Topic 10: Stereo Imaging

Aim

The lecture covers the basic background of stereo imaging and how it can be used to extract depth information from images.

Contents:

- Three Dimensional Imaging.
- Parallel Geometry Stereo
- Aerial and Satellite systems
- Converging Geometry
- Automatic Extraction of Depth
- Region Segmentation by Stereo

Three Dimensional Imaging

Image detected by two-dimensional camera contains no **depth** information. However in many system we need depth information, for example

- Map making.
- Robotic vision
- Target tracking

Active Measurement: Range of pulse-echo techniques to measure distance to a point. (Radar, Ultrasound, Laser Pulse). Not really imaging.

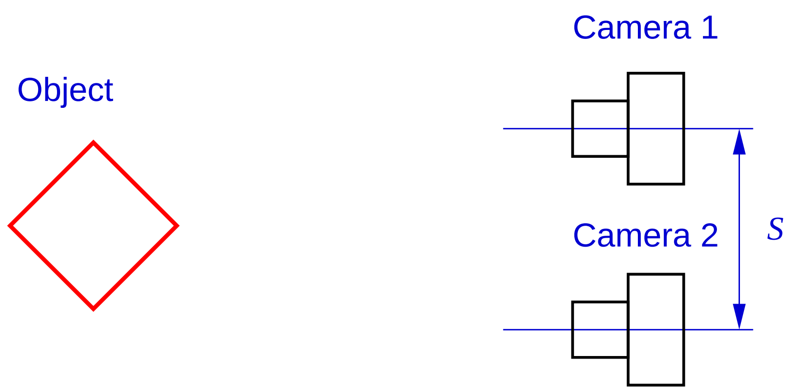
Stereo Imaging: Two (or more) cameras (images), with the depth information extracted from the differences.

Holography: Active optical system that records full three-dimensional object information. Very difficult to extract the information.

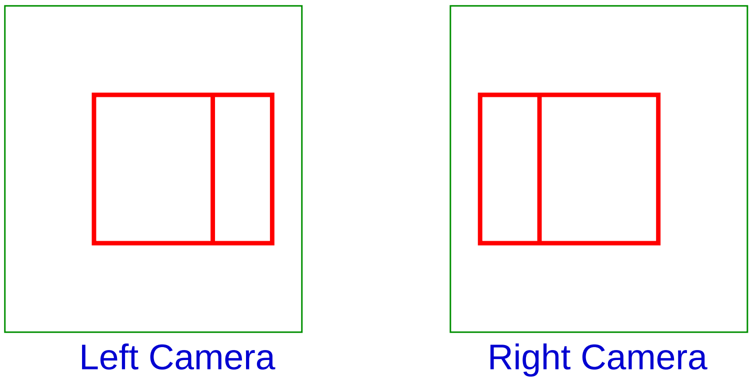
Of these three *Stereo Imaging* uses conventional cameras and image processing to extract the information.

Basic Scheme

Use **two** cameras to look at three-dimensional object,



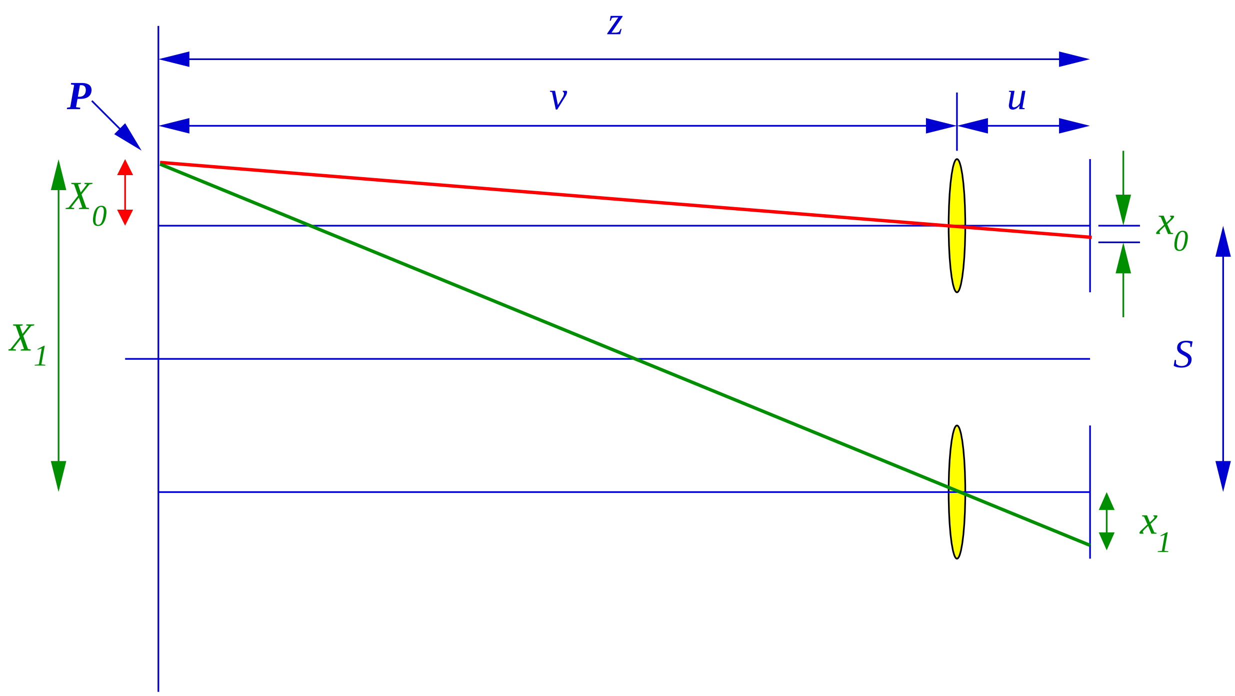
From the two camera we get a different view, typically, for a cube:



So we get shifts in the **vertical lines** only, from which we can extract depth information.

Parallel Geometry

Assume the optical axis of the two camera is parallel and separated by distance S.



Consider image of point P, for the two camera

Parallel Geometry

we have

$$\frac{x_0}{u} = \frac{X_0}{v} \quad \& \quad \frac{x_1}{u} = \frac{X_1}{v}$$

so that

$$x_0 = \frac{uX_0}{v} \quad \& \quad x_1 = \frac{uX_1}{v}$$

but from the geometry we have that

$$X_1 = X_0 + S$$

so that by substitution we have that

$$x_1 = x_0 + \frac{uS}{v}$$

so we get that

$$v = \frac{uS}{(x_1 - x_0)} = \frac{uS}{\Delta x}$$

where Δx is the difference between the image of P in the two images.

Parallel Geometry

For a lens of focal length f we have that

$$\frac{1}{u}+\frac{1}{v}=\frac{1}{f}$$

however in most practical systems $v \gg u$, so we can take the approximation that $u \approx f$, giving that

$$v \approx \frac{fS}{\Delta x}$$

or noting that z the distance from the image plane to object plane is $z = u + v$, then we get that

$$z = f\left(1+\frac{S}{\Delta x}\right)$$

so given the focal length of the lenses and the separation, then the depth z can be measured from Δx which is the displacement imaged point.

Depth Accuracy Problem

Take an error of δx in the measurement of Δx, so that

$$\Delta x = \Delta x_0 \pm \delta x$$

so if we define,

$$v_0 = \frac{uS}{\Delta x_0}$$

then we can write $v = v_0 \pm \delta v$, where we find (by Taylor expansion), that

$$\delta v = \frac{uS}{\Delta x_0^2} \delta x$$

so substituting for Δx_0, we get that

$$\delta v = \frac{v_0^2}{uS} \delta x$$

which shows that for a fixed S and δx then the error in the depth measurement rises as the **square** of distance from the camera.

To get good depth resolution you need a large S (camera separation).

Example System

For a CCD camera the error δx is given by the size of the CCD sensor, typically about 20μm is a reasonable camera.

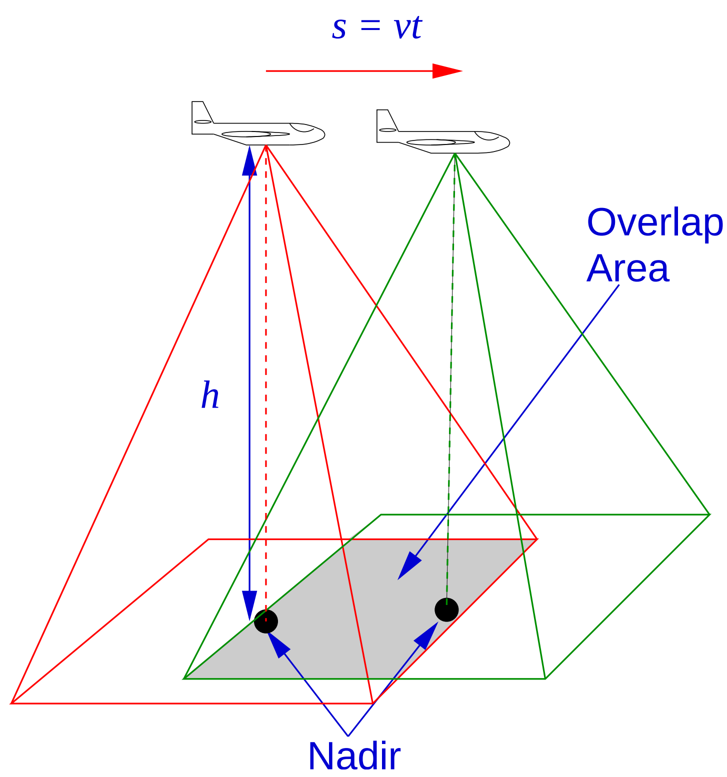
Sensor Size (δx)	20μm
Focal length f	25mm (typical)
Separation	100mm

$$v_0 \approx 1\text{m} \quad \Rightarrow \quad \delta v \approx 8\text{mm}$$
$$v_0 \approx 10\text{m} \quad \Rightarrow \quad \delta v \approx 800\text{mm}$$

Very similar to the human visual system, stereo vision useful up to about 10m, beyond that we use size and perspective to estimate distance.

Aerial Photography

Camera set to point "straight down". Take two images separated by time t, so distance $s = vt$.



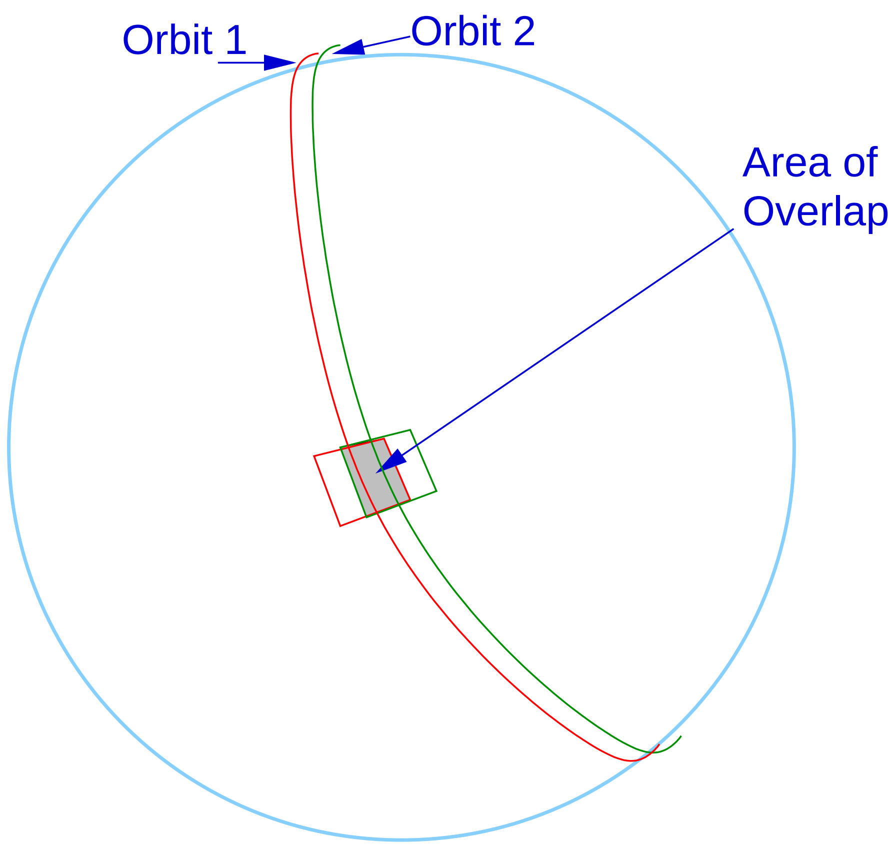
Know cemara separation we can calculate height variation in overlap area.

Use in automatic map making.

Problem: Height and orientation of aircraft can change between exposures making this *much* more difficult than expected.

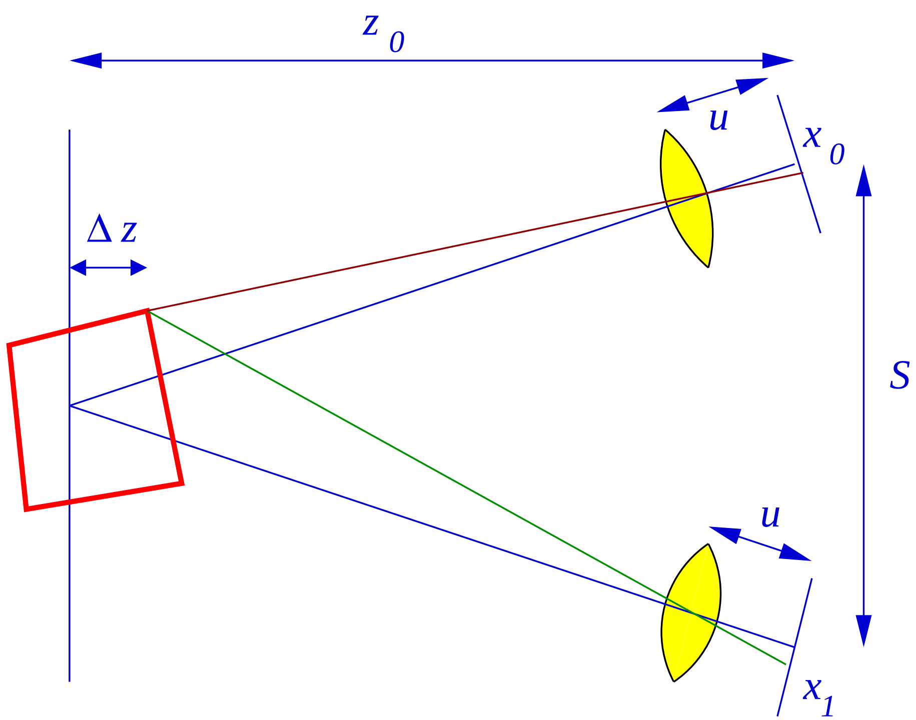
Satellite Stereo

Satellite Stereo: Two images detected from different orbits (at different times).



Converging Systems

In practical system the cameras as **not** parallel, but convergent to an average point in a plane at distance z_0.



We *can* solve for Δz, from the positions x_0 and x_1 in the two images. (Challenge).

Expressions are much more difficult, but have the same form, and problems as the parallel case.

Extraction of Depth Information

Manually: Use human operator and optical viewing system. Used in most commercial map-making systems to trace contours.

Box World: Simple geometric objects, such as found in most computer vision systems (automated inspection).

1. Enhance and detect *vertical* edges.
2. Search for corresponding edges in the two images.
3. Use difference to give depth-of-edge.

Works well for low noise images where edges are easy to find, problems are

- Missing edges in one image due to perspective change. (difficult to deal with).
- Missing edges due to noise (process images to form continuous edges).
- Confusion between edges. Problem if there are many edges, often solved by applying continuity rules between adjacent lines.
- Noise points. Try and remove before extracting depth information, but can severely upset analysis.

Some modern systems use **three** cameras in a triangle to resolve missing edge problems.

Converging Systems

General Images: No simple, or general solutions. Range of possibilities based:

1. Identify and locate “known-objects” in each image (eg. cross-roads on a map). Works well for slow varying height information as found in aerial photography. (exact analogue of the human matching technique).
2. Region analysis of each image, and then match-up boarders of regions. Works well if you can break the image into coherent regions. (eg. in “box-world”).
3. Relaxation Labeling of edges or pixels. Hypothesis testing technique based on trying to form regions of common depth or slope within an image. AI scheme.

All of these schemes work to a greater or lesser extend depending on the type of image and how much knowledge you have about it.

Extraction of Information in a Plane

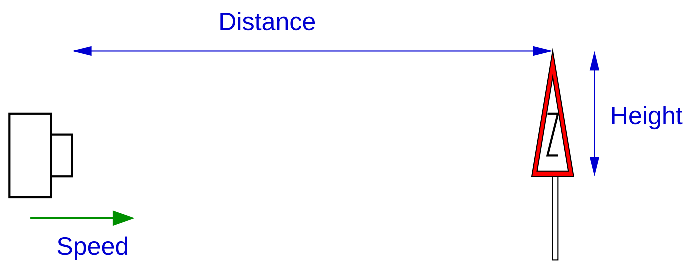
Practical problem form computer vision Esprit project we were involved with.

Problem: Make a road-sign recognition system for in-vehicle use.

To aid the recognition process we want to image of the approaching road-sign of known size, with as little background "clutter" as possible. (We really want an edge detected road sign extracted from the image).

Note: all road signs come in three standard sizes (EEC regulation), (assume single size at the moment). Want to use this to extract the sign.

Depth Information



As car approached road we want to detect edges from one particular plane, so if the sign is in this plane, we will then detect the sign of the right size and ignore all other edge points in the image.

Scheme: Mount **two** cameras in the vehicle separated by a distance S.

1. Detect images from both cameras, and perform real time (simple) edge detection.
2. Threshold each to form binary edge images.
3. Reject all edge points **except** those that occur in both images with displacement Δx.
4. Use selected edge points to extract region(s) from input image.

All operations simple, (vertical edge detection, threshold, shift and logical `or`). Actually all can be done in analogue hardware.

Initial results form laboratory demonstrations looked hopeful, but it was never used on real system.