Module 1: Basics of 3D Computer Vision

Lesson 1 Part 1: The Camera Sensor

単語
- aperture: In photography, the aperture of a camera is the size of the hole through which light passes to reach the film. 口径

内容
- What makes a camera useful for self-driving cars.
- The characteristics of a camera as a sensor, and how images are formed.

The Camera Sensor
- Appearance information is particularly useful for scene understanding tasks such as object detection, segmentation and identification.
- Appearance information is what allows us to distinguish between road signs or traffic lights states, to track turn signals and resolve overlapping vehicles into separate instances.
- Captures detailed appearance information.
  - Because of its high resolution output, the camera is able to collect and provide orders of magnitude more information than other sensors used in self-driving while still being relatively inexpensive.
- High rate of information capture.

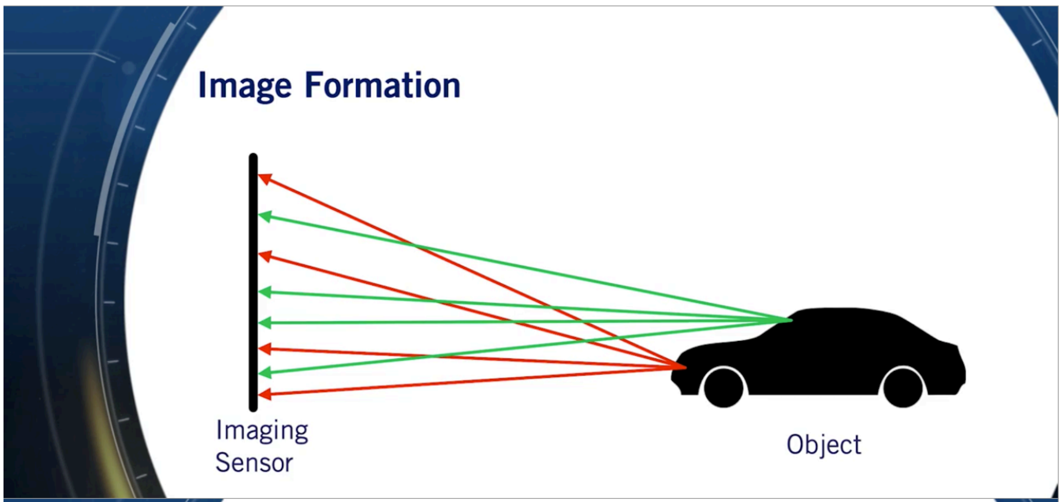


Image Formation

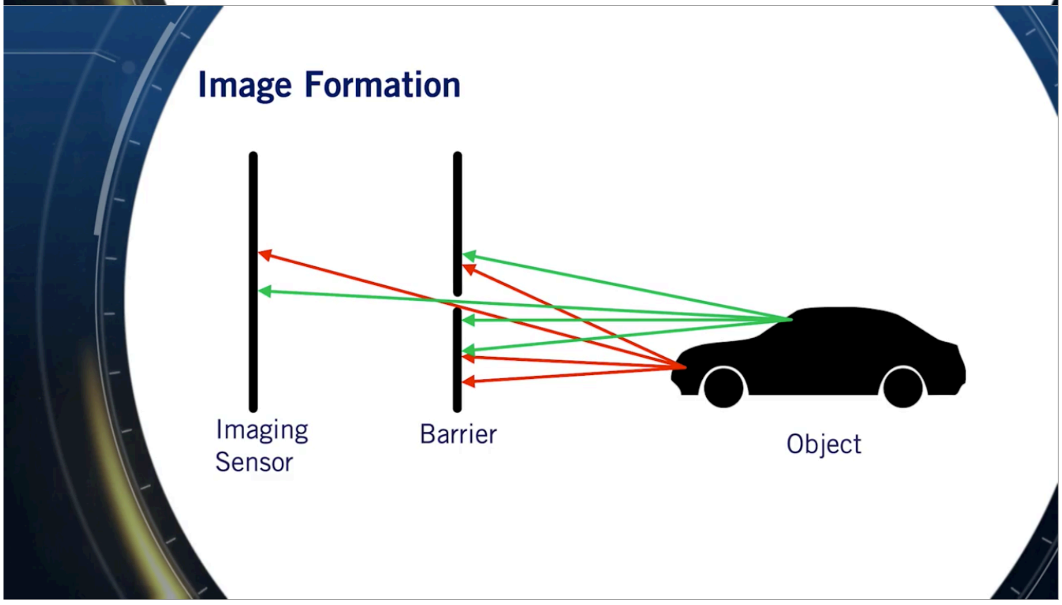


Image Formation

Image Formation
- Image capture was originally done with film but nowadays with rather sophisticated silicon chips.
- 上の図：basic open sensor camera design.
  - we'll end up with blurry images because the imaging sensor is collecting light rays from multiple points on the object at the same location on the sensor. つまりobjectの違う点がsensorの同じ点に投影される。
- 下の図：solution (pinhole camera model), put a barrier in front of the imaging sensor with a tiny hole or aperture in its center.
  - The barrier allows only a small number of light rays to pass through the aperture, reducing the blurriness of the image.

Pinhole Camera Model
- Describe the relationship between a point in the world and its corresponding projection on the image plane.
- 重要なパラメータ
  - Focal length f, the distance between the pinhole and the image plane.
    - The focal length defines the size of the object projected onto the image.
    - Plays an important role in the camera focus when using lenses to improve camera performance.
  - Camera Center (c_u, c_v), the coordinates of the center of the pinhole.
    - These coordinates defines the location on the imaging sensor that the object projection will inhabit.

Modern Day Cameras
- They can operate in low-light conditions or at a long range due to the advanced lens optics that gather a large amount of light and focus it accurately on the image plane.

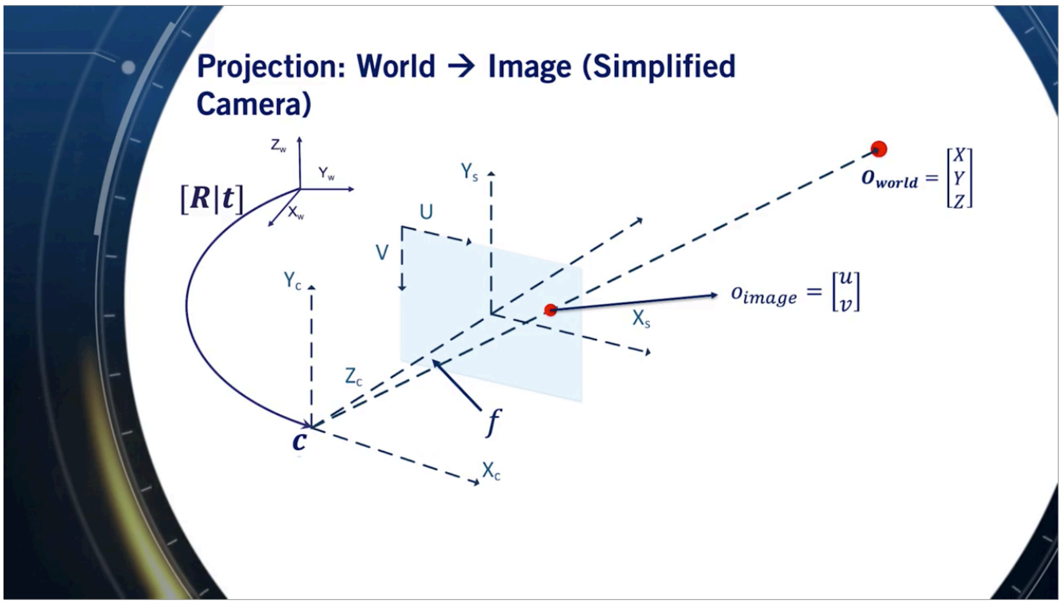
Lesson 1 Part 2: Camera Projective Geometry

単語
- aspect ratio: アスペクト比, the ratio of width to height of the picture on a television or cinema screen.

内容
- How to model the camera's projective geometry through coordinate system transformation.
- How to model these transformations using matrix algebra and apply them to a 3D point to get its 2D projection on the image plane.
- How a digital image is represented in software.

Projection: World → Image (Real Camera)
- The projection onto the sensor surface through the aperture results in flipped images of the objects in the world.
  - To avoid this confusion, we usually define a virtual image plane in front of the camera center.
  - 下記のsimplified camera model.

Projection: World → Image (Simplified Camera)
- World coordinate frame: X_w, Y_w, Z_w
- Camera coordinate frame: attached to the center of the lens aperture known as the optical sensor.
  - X_c, Y_c, Z_c
  - Camera coordinate frameの原点はlens aperture center.
  - Refer to the parameters of the camera pose as the extrinsic parameters, as they are external to the camera and specific to the location of the camera in the world coordinate frame.


Projection: World → Image (Simplified Camera)
$[\boldsymbol{R}|\boldsymbol{t}]$
Z_w
Y_w
X_w
Y_s
U
V
Y_c
Z_c
X_s
f
c
X_c
$O_{world} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$
$o_{image} = \begin{bmatrix} u \\ v \end{bmatrix}$

- Image coordinate frame (X_s, Y_s): attached to the virtual image plane emanating from the optical center.
  - Image coordinate frameの原点はoptical center.
- Image pixel coordinate system (U, V): attached to the top left corner of the virtual image plane.

Computing the Projection

- Projection from World coordinates to Image coordinates:
  1. Project from world coordinates to camera coordinates.
  2. Project from camera coordinates to image coordinates.
  3. transform image coordinates to pixel coordinates through scaling and offset.
- World -> Camera: $o_{camera} = [R\,|\,t]O_{world}$.
- Camera -> Image: $o_{image} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} o_{camera} = Ko_{camera}$.
  - K depends on camera intrinsic parameters.
- 上記の２ステップを合わせるWorld -> Image: $P = K[R\,|\,t]$.
  - Projection from world coordinates to image coordinates: $o_{image} = PO_{world} = K[R\,|\,t]O_{world}$.
  - When we expect the matrix dimensions, we noticed that the matrix multiplication cannot be performed.
    - $K: 3 \times 3$, $[R\,|\,t]: 3 \times 4$, $P: 3 \times 4$, $O_{world}: 3 \times 1$.
    - To remedy this problem, we transform the coordinates of O into homogeneous coordinates: $O_{world} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$.
      - $K: 3 \times 3$, $[R\,|\,t]: 3 \times 4$, $P: 3 \times 4$, $O_{world}: 4 \times 1$.
- World coordinates to Image coordinates: $o_{image} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = K[R\,|\,t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$.

- Image coordinates to Pixel coordinates: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$.
- 実際のもっと複雑な課題：non-square pixels, camera axis skew, distortion, non unit aspect ratio.
  - Luckily, this only changes the camera K matrix.

The Digital Image: Grayscale
- Each point in 3D projects to a pixel (a grid) on the image defined by the uv coordinates.
- In grayscale, brightness information is written in each pixel as an unsigned 8 bit integer. (0 ~ 255).
- Some cameras can produce unsigned 16-bit integers for better quality images.

The Digital Image: Color
$M \times N \times 3$ arrays.

- These equations rely on the camera intrinsic parameters, as well as its extrinsic location in the world coordinate system.
- A color camera image is represented digitally as an MxNx3 array of unsigned 8 bit or 16 bit integers.
- How to tailor the camera model to a specific camera by computing its intrinsic and extrinsic camera parameters through a process known as camera calibration.

Lesson 2: Camera Calibration

内容
- How to find the camera matrix P through a process called camera calibration.
- How to extract the intrinsic and extrinsic parameters from the camera P matrix.

Computing the Projection
- Image coordinates to Pixel coordinates: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} su \\ sv \\ s \end{bmatrix}$.
  - u and v can be multiplied with an arbitrary scale s.
  - we multiply by s, as it will be useful when we formulate the calibration problem.
  - It is important to note that scale plays a challenging role in understanding monocular image information, as once points are projected from the 3D world onto the 2D image plane, scale information is lost.
  - Points in 3D space along a ray from the camera center, all project to the same location on the image plane, and it is therefore not possible to directly associate a depth to a point given only image information.

Camera Calibration: Problem Formulation
課題記述：Finding unknown intrinsic and extrinsic camera parameters, given n known 3D point coordinates and their corresponding projection to the image plane.
方法：getting P matrix first, and then decomposing it into the intrinsic parameters K and the extrinsic rotation parameters R and translation parameters t.
- Use scenes with known geometry to:
  - Correspond 2D image coordinates to 3D world coordinates.
  - Resolving the scale issue by measuring the actual 3D distance between the points that are observed in the image.
  - The most commonly used example would be a 3D checkerboard, with squares of known size providing a map of fixed point locations to observe.
    - Associating 3D points to 2D projections can be done either manually, or automatically, with checkerboard detectors.

- Find the Least Squares Solution (or non-linear solution) of the parameters of P:

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$

- Expanding the equation results in: $\begin{aligned} su &= p_{11}X + p_{12}Y + p_{13}Z + p_{14} \\ sv &= p_{21}X + p_{22}Y + p_{23}Z + p_{24}. \\ s &= p_{31}X + p_{32}Y + p_{33}Z + p_{34} \end{aligned}$
- Move to LHS (left-hand side): $\begin{aligned} su - p_{11}X - p_{12}Y - p_{13}Z - p_{14} &= 0 \\ sv - p_{21}X - p_{22}Y - p_{23}Z - p_{24} &= 0. \\ s - p_{31}X - p_{32}Y - p_{33}Z - p_{34} &= 0 \end{aligned}$
- Replace s in Eq. (1) and Eq. (2) to get 2 equations per point:
  $p_{31}Xu + p_{32}Yu + p_{33}Zu + p_{34}u - p_{11}X - p_{12}Y - p_{13}Z - p_{14} = 0$
  $p_{31}Xv + p_{32}Yv + p_{33}Zv + p_{34}v - p_{21}X - p_{22}Y - p_{23}Z - p_{24} = 0$.
- If we have N 3D points and their corresponding N 2D projections, set up homogeneous linear system:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -u_NX_N & -u_NY_N & -u_NZ_N & -u_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -u_NX_N & -u_NY_N & -u_NZ_N & -u_N \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = 0.$$

  - Since this is a homogeneous linear system, we can use the pseudo-inverse or even better, the singular value decomposition to get the least squares solution.
- Advantages of such a simple linear system are:
  - Easy to formulate.
  - Closed form solution.
  - Often provides really good initial points for non-linear calibration approaches.
- Disadvantages:
  - Does not directly provide camera parameters.
  - Does not model radial and tangential distortion and other complex phenomena.
  - Since we are solving via the linear least squares method, we cannot impose constraints on the solution, such as requiring the focal length to be non-negative.
    - 同じ課題は勉強したことがある。どこだろう?

Factoring the P matrix (RQ factorization)

$$\begin{aligned} P &= K[R\,|\,t] \\ &= K[R\,|-RC]\ . \\ &= [KR\,|-KRC] \end{aligned}$$

1. alter the representation of P to be a function of the camera center C.
   1. C is the point that projects to zero when multiplied by P.
   2. C: 3D camera center: $PC = 0$.

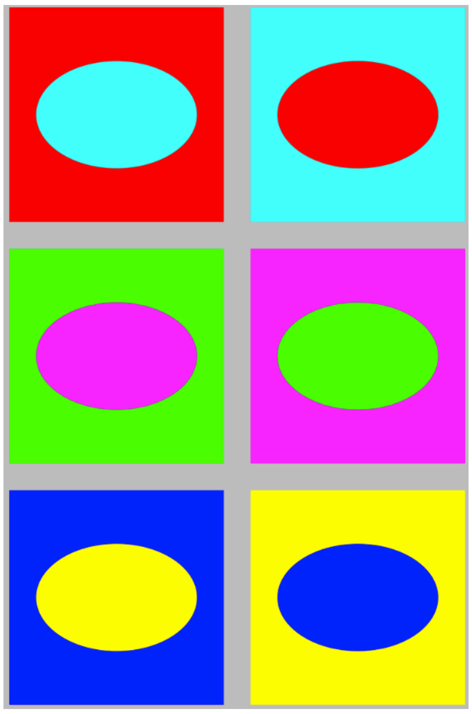
2. Let $M = KR$.
    1. $P = [M \mid -MC]$.
    2. $M = \mathscr{R}\mathscr{Q}$.
        1. use the fact that any square matrix can be factored into an upper triangular matrix $\mathscr{R}$ and an orthogonal basis to decompose M.
        2. だからRQ factorizationという名前だ。
        3. Which is a variant of the more commonly referred to QR factorization.
        4. $\mathscr{R}$ is not the Rotation matrix!
3. Intrinsic Calibration Matrix: $K = \mathscr{R}$.
4. Rotation matrix: $R = \mathscr{Q}$.
5. Translation vector: $t = -K^{-1}P[:,4] = -K^{-1}MC$.

Camera Calibration
Monocular camera calibration is a well-established tool that has excellent implementations in C++, Python and MATLAB.

- The camera matrix P can be found through a process known as camera calibration.
- The intrinsic and extrinsic camera parameters can be extracted from the P matrix using $\mathscr{R}\mathscr{Q}$ factorization.
- Camera Calibration with OpenCV: https://docs.opencv.org/3.4.3/dc/dbb/tutorial_py_calibration.html

Lesson 3 Part 1: Visual Depth Perception - Stereopsis

単語
- anaglyph: a stereoscopic picture consisting of two images of the same object, taken from slightly different angles, in two complementary colors, usually red and cyan (green-blue). When viewed through spectacles having one red and one cyan lens, the images merge to produce a stereoscopic sensation.
    - redとcyanは、chromatically opposite colors.
    - Complementary colors: pairs of colors which, when combined or mixed, cancel each other out (lose hue) by producing a grayscale color like white or black. When placed next to each other, they create the strongest contrast for those two colors.

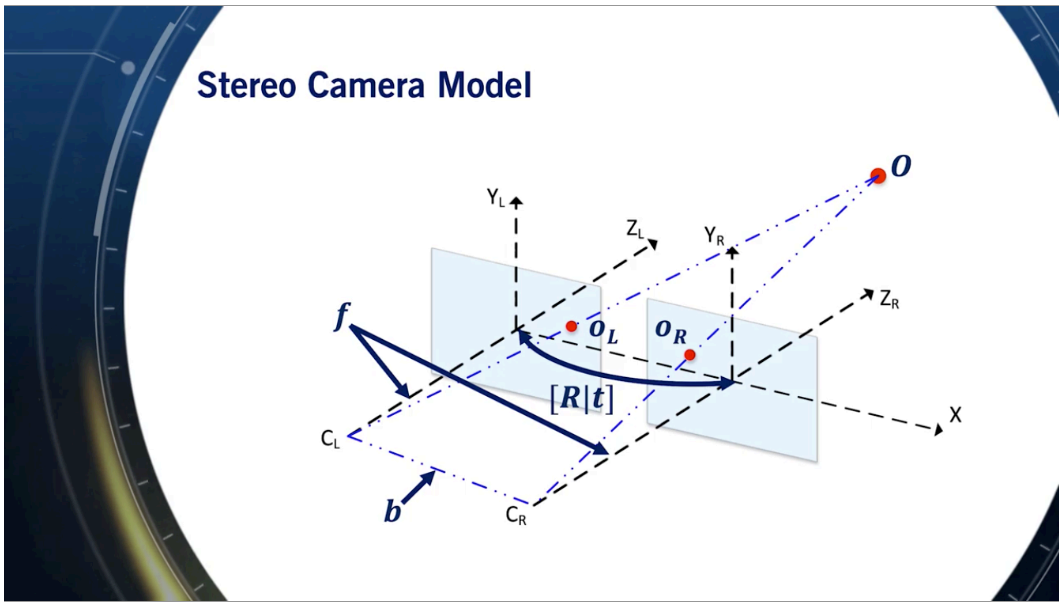


-Modern color theory uses either the RGB additive color model or the CMY subtractive color model, and in these, the complementary pairs are red-cyan, green-magenta, and blue-yellow.

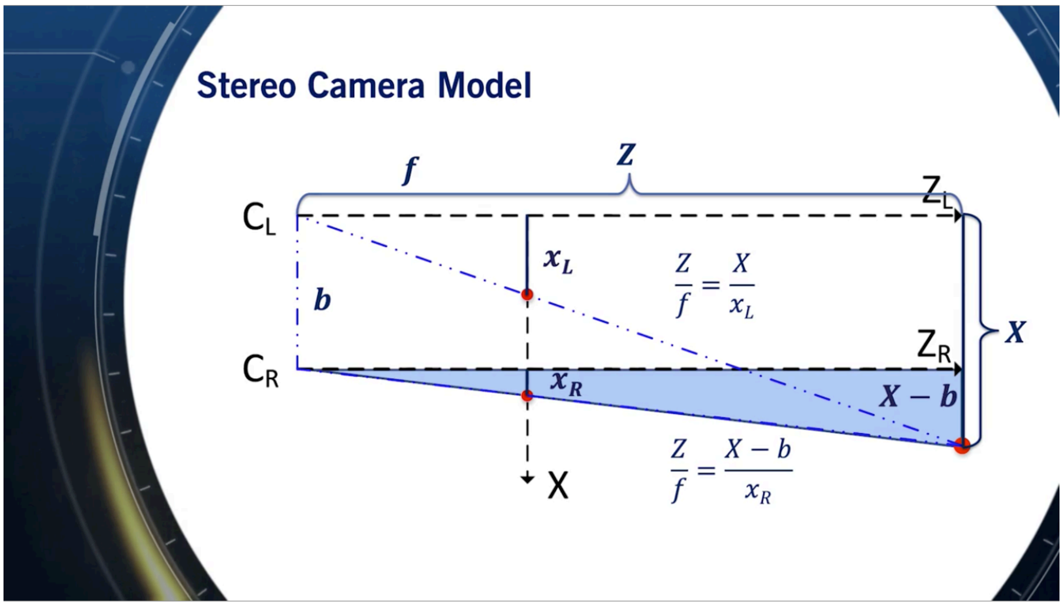
内容
-The geometry of a stereo sensor, and how the two camera are related.
-How to derive the location of a point in 3D given its projection on the two images of a stereo sensor.

Stereo Camera Model
-A stereo sensor is usually created by two cameras with parallel optical axes.
-To simply the problem even more, most manufacturers align the cameras in 3D space so that the two image planes are aligned with only an offset in the x-axis.
-Given a know rotation and translation between the two cameras and a known projection of a point O in 3D to the two camera frames resulting in pixel

locations O_L and O_R respectively, we can formulate the necessary equations to compute the 3D coordinates of the point O.



- Assumptions
  - The two cameras used to construct the stereo sensors are identical.
  - While manufacturing the stereo sensor, tried as hard as possible to keep the two cameras optical axes aligned.
- Parameters
  - b, baseline, the distance along the shared x-axis between the left and right camera centers.
    - By defining a baseline to represent the transformation between the two camera coordinate frames, we are assuming that the rotation matrix is identity and there is only a non-zero x component in the translation vector.
    - The $[R\,|\,t]$ transformation therefore boils down to a single baseline parameter b.



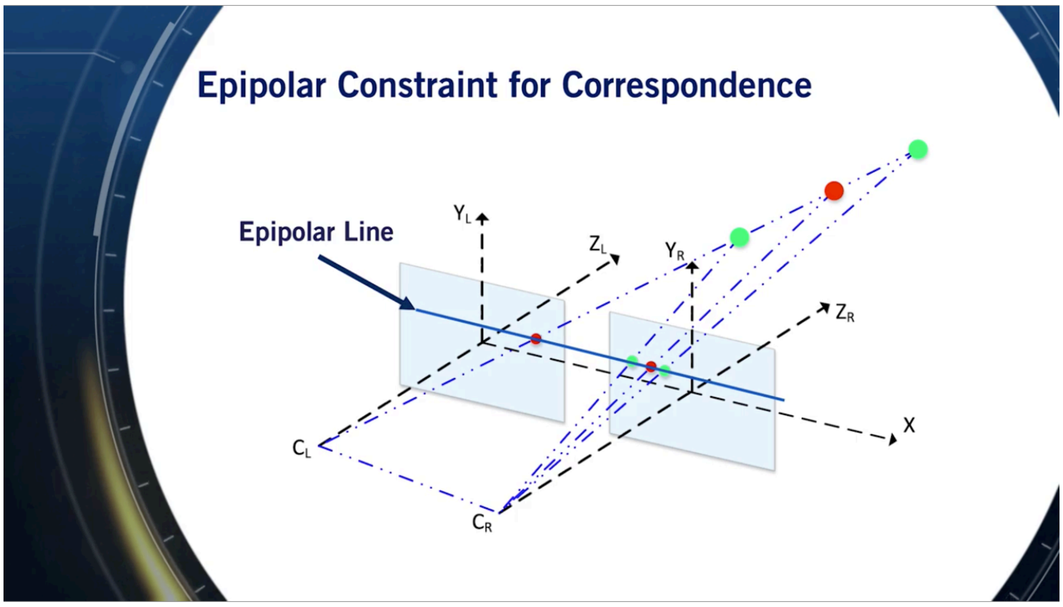
- Project to Bird’s eye view for easier geometry.
  - 問題記述：to compute the x and z coordinates of the point O with respect to the left camera frame.
  - The y coordinate can be estimated easily after the x and z coordinates are computed.
  - 上記の図に2つsimilar triangle equationsがある。

Computing 3D Point Coordinates

- Disparity: $d = x_L - x_R$.
  - the difference between the image coordinates of the same pixel in the left and right images.
  - where $\begin{matrix} x_L = u_L - u_0 \\ x_R = u_R - u_0. \\ y_L = v_L - v_0 \end{matrix}$
- Main stereo relations:
  - $\dfrac{Z}{f} = \dfrac{X}{x_L} \to Zx_L = fX.$
  - $\dfrac{Z}{f} = \dfrac{X-b}{x_R} \to Zx_R = fX - fb.$
  - $Zx_R = Zx_L - fb.$
  - $Z = \dfrac{fb}{d}.$
  - From above: $X = \dfrac{Zx_L}{f}$, $Y = \dfrac{Zy_L}{f}$.
- Two main problems:
  - We need to know f, b, u_0, v_0.
    - use stereo camera calibration.
    - for which well-established implementations are available.
  - We need to find corresponding x_R for each x_L.
    - use disparity computation algorithms.

Lesson 3 Part 2: Visual Depth Perception - Computing the Disparity

内容



Epipolar Constraint for Correspondence

- How to estimate the disparity through stereo matching.
- How to constrain disparity estimation through epipolar constraints.
  - efficient disparity estimation is possible due to epipolar constraints.

Computing 3D Point Coordinates (Review)
- Disparity: The difference in image location of the same 3D point under perspective to two different cameras.
- Correspond pixels in the left image to those in the right image to find matches.
  - The stereo correspondence problem.
- Brute Force Solution:
  - Search the whole image for each pixel?
  - It's also unlikely to succeed as many pixels will have similar local characteristics, making it difficult to match them correctly.
- Luckily, we can use stereo geometry to constrain the search problem from 2D over the entire image space to a 1D line.
- Why such a simplification is valid?

Epipolar Constraint for Correspondence
- Move the 3D point along the line connecting it with the left camera's center.
  - The projection on the right camera plane, moves along the horizontal line.
    - Epipolar line.
    - We can constrain our correspondence search to be along the epipolar line, reducing the search from 2D to 1D.

Non-Parallel Optical Axes
- Horizontal epipolar lines only occur when the optical axes of the two cameras are parallel.
- If this condition is not met, epipolar lines will be skewed.
  - Known as multi-view geometry.

Disparity Computation
- In the case of stereo camera, a skewed epipolar line is not a huge problem.
- We can use stereo rectification to warp images originating from two cameras with non-parallel optical axes to force epipolar lines to be horizontal.
- stereo rectificationの実現も既にある。(available in standard computer vision packages such as OpenCV and MATLAB)

A Basic Stereo Algorithm
Given: Rectified Images and Stereo Calibration.
For each epipolar line,
1. Take each pixel on this line in the left image.
2. Compare these left image pixels to every pixel in the right image on the same epipolar line.
3. Pick the pixel that has minimum cost.
   1. for example, a very simple cost can be the squared difference in pixel intensities.
   2. pixel intensityは多分pixelのvalue。例えばgrayscaleの0~255?
4. Compute disparity, d.

Stereo Matching
- As with most problems in computer vision, the stereo vision algorithms are evaluated on a public benchmark.
  - The most famous of which is the middlebury stereo benchmark.
- Stereo matching is a very well-studied problem in computer vision.

- Disparity estimation can be performed through stereo matching algorithms.
- Efficient solutions exist as the problem is constrained with epipolar constraints.

Lesson 4: Image Filtering

単語

- finite difference, 有限差分: Three types are commonly considered: forward, backward, and central finite differences.
  - forward difference, $\Delta_h[f](x) = f(x+h) - f(x)$.
  - backward difference, $\nabla_h[f](x) = f(x) - f(x-h)$.
  - central difference, $\delta_h[f](x) = f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right)$.
- Sobel operator, Sobel-Feldman operator, Sobel fiter: is used particularly within edge detection algorithms where it creates an image emphasizing edges.
  - The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical.
  - $G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$.
  - $G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$.
  - Aはsource image, G_x, G_yはtwo images which at each point contain the vertical and horizontal derivative approximations respectively.

内容
- Perform image filtering through cross-correlation and convolution operations.
- Some uses for these operations in context of image understanding.

Image Filtering
- Image Filtering is a simple and efficient method to eliminate noise.
- How image filtering helps reduce salt and pepper noise.
- outlier pixels, low-value pixels in a high-value neighborhood, or high-value pixels in a low-value neighborhood.
- One idea to reduce this noise is to compute the mean of the whole neighborhood, and replace the outlier pixel with this mean value.
- Mean of the whole neighborhood: $G[u,v] = \frac{1}{(2k+1)^2} \sum_{i=-k}^{k} \sum_{j=-k}^{k} I[u-i, v-j]$.
  - $(2k+1)$: filter size.
  - (u,v): center pixel coordinates.

Cross-Correlation
- The mean equation can be generalized by adding a weight to every pixel in the neighborhood:
  $G[u,v] = \sum_{i=-k}^{k} \sum_{j=-k}^{k} H[i,j] I[u-i, v-j]$.
  - The weight matrix H is called a kernel.
- This generalized form is termed cross-correlation, as it defines a correlation between each pixel and every other pixel in the neighborhood.
- Mean filter: $H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.
  - 効果：It also blur the image, an inevitable consequence of these linear filters.
- Gaussian filter: $H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$.

- where the center pixel is weighted more than the neighboring pixels and the weights follow a Gaussian distribution.

Convolution

- A convolution is a cross-correlation where the filter is flipped both horizontally and vertically before being applied to the image.
  - 分かりにくいので、例は:

$$\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)[2,2] = (i \cdot 1) + (h \cdot 2) + (g \cdot 3) + (f \cdot 4) + (e \cdot 5) + (d \cdot 6) + (c \cdot 7) + (b \cdot 8) + (a \cdot 9)$$

  - $[2,2]$の意味は真ん中のelement (of the resulting image)のvalueはこの式の結果だよ。
  - 左側のmatrixはkernel、右側のmatrixはimage piece.
  - なぜわざわざひっくり返さないといけないだろうと思われるかもしれないが、下のメリットがあるから。

- $$G[u,v] = \sum_{i=-k}^{k} \sum_{j=-k}^{k} H[i,j]I[u-i, v-j].$$
  - flipped both horizontally and verticallyはHとIのindexからわかるでしょう。
  - simply manipulated the image coordinates instead of flipping the kernel.
- What are the advantages of using a convolution over a kernel?
  - Unlike Cross-Correlation, Convolution is associative.
    - If H and F are filter kernels, then: $H * (F * I) = (H * F) * I$.
    - We can therefore apply as many consecutive linear kernels to an image as we want by precomputing the combined convolution of all the kernels.
    - Precompute filter convolutions $(H * F)$ then apply it once to the image to reduce runtime.

Applications: Template Matching

問題記述：Template matching is the problem where given a pattern or a template, and we want to find its location in the image.

- This can be done by finding the location of the highest value of the output of cross-correlation, between the template and the image.
- Cross-correlation is useful for template matching.
- The pixel with the highest response from Cross-correlation is the location of the template in an image.
- This method can be used as a starting point for the identification of signs, and even for lane detection.

Applications: Gradient Computation

- Image gradients can be computed by a convolution with a kernel that performs finite difference.
- We can rotate the kernel in different orientations to get vertical, horizontal or even diagonal gradients of an image at a pixel.
- Image gradients are extremely useful for detection of edges and corners.
- Define a finite difference kernel, and apply it to the image to get the image gradient.
- Finite difference kernel, 例えばHorizontal Sobel Kernel.（単語部分を参考）

- Cross-Correlation and Convolution are two operations that can be used to apply a filter to an image.
- Cross-Correlation can be used to match image regions, while convolutions can be used for edge detection.

学んだこと：

1. 一番大事なのはdisparity_mapの計算です。

cv2.StereoSGBM_create()を使っても、パラメータをチューニングしないといけない。

disparity_mapのsmoothnessをコントロールできるパラメータをチューニングする。
これをしないとdisparity_mapはnoise多いでしょう。

例えばcv2.StereoSGBM_create()の引数の中に、
P1: The first parameter controlling the disparity smoothness.
P2: The second parameter controlling the disparity smoothness.

- The larger the values are, the smoother the disparity is.
- P1 is the penalty on the disparity change by +1 or -1 between neighbor pixels.
- P2 is the penalty on the disparity change by more than 1 between neighbor pixels.
- The algorithm requires P2 > P1.
- See stereo_match.cpp sample where some reasonably good P1 and P2 values are shown.
- Like $8 * number\_of\_image\_channels * SADWindowSize^2$ and $32 * number\_of\_image\_channels * SADWindowSize^2$, respectively.

2. cv2.matchTemplateを使う時、cross-correlationですが、method=cv2.TM_CCOEFFが使われている。cv2.TM_CCORRは使われてない。

3. Kの中に$f_x \neq f_y$かもしれないが、disparityはx方向なので、f_xを使えばいい。

4. https://en.wikipedia.org/wiki/Camera_resectioning

- t is the position of the origin of the world coordinates system expressed in coordinates of the camera-centered coordinate system.
- t is often mistakenly considered the position of the camera.