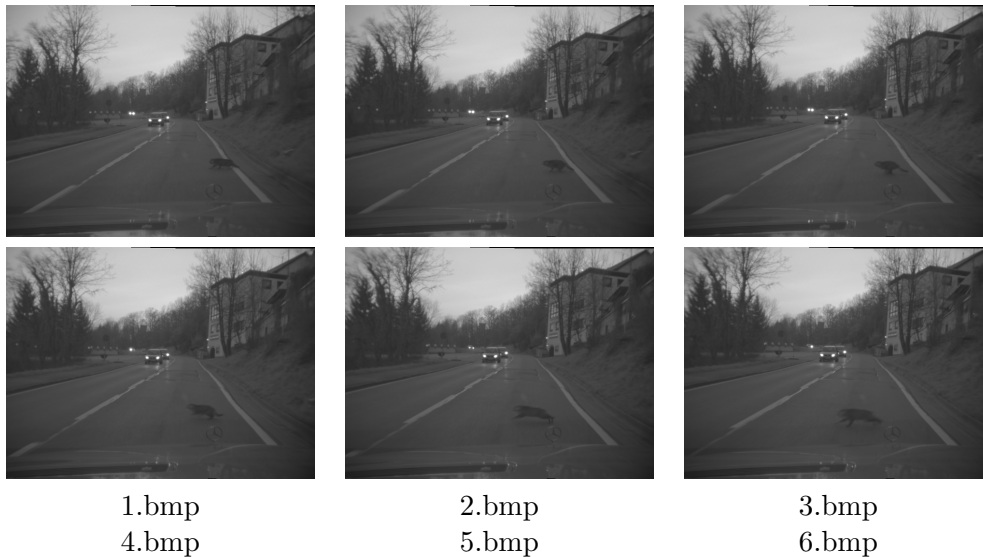


Figure 4: Video sequence.

Implement an optical flow method to determine the motion between consecutive images. The methodology is based on the basic assumption of brightness constancy $\nabla E \cdot v + \frac{\partial E}{\partial t} = 0$. 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

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. Choose 2x2 pixel neighborhoods for a local estimate of velocities using the following solution strategy:

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

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

$$Av + b = 0$$

$$A^T Av = -A^T b$$

$$v = -(A^T A)^{-1} A^T b$$

$$C = A^T A$$

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).

6. Display the resulting flow vectors as a 2D image. Maybe you find a way to overlay these vectors and the gray level image.
7. Apply your program to more consecutive image pairs of the 6 image video sequence and compare the results.
8. Look at the cat, incoming car and other features that change and discuss your solution.

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.

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 .

4 Bonus: Horn and Schunck 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 algorithm that provides a smoothing term to regularize the flow field (see details in slides Optimal-Flow-I and in the Horn-Schunck paper on WebCt).

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