What does the robot sense?

- "Where" geometry
- "What" semantics
- "Who"

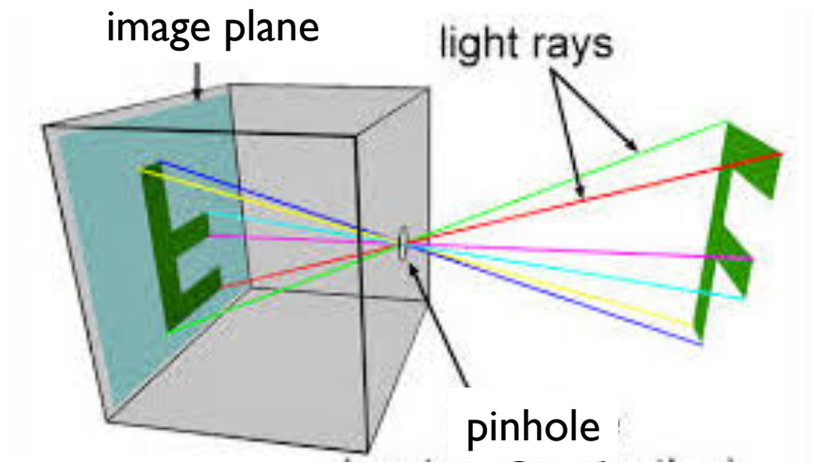
Cameras. Visual cameras provide **rich information** at a moderate cost.

webcam

GoPro video camera

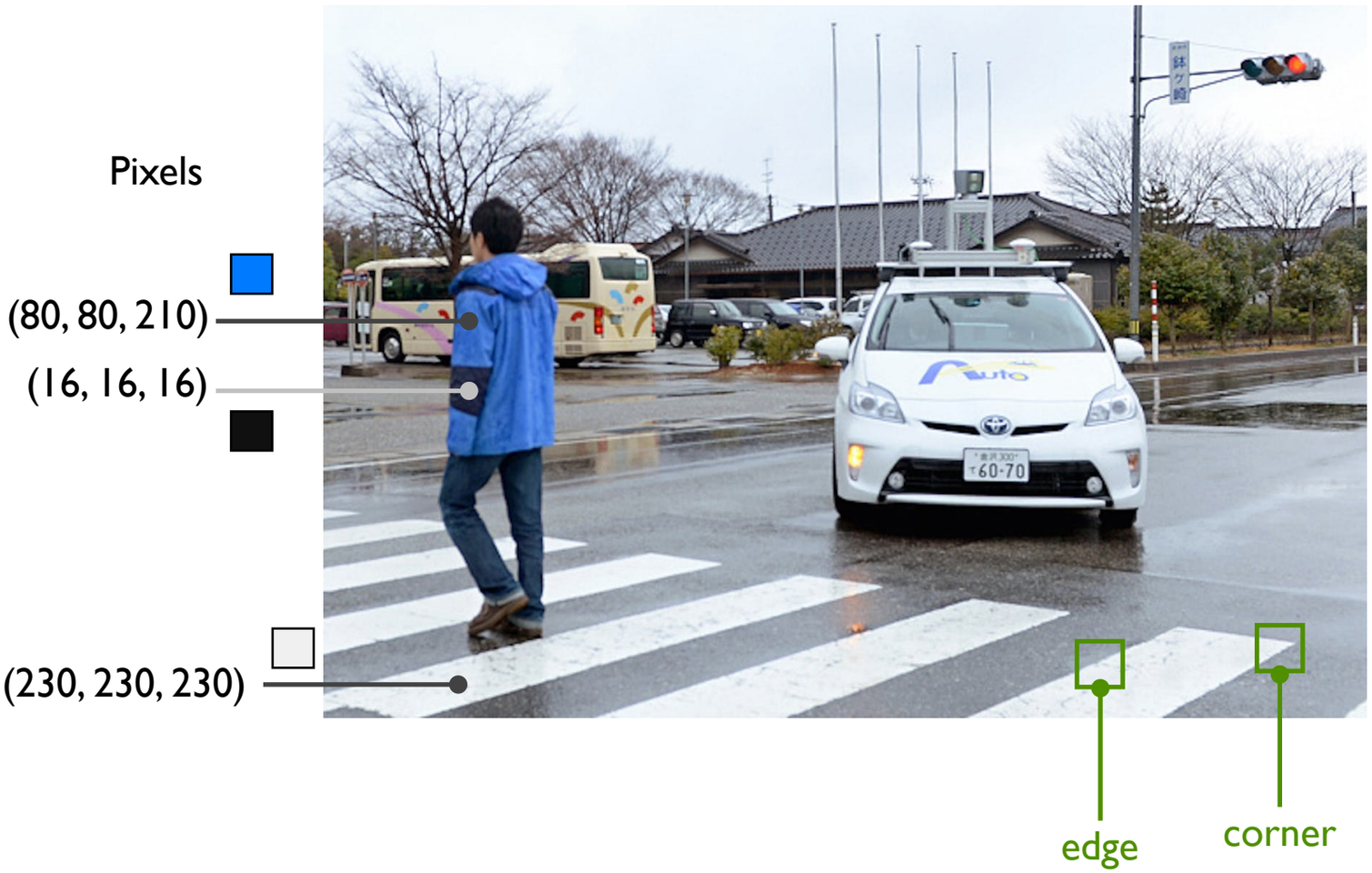
omnidirectional camera

Light goes through the camera lens according to the principles of optics and geometry and gets recorded on the image plane.



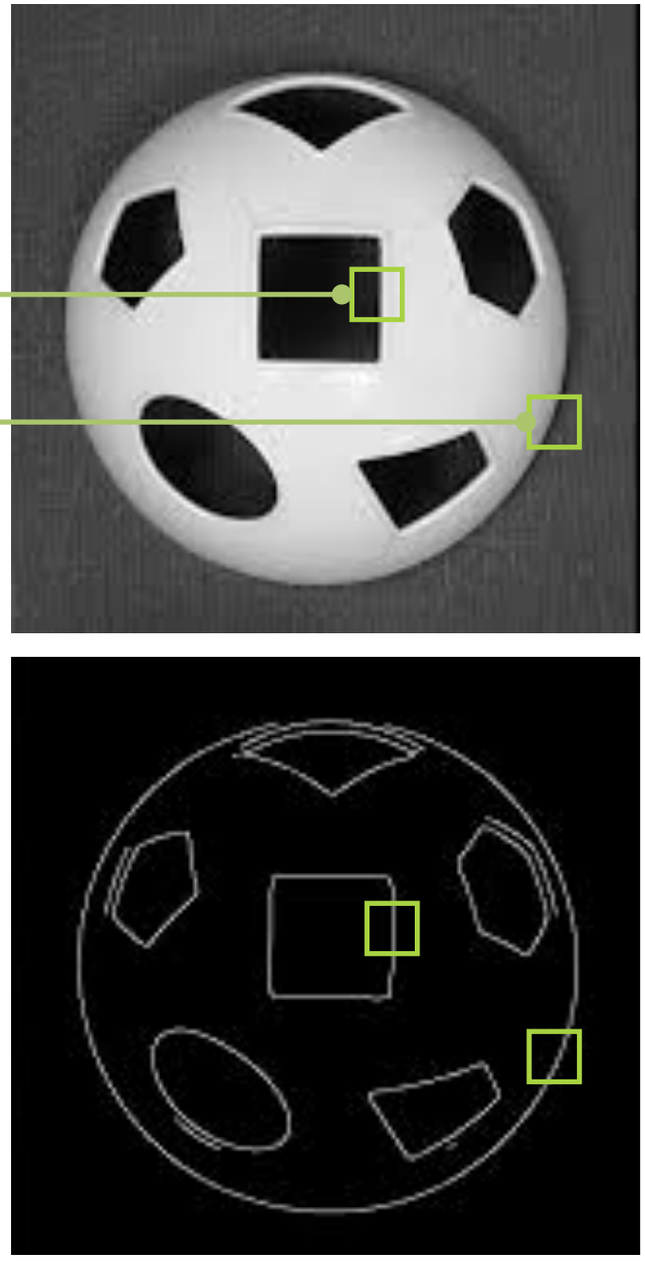
pinhole camera model

Images. An image is an array of pixel values (r,g,b). An image feature is an image patch with distinct visual characteristics that can be easily **identified** and **localized**. Image features can be grouped to identify objects, people, …



An image is an indexed array of RGB values.

Edge detection. For simplicity, consider gray-scale images. An image is then a function $I(u, v)$. An edge is an image location (u, v) where significant change in the **image brightness** occurs.



In general, we compute the gradient of $I(u, v)$ to determine the location and the orientation of the edge.

There are many other classes of features:

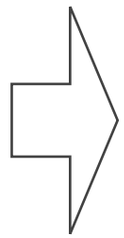
- “flat” region, i.e., a small patch of uniform color
- Corners
- Harris
- FAST
- SIFT
- MSER
- SURF
- …

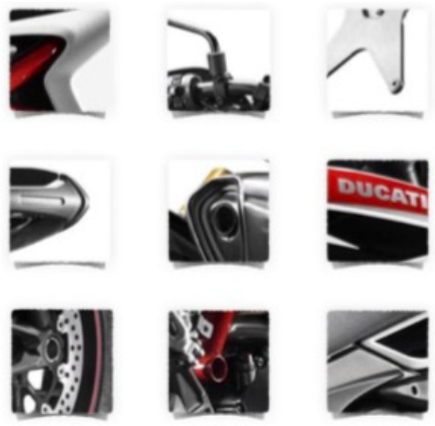
They differ in robustness, localization accuracy, abundance, and computational efficiency.

After feature extraction, we represent an image as a set of features, possibly with feature locations.

An image is a set of features and sometimes together with feature locations.

Object recognition or place recognition. A bag of features represents an object as a set of local features without their spatial locations in the image.





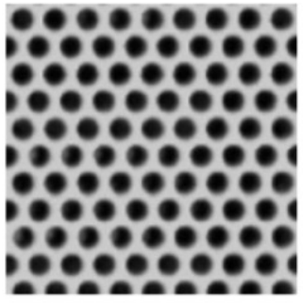
While this may sound surprising initially, it actually works.
What is the object below?

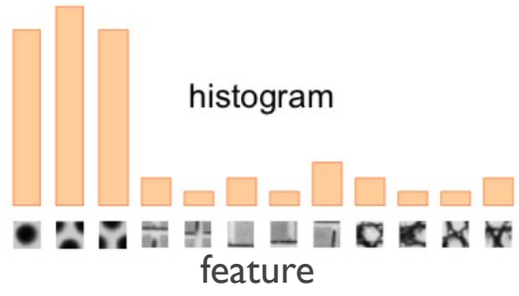
While the eyes, ears, … are not in their usual spatial arrangement, we nevertheless recognize that the image most likely represents a human face. Similarly, we can represent the visual appearance of a physical place as a bag of features.

Image credit: Kris Kitani

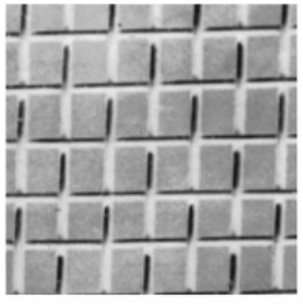
Consider the three object types below, each represented as a histogram over a set of features.

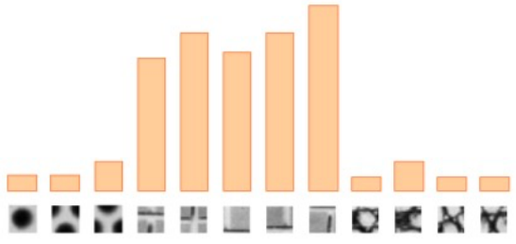
metal grid



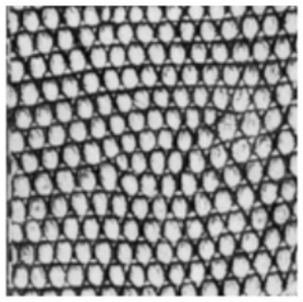


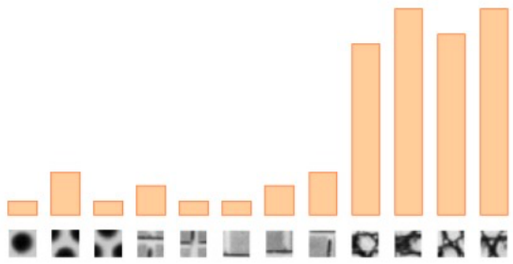
brick wall





wire mesh





Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be two feature frequency vectors. A common distance measure between $\mathbf{v}_1$ and $\mathbf{v}_2$ is

$$d(\mathbf{v}_1, \mathbf{v}_2) = \cos(\theta) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|}$$

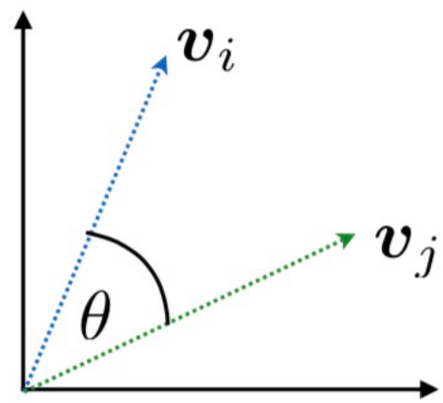


Image credit: Kris Kitani

Some feature vectors are very "close", in particular, when two vectors may represent slightly different views of the same object, maybe under different lighting conditions. We want to group them together.

Learn the visual vocabulary. We apply K-means clustering to the set of feature vectors and create a visual vocabulary dictionary.

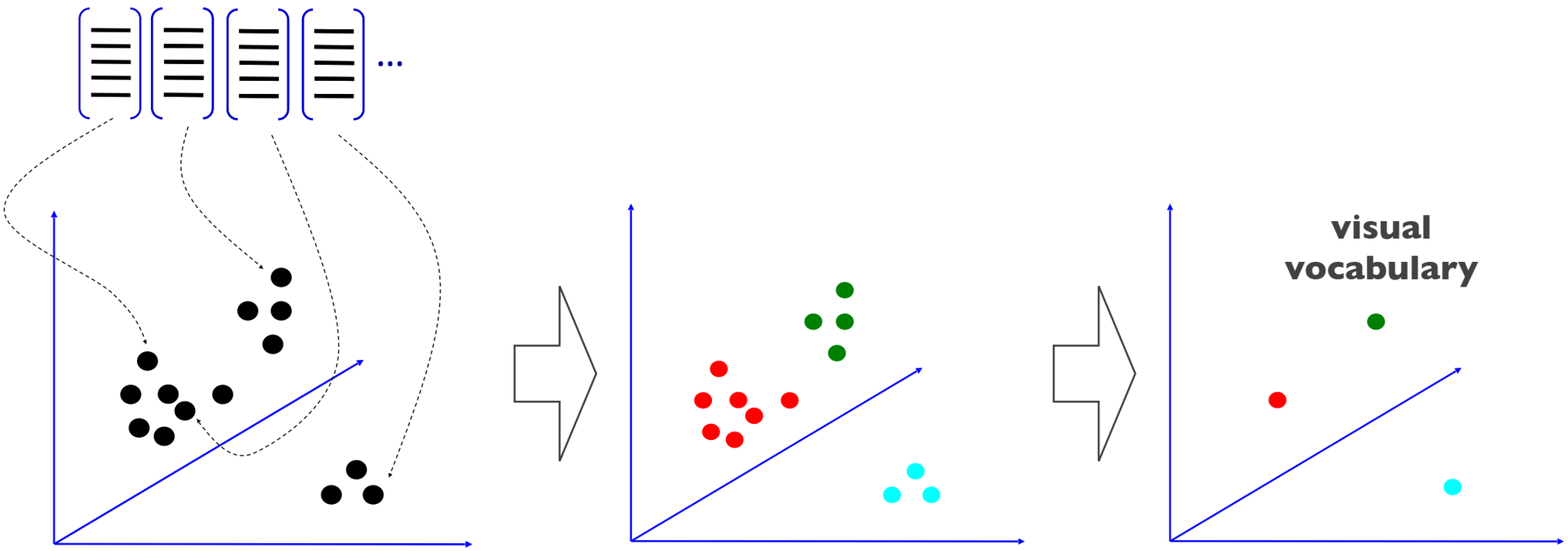


Image credit: Kris Kitani

K-means clustering is an unsupervised learning algorithm. Given a set of points and an integer $K \geq 1$, it divides the point set into K clusters so as to minimize the distance of each point to the cluster "center".

- To initialize, assign each point to a cluster randomly.
- Compute the mean of each cluster.
- Assign each point to the nearest cluster mean.
- Repeat the 2 previous steps until no change occurs.

Bag of visual words.

1. Extract features.
2. Learn the visual vocabulary.
3. Represent an image as frequencies of visual words.

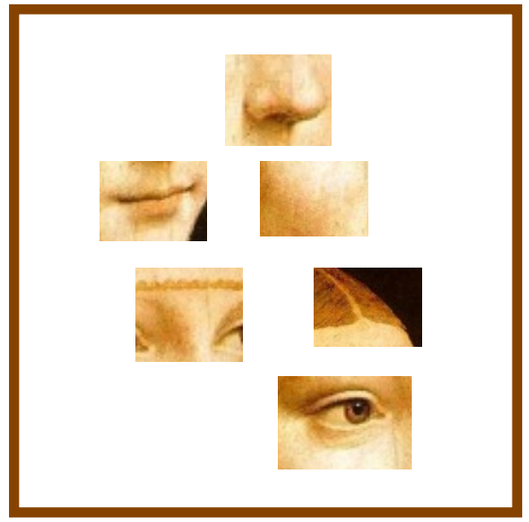


Image credit: Kris Kitani

Bag of visual words.

1. Extract features.
2. Learn the visual vocabulary.
3. Represent an image as frequencies of visual words.

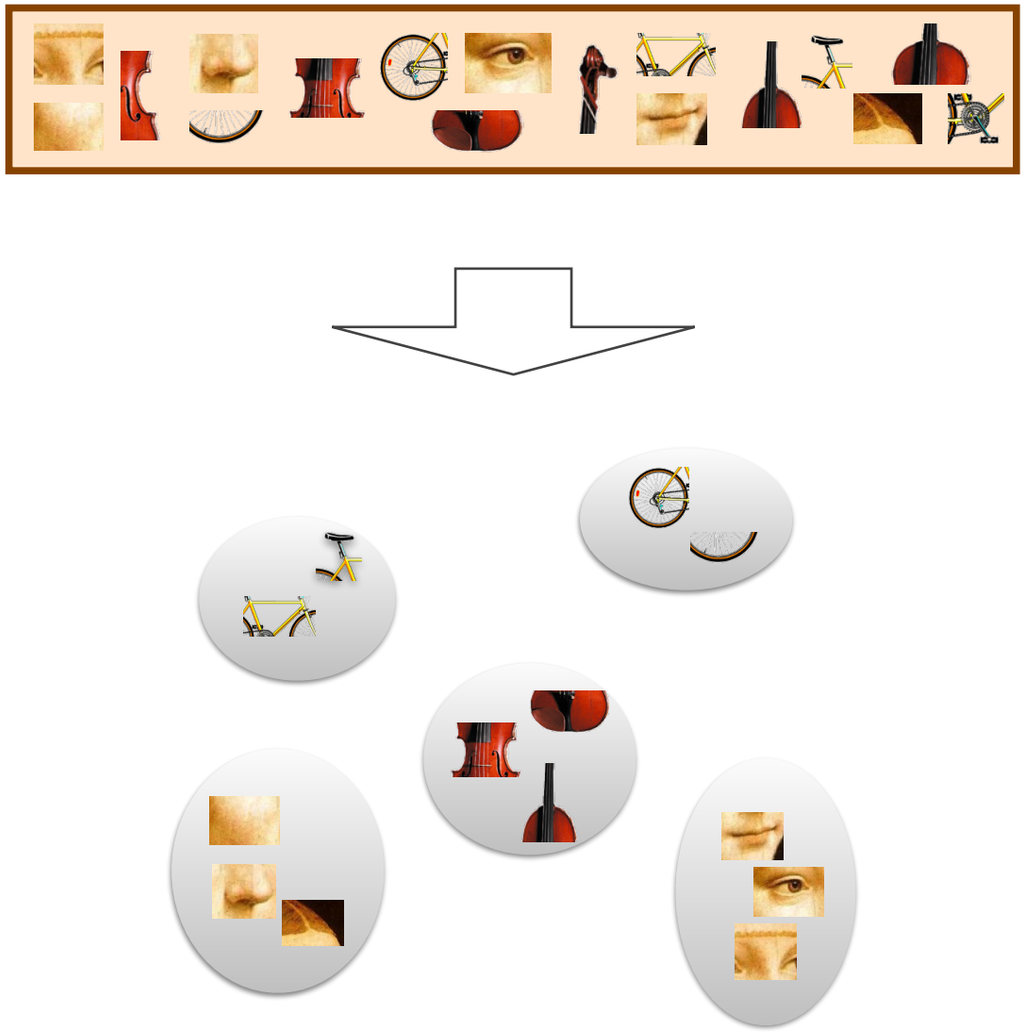
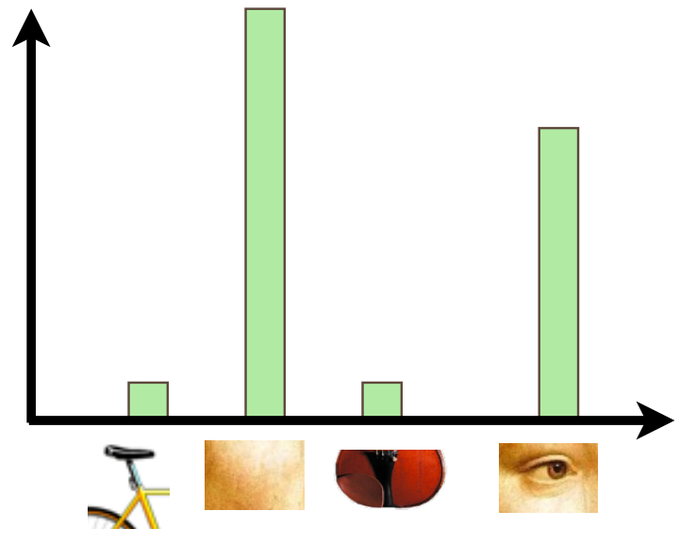


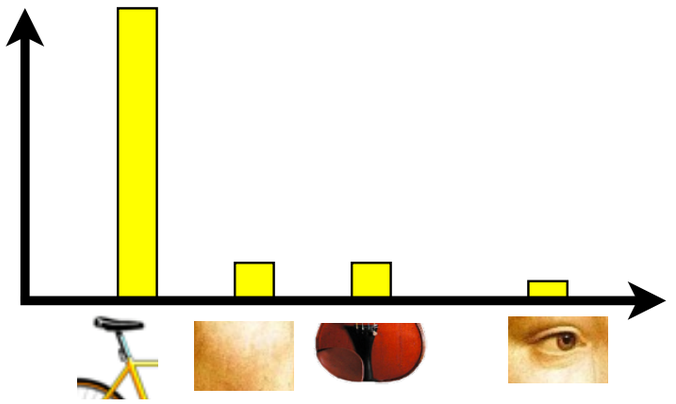
Image credit: Kris Kitani

Bag of visual words.

1. Extract features.
2. Learn the visual vocabulary.
3. Represent an image as frequencies of visual words.



Choose K. The parameter K determines the size of the vocabulary and, in turn, the accuracy and the efficiency of image representation. If K is too small, then the image cannot be accurately represented. Consider, for example, the extreme case if $K=1$. If K is too large, the image is over-represented and susceptible to noise.



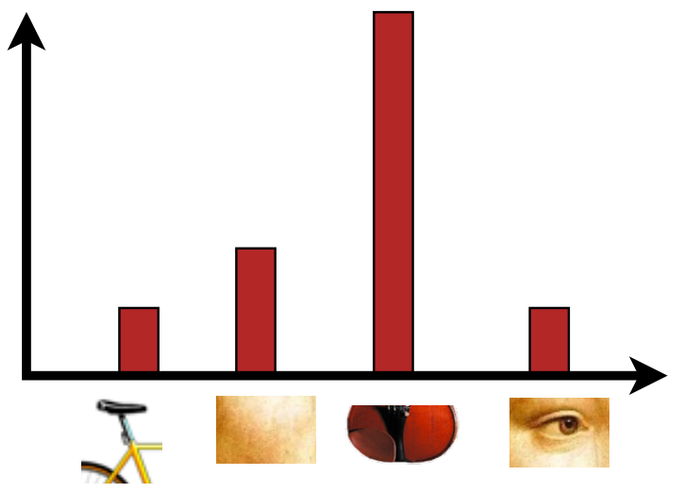


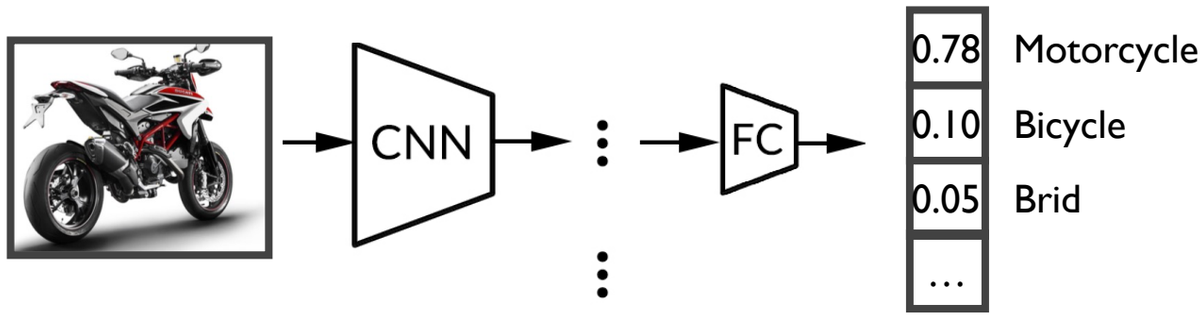
Image credit: Kris Kitani

The bag-of-words approach ignores spatial information of features. Suppose that our task is to differentiate dogs and chicken. A good feature is the legs. Counting the number of legs is sufficient to recognize dogs versus chicken.

Sometimes, spatial information is important. Now our task is to determine whether a table can stand properly. Again, the feature is the legs. Counting the number of legs is useful. A table with two legs cannot stand. However, a table with more than two legs may still fail to stand if they are not in a proper spatial arrangement.

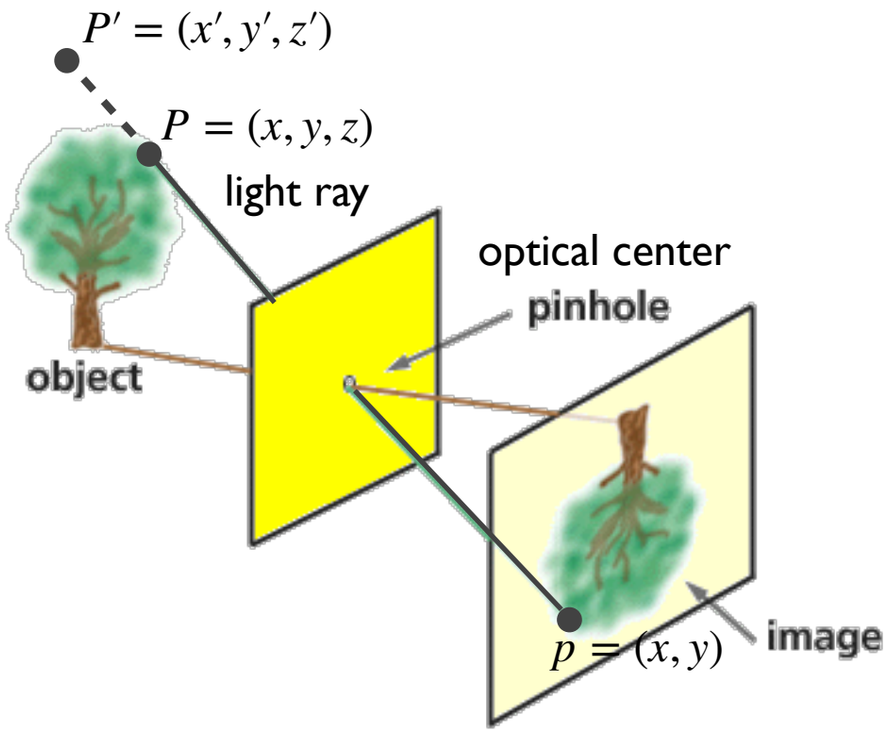
We can use the bag-of-words approach to identify some candidates and then refine using spatial information.

Deep learning for image classification.



Question. How does the bag of visual words compare with deep learning for object or place recognition?

The geometry of image formation. To acquire spatial information in 3-D from a 2-D image, we need to understand the relationship between points in the 3-D world and points in the corresponding 2-D image.

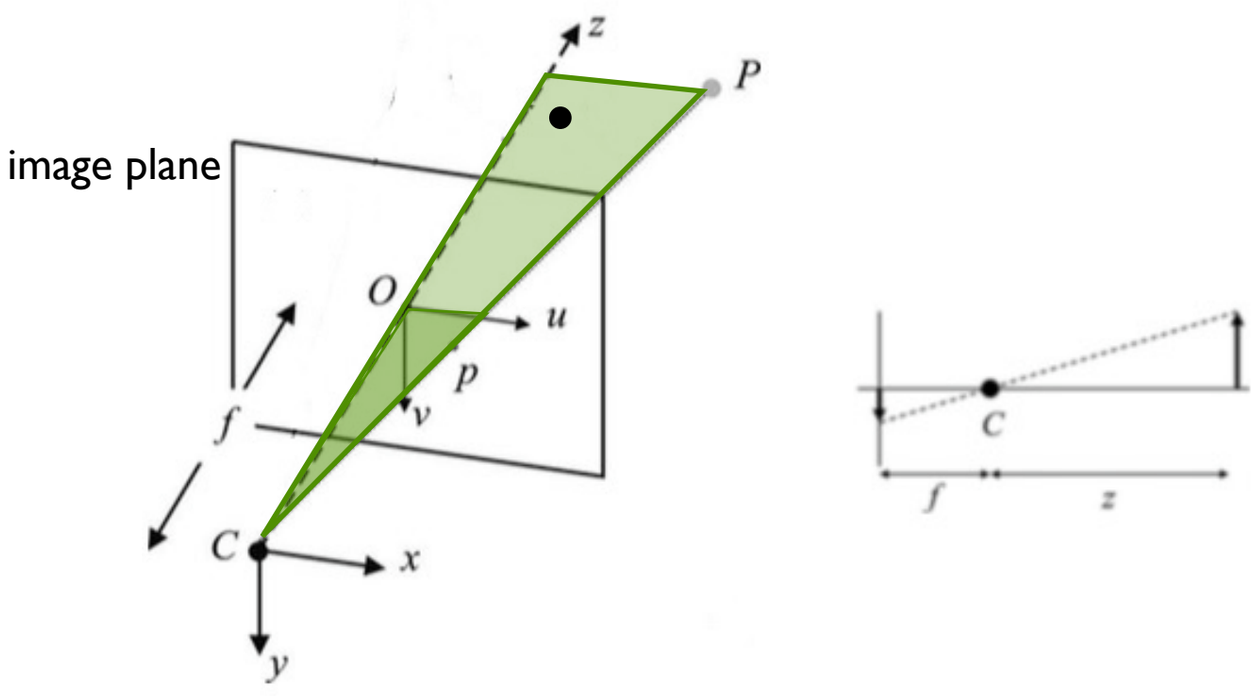


The pinhole camera model is a simple model of image formation. It captures the **geometry** of image formation.

We cannot differentiate the points along the same light ray, as they all project to the same point in the image. In this sense, the camera is a **bearing** sensor and provides **no depth** information. So the image provides geometric information up to a scale factor.

The earlier statement is not entirely true. A more sophisticated model of image formation captures the **optics**, in addition to the geometry of image formation. Using such a model, we can recover some depth information, but it requires hard work. For example, the camera autofocusing system does so by actively adjusting the focal length to make the object of interest appear sharp in the image. This is called depth from focus.

In general, let $P = (x, y, z)$ be point in the 3-D world and $p = (u, v)$ be the corresponding projected point in the 2-D image.



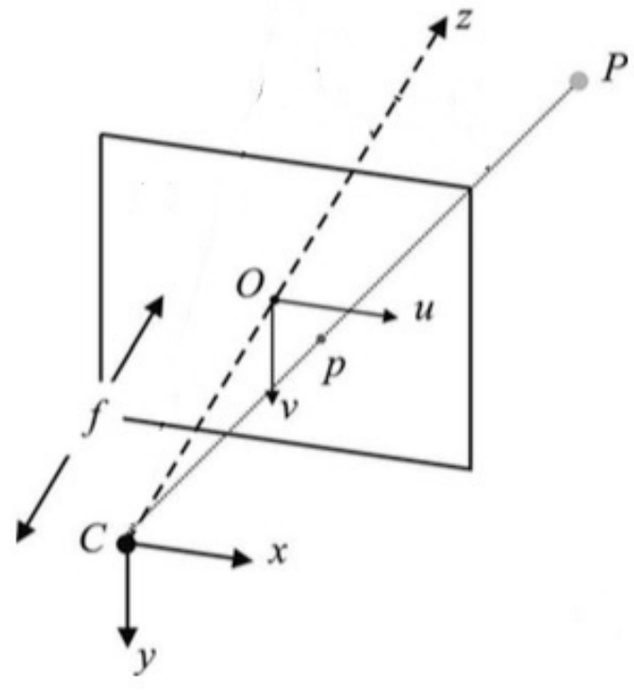
The camera coordinate system has the x-y plane parallel to the image plane.

By the similarity of triangles, we have

$$\frac{f}{z} = \frac{u}{x} = \frac{v}{y}$$

Therefore, the perspective projection gives

$$u = \frac{f}{z}x \quad \text{and} \quad v = \frac{f}{z}y\,.$$



To rewrite the above in a linear form. We use homogeneous coordinates:

$$\begin{bmatrix} u \\ v \end{bmatrix} \longrightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \end{bmatrix} \longrightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

In homogeneous coordinates,

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda z \\ \lambda \end{bmatrix} \qquad \text{for all } \lambda \neq 0.$$

Now we rewrite the equations for u and v as

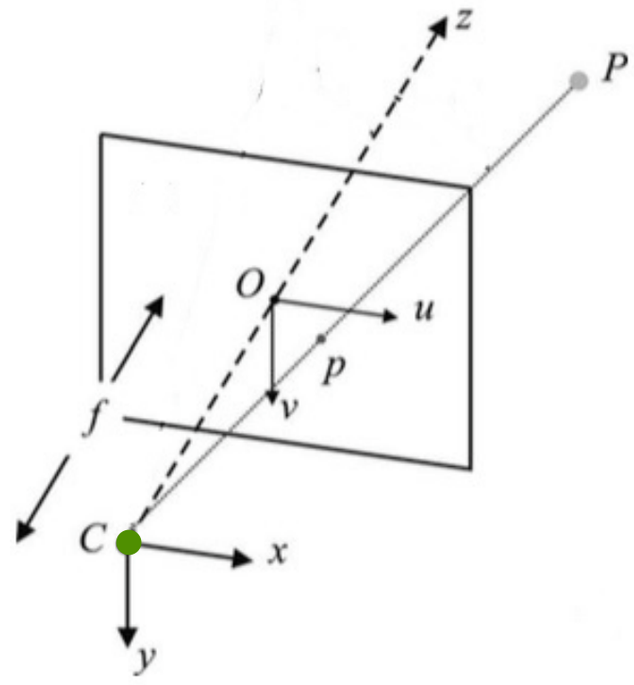
divide by λ — divide by z — We get back the perspective project equations.

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

To figure out (x, y, z) from (u, v), we need to know f. In addition, we only know (x, y, z) up to a scale factor proportional to the depth z.

Our simplified perspective projection equations do not account for the following:

- By convention, the origin of the image coordinate system is located at the lower left corner of the image. The optical center goes through the point O with coordinates (u_0, v_0).
- Image coordinates are measured in pixels. We need two scale factors k_u and k_v proportional to pixel width and height, respectively.



The perspective projection equations now become

$$u = \frac{k_u f}{z} x + u_0 \quad \text{and} \quad v = \frac{k_v f}{z} y + v_0 \,.$$

In homogeneous coordinates,

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} k_u f & 0 & u_0 & 0 \\ 0 & k_v f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Now, to figure out (x, y, z) from (u, v), we need values for 5 different parameters: $f, k_u, k_v, u_0,$ and v_0.

We can combine some parameters by setting $\alpha_u = k_u f$ and $\alpha_v = k_v f$. Then,

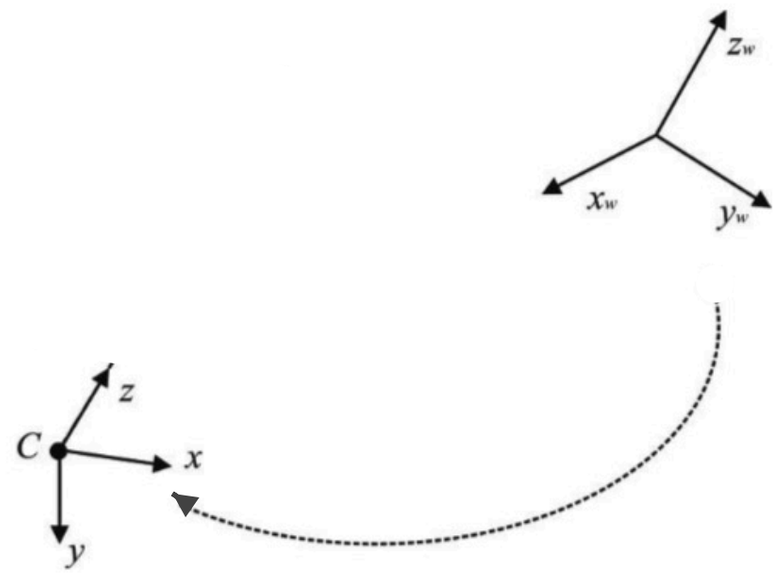
$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

perspective projection

In addition, we must account for the camera pose in the world:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_{\mathrm{w}} \\ y_{\mathrm{w}} \\ z_{\mathrm{w}} \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

rotation translation

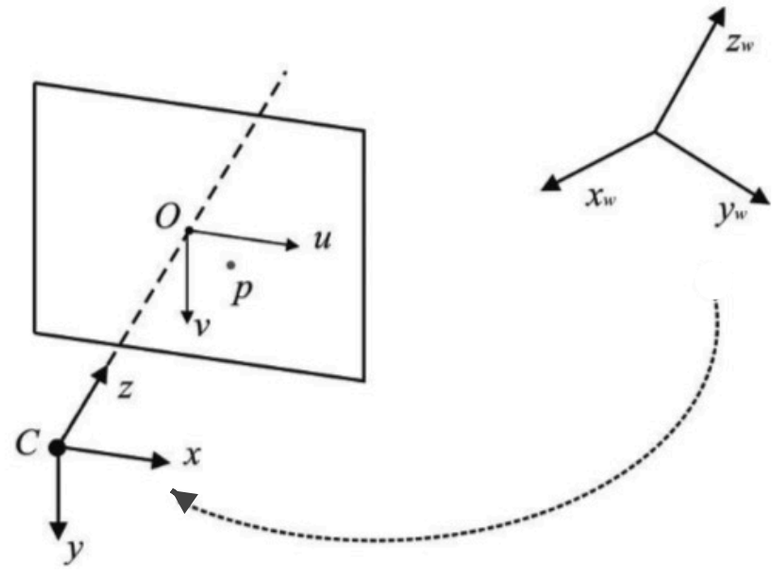


Putting the two equations together, we have

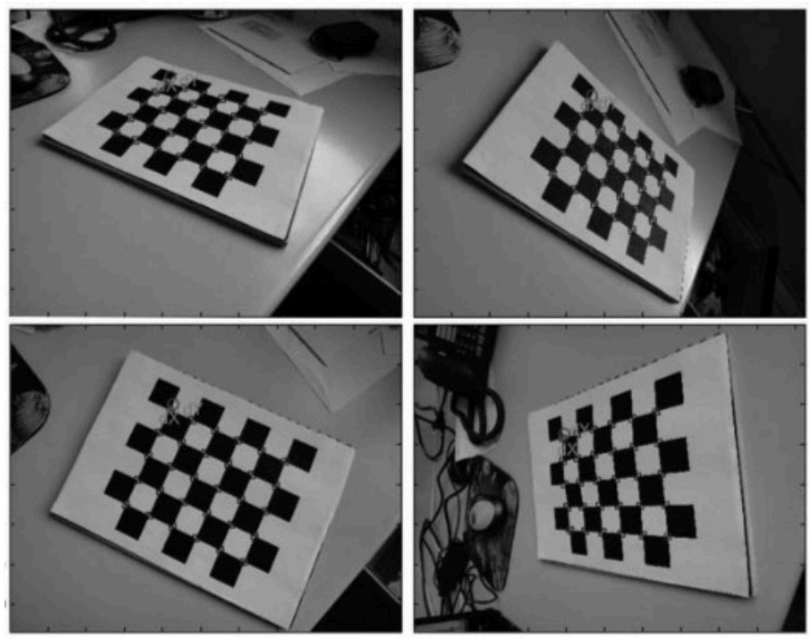
$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_{\mathrm{w}} \\ y_{\mathrm{w}} \\ z_{\mathrm{w}} \\ 1 \end{bmatrix}$$

intrinsic camera parameters

extrinsic camera parameters

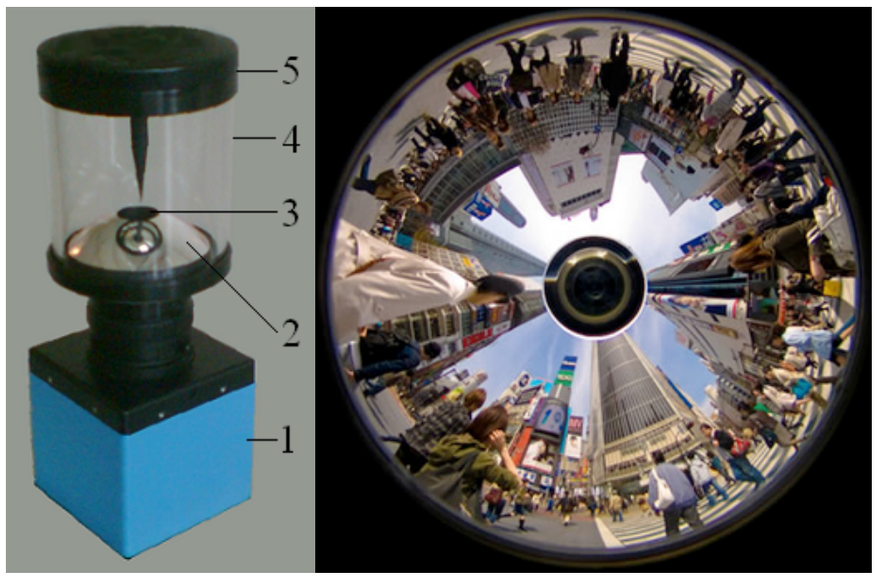


Camera calibration. Camera calibration determines the intrinsic as well as extrinsic parameters of the camera model. The well-known planar grid method uses several images of a planar checkerboard in different poses. It is simple and practical.



Omnidirectional cameras.

catadioptric camera



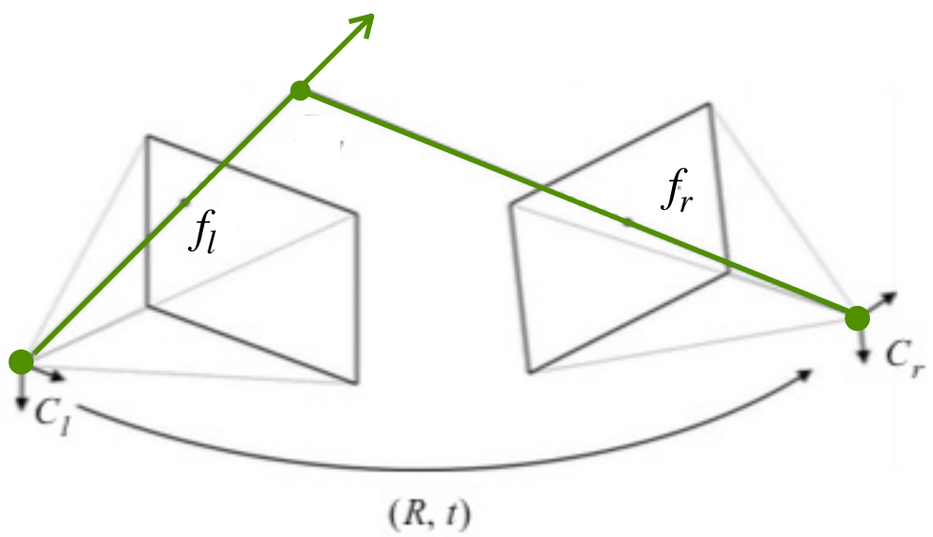
polydioptric camera

An omnidirectional camera provides a wide field of view, at least 180 degrees.

- Dioptric cameras use multiple specially shaped lenses to achieve a wide field of view.
- Catadioptric cameras combine a standard camera with a shaped a parabolic, hyperbolic, or elliptical mirror.
- Polydioptric cameras use multiple cameras with overlapping field of view. They can provide a real omnidirectional (spherical) view.

Obviously, the geometry of image formation for omnidirectional camera is very different from the linear projection model that we have had.

Structure from stereo. Using one camera, we can determine the 3-D location of a point up to a ray that goes through the camera optical center. Suppose that two or more cameras, with known poses, look at the the same point. Then we can recover the full 3-D location of the point, including, in particular, the depth. The human vision system works this way.



We can do this for all points in an environment and thus reconstruct the 3-D structure, i.e., determine the 3-D position of all points of interest in the environment.

In summary, given two images, represented as corresponding feature pairs obtained from two cameras with known relative pose, we can reconstruct the 3-D environment.

In fact, we can do more. Suppose that the relative translation t and the relative rotation R between the two cameras are not known. Can we solve the problem?

The answer is yes, if there are enough feature pairs. We just treat t and R as additional unknowns. Every feature pair adds additional constraints. With enough feature pairs, there are more constraints than unknowns, and we can solve the problem. That is the intuition behind the highly efficient eight-point algorithm, which requires the minimum of eight feature pairs.

Structure from motion and visual odometry. Instead of multiple cameras, we can use a single moving camera to generate the multiple images required, e.g., a camera mounted on a moving vehicle. There are two closely related variants:

- Structure from motion (SfM). The primary objective is to reconstruct the 3-D environment.
- Visual Odometry. The primary objective is the motion trajectory, i.e., the relative motion between successive image frames.

Visual odometry allows us to estimate the relative motion from two successive images instead of counting wheels turns using an encoder. Hence the name.

Question. What are the main sources of error for wheel odometry and visual odometry? How do they compare in accuracy?

Question. Shall I use cameras or LIDAR to acquire 3-D information?

There is no simple answer to this question. For both structure from stereo or structure from motion, one key issue is to find corresponding **feature** pairs (f_l, f_r) and their spatial location reliably and accurately. This is challenging because of image noise.

While there is a lot of excellent work in computer vision devoted to this topic, LIDAR usually provides better results. However, LIDAR is bulky and expensive.