Lecture 9 - Structured light

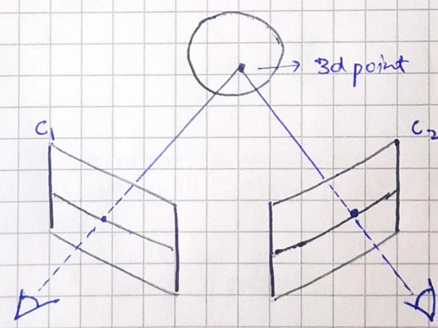
→ Structured light is a method for dense ~~reconstruction~~ correspondence generation.
→ It uses active devices (lasers/projectors)

→ motivation for structured light.
① If we take any object, and we reconstruct the scene, we will get good result where we have structures.
② In the regions where we do not have heterogeneous areas, we cannot create any correspondences
③ The idea of structured light is to overcome this problem. i.e.

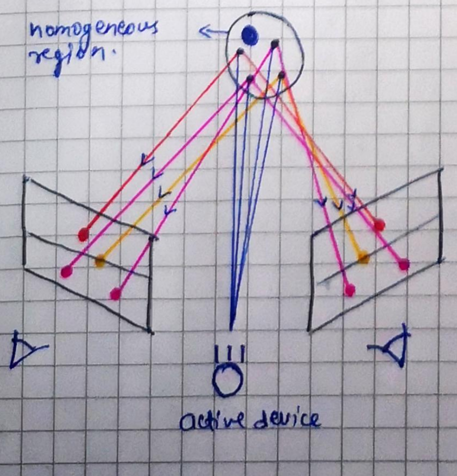
① Robustness / Applicability
(1) Reconstruct the objects independent of its own features.
(2) If there are no features on the object, add some by active devices like:
(a) Point / line lasers
(b) Video projector

→ concept (Active Reconstruction)

① Traditionally, point correspondences with 2 stereo cameras can be used to triangulate a 3d point in real world.

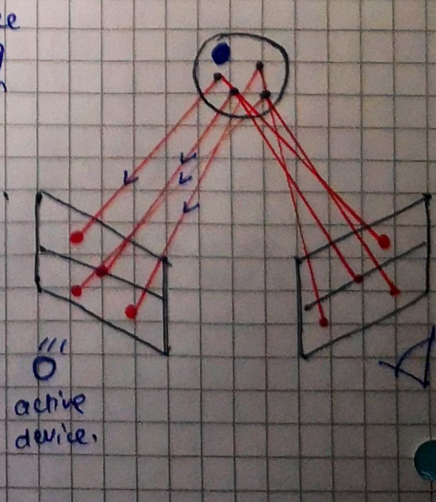


② But if we have a 3d point in which point correspondences cannot be drawn, because the feature is not detected, we add an active device with the two cameras called "Active stereo" which adds features to the object.



we can also replace this setup by using a video projector in place of a camera.

If we know the calibration b/w 2 cameras, we do the triangulation and get the point correspondences.



→ Calibration of a structured light setup.

To perform the calibration, we need (a) intrinsics of camera, Projector (b) position of both devices so that they are in the same coordinate system. (Relative extrinsics)

① Intrinsic Camera calibration → The calibration of Intrinsic camera parameter is usually done using calibration target - i.e.

(a) Capturing the sequence with a known calibration sequence (eg. checkerboard)
(b) finding the checkerboard in the images.
(c) Using the found correspondences in order to compute the camera's intrinsic calibration matrix K_C

② Camera Projector calibration → Recalling the stereo camera case, we know that

(a) The same checkerboard is seen by both the cameras.
(b) Extracting the corners of the checkerboard in the images gives the corresponding image points b/w the camera view.
(c) The 3d points of the checkerboard stay the same. what changes are their projections in the images.
(d) from this information, the intrinsic camera Parameters, that cause this projections can be computed.

note: But how do we do this to a camera - Projector setup?

(a) model the projector by the same pinhole model as for cameras.

note: But how to generate correspondences b/w calibration board and the projector?

(a) The projector cannot see anything, thus no correspondences can be generated b/w the calibration board and the projector.

(b) we need 2 checkerboards, one to be seen by camera, and one to be projected by the projector.

★ (i) The image plane of the projector is the image to be projected and is static, independent of the board position.

(ii) 3d positions where the chessboard corners are projected change with the board position.

(iii) Printed chessboards can be seen by the camera and thereby the camera pose wrt the board can be calculated (since K_C is known)

Calibration	Camera	Projector
① 3d checkerboard corners	fixed	Variable
② 2d checkerboard corners	Variable	fixed

★ How to get the variable values?

* Camera's 2D point: Use checkerboard detection algorithm.
* Projector's 3D point: Use calibrated camera to compute the position.

* With the 3D ↔ 2D correspondences of the camera and the projector compute independently
  (i) Intrinsics calibration matrices: K_C and K_P
  (ii) Extrinsics wrt n different checkerboards:
  $R_{C,i}, t_{C,i}$ and $R_{P,i}, t_{P,i}$ $(i = 1 \cdots n)$

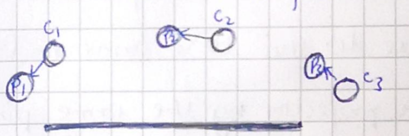
* However, we need one relative orientation b/w camera and projector:
  (1) a trick is to map projector coordinates to camera coordinates for some i:

$$[R, t] = [R_i, t_i] = [R_{C,i} R^T_{P,i},\ t_{C,i} - R_{C,i} R^T_{P,i} t_{P,i}]$$

  (2) Higher accuracy can be achieved by formulating an optimization problem that takes into account all captured checkerboxes $(i = 1 \cdots n)$ and simultaneously optimizing K_C, K_P and $[R, t]$

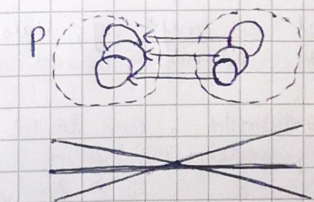
→ To perform the calibration setup of camera - projector we have 2 choices

① Initialisation with a ~~static~~ simple method
In this method, the calibration board is static. i.e. for calibration, we assume that we move the camera - projector around the board.



for each position, we get a rotation and a translation.

② An Equivalent setup
→ In reality, we move the calibration board while keeping the camera-projector setup static.

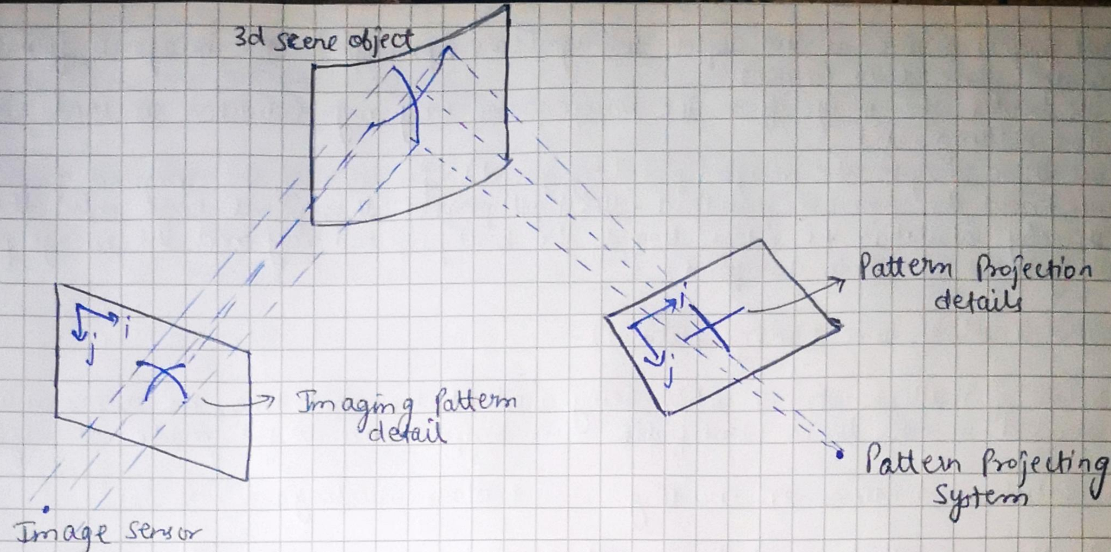


→ we search for 1 optimal pose which mostly approximates the input correspondences
→ This is done by minimising the reprojection error using Levenberg - Marquardt (or any other gradient descent method)

→ Correspondence generation → After setting up the camera - projector calibration, the most important thing is to generate correspondences.

Correspondences → Triangulation → 3D information.

→ Here, we only focus on correspondence generation b/w cameras and light emitting devices.
* → There exists several classes of laser scanner. we will only consider those based on triangulation.

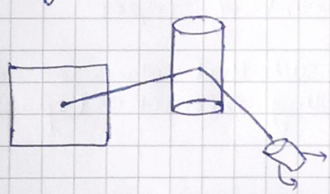


→ It is a general setup in which we project a pattern (through pattern projection system) and receive correspondences at the image sensor.

★ The question here is, how do we know which pattern to use?

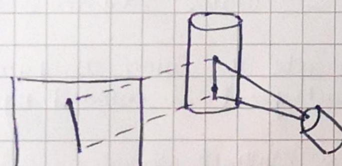
→ Types of projection pattern

① single dot



→ No correspondence problem
→ Scanning both axis

② single stripe



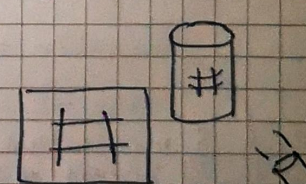
→ correspondence problem among points of same slit.
→ scanning the axis orthogonal to the stripe

③ multiple stripes

① we move our projection device to get stripe patterns.
② Correspondence problem among slits (we cannot recognise which point to slit)
③ NO scanning

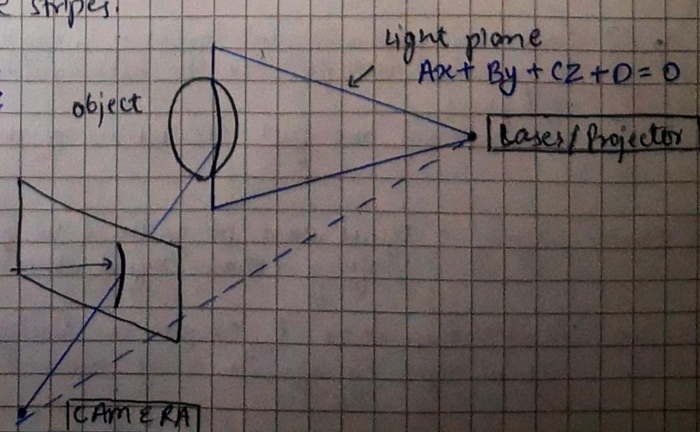
④ Grid, multiple dots:

① Correspondence problem among all the image segments
② NO scanning.



→ Correspondence generation → single stripes.

→ Because the lines span the planes in 3D, it is possible to intersect the camera rays with the plane spanned by emitted light rays.



① we project a line at the object through (laser / projector). The image is formed on the image plane of the camera.
② we extract the image from the image plane (by any extraction methods such as thresholding, etc)
③ we then sample the image (pixel by pixel). for each of the sampled point we know the camera position. we back project this point back into the scene thereby calculating the intersection of the backprojected ray and the image plane.
④ we get the 3d coordinate point.

★ Takes a lot of time (many pictures)

① In case of multiple stripes when being used at the same time, the stripes must be encoded in order to be identifiable i.e. stripes have to be made distinguishable.

solution → 1D or 2D encoding. (colour encoding)

② A pattern is called encoded if a set of regions of the observed projection can be easily matched with the original pattern after projecting it onto a surface.
Eg. Encoding by colour.

③ Decoding a projected pattern allows a large set of correspondences to be easily found, thanks to the prior knowledge of the pattern

Problem: ① colours are altered during light transport
② colours interfere with surface colours
→ lower robustness for larger amount of stripes

Solution: ① Encode the colour assignment itself, such that colors can repeat ~~themselves~~
② Each point is encoded by its surrounding intensities (spatial codification)

④ Binary coding of multiple stripes.

① A more robust way is to do binary encoding.
② A temporal method in which "n" patterns / images must be encoded in order to get "2^n" stripes.

③ Assign each stripe a unique illumination code over time.

Limitation → ① One of the limitations is that the resolution we get depends upon the resolution of last pattern / image.

④ more robust than colour encoding but requires a lot of images.
⑤ Instead of binary codes, grey codes are often used in practise.
⑥ Adjacent code words differ only in 1 bit.
⑦ This allows to correct some errors.

② Large resolutions require a lot of images to be projected ($1024 \times 768 \xrightarrow{2^{10}}$ 10 images)
③ In practise not possible to distinguish projected stripes with only a width of 1 pixel.
④ consequently, the full resolution of the projector cannot be exploited.

⑤ Phase-shifting → If we want to go close to pixel level we can use Phase-shifting. (Phase-shifted structured light)

→ Consider a function $f_{ref}(x,y) = x$, coding the projector reference image from left to right by different gray levels.

→ Projecting this function could be used for direct codification of the scene.
→ Problem: The camera cannot precisely distinguish b/w the gray scales.
→ Problem → The scene colorization interferes with the projected gray levels.

Solution → ① Phase-shifted structured light can be used to encode the function $f_{ref}(x,y)$ in a more efficient way.

② multiple sine/cosine waves with shifted phase are projected subsequently.

$$g_{ref}(x,y) = \cos\Big(f_{ref}(x,y) + \underbrace{\frac{2\pi n}{N}}_{\text{Phase shift}}\Big) = \cos\left(x + \frac{2\pi n}{N}\right) \qquad x \in [0, 2\pi]$$

③ This allows to higher constrast over the full image.

→ 3-step Phase-shift algorithm.

① Encoding is done by projecting cosine waves being shifted equidistantly atleast 3 times

$$g_i(x,y) = \cos\left(f_{ref}(x,y) + \frac{2\pi i}{3}\right) \qquad i = 0, 1, 2 \ldots$$

② The resulting images are of the form

$$I_i(x,y) = A(x,y) + B(x,y)\cos\left(f_{obj} + \frac{2\pi i}{3}\right)$$

Background image (Ambient light intensity)

cosine amplitude

phase of the object

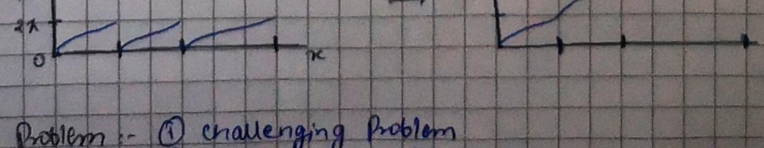
③ Similar to the above equation, we have 2 more equations (I_1, I_2, I_3). From these images, the object phase can be directly computed:

$$f_{obj} = \arctan\Big(\underbrace{\sqrt{3}\,\frac{I_1 - I_3}{2I_2 - I_1 - I_3}}_{\text{A and B term vanishes}}\Big)$$

④ Higher frequencies of sin/cosine waves can be used to refine the phase (Phase unwrapping)

★ Problem → ① Due to arctangent, the phase is wrapped to the interval $[0, 2\pi]$ by applying a modulo 2π operation.

② The discontinuities of the arctangents can be removed by adding or substracting multiples of 2π to $f_{obj}(x,y)$ "unwrapping".



Unwrapping Problem :- ① challenging problem
② Unrobust, if using only a single wrapped phase - especially at discontinuities.

Solution → A robust solution is to use level based unwrapping algorithm.

→ Capture multiple fringe levels

Level 0 → shift 1 stripe
→ no wrapping

Level 1 → shift eg' 5 stripes
→ wrapping.

→ Phase shift algorithm.

1. Generate patterns
2. Project patterns to the scene
3. Compute phase from captured images
4. Refine the phase with patterns of higher frequency and unwrapping.
5. find matches b/w images using horizontal and vertical phases.
6. Triangulate the dense point cloud.

→ Epipolar geometry can also be used.