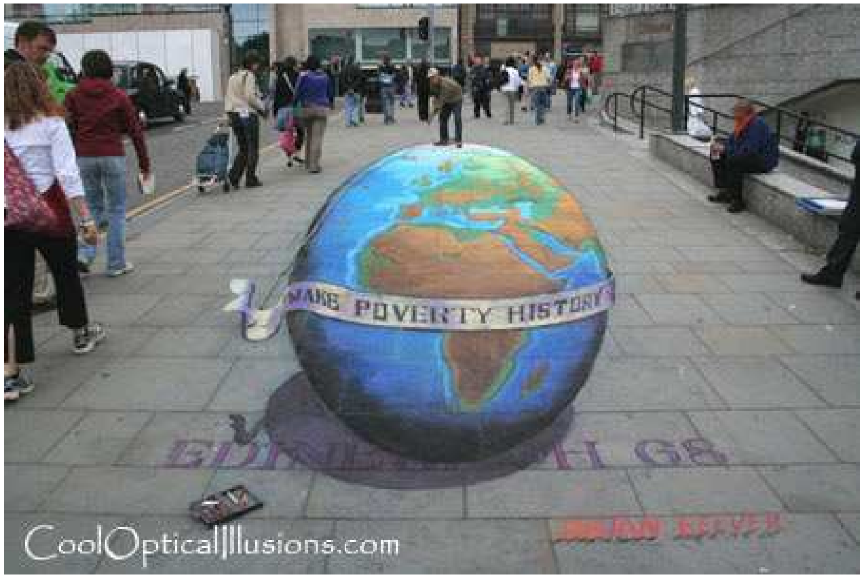
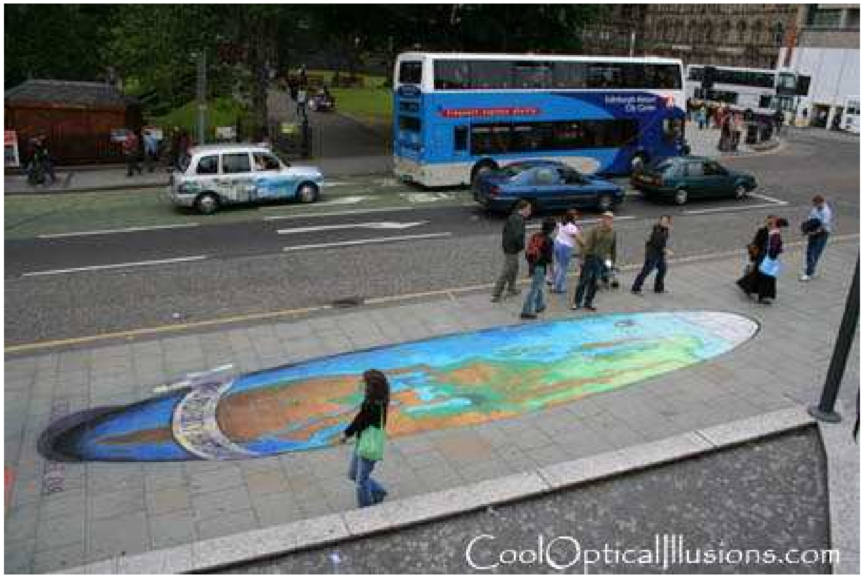
Image Projection

Goal: Introduce the basic concepts and mathematics for image projection.

Motivation: The mathematics of image projection allow us to answer two questions:



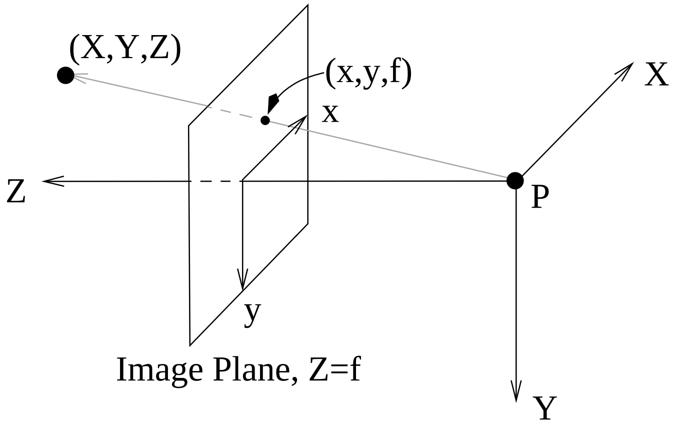


- Given a 3D scene, how does it project to the image plane? (“Forward” model.)
- Given an image, what 3D scenes could project to it? (“Inverse” model.) Vision is all about guessing the scene and the story behind it. The latter is a (largely ignored) holy grail of computer vision.

Readings: Szeliski, Chapter 2.

The Pinhole Camera

Image formation can be approximated with a simple pinhole camera,



The image position for the 3D point (X, Y, Z) is given by the projective transformation

$$\begin{pmatrix} x \\ y \\ f \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

The distance between the image plane and the projective point P is called the "focal length," f. Note:

- for mathematical convenience we put the image plane in front of the nodal point (since this avoids the need to flip the image coords about the origin);
- image coordinate x is taken to the right, and y *downwards*. This agrees with the standard raster order and the convention of a right-handed coordinate frame (X, Y, Z).
- the primary approximation here is that there is no optical blur, distortion, or defocus (discussed later).

Coordinate Frames

Consider the three coordinate frames:

- **World coordinate frame**, $\vec{X}_w$. These are 3D coordinates fixed in the world, say with respect to one corner of the room.
- **Camera coordinate frame**, $\vec{X}_c$. These are 3D coordinates fixed in the camera. The origin of the camera coordinates is at the center of projection of the camera (say at $\vec{d}_w$ in world coords). The z-axis is taken to be the optical axis of the camera (with points in front of the camera in the positive z direction).
- **Image coordinate frame**, $\vec{p}$. The image coordinates are written as a 3-vector, $\vec{p} = (p_1, p_2, 1)^T$, with p_1 and p_2 the pixel coordinates of the image point. Here the origin is in the top-left corner of the image (or, in Matlab, the top-left corner has pixel coords (1,1)). The first image coordinate p_1 increases to the right, and p_2 increases downwards.

Next we express the transforms from world coordinates to camera coordinates and then to image coordinates.

Extrinsic Calibration Matrix

The extrinsic calibration parameters specify the transformation from world to camera coordinates, which is a standard 3D coordinate transformation,

$$\vec{X}_c \;=\; M_{ex}[\vec{X}_w^T, 1]^T. \tag{1}$$

Here the extrinsic calibration matrix M_{ex} is a 3×4 matrix of the form

$$M_{ex} \;=\; \begin{pmatrix} R & -R\vec{d}_w \end{pmatrix}, \tag{2}$$

with R is a 3×3 rotation matrix and $\vec{d}_w$ is the location, in world coordinates, of the center of projection of the camera. The inverse of this mapping is simply

$$\vec{X}_w \;=\; R^T \vec{X}_c + \vec{d}_w. \tag{3}$$

The perspective transformation can now be applied to the 3D point $\vec{X}_c$ (i.e., in the camera's coordinates),

$$\vec{x}_c \;=\; \frac{f}{X_{3,c}}\vec{X}_c \;=\; \begin{pmatrix} x_{1,c} \\ x_{2,c} \\ f \end{pmatrix}. \tag{4}$$

Everything here is measured in meters (say), not pixels, and f is the camera's focal length.

Intrinsic Calibration Matrix

The intrinsic calibration matrix, M_{in}, transforms the 3D image position $\vec{x}_c$ (measured in meters, say) to pixel coordinates,

$$\vec{p} = \frac{1}{f} M_{in} \vec{x}_c, \tag{5}$$

where M_{in} is a 3×3 matrix. The factor of $1/f$ here is conventional.

For example, a camera with rectangular pixels of size $1/s_x$ by $1/s_y$, with focal length f, and piercing point (o_x, o_y) (i.e., the intersection of the optical axis with the image plane provided in pixel coordinates) has the intrinsic calibration matrix

$$M_{in} = \begin{pmatrix} f s_x & 0 & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

Note that, for a 3D point $\vec{x}_c$ on the image plane, the third coordinate of the pixel coordinate vector $\vec{p}$ is $p_3 = 1$. As we see next, this redundancy is useful.

Equations (1), (4) and (5) define the transformation from the world coordinates of a 3D point, $\vec{X}_w$, to the pixel coordinates of the image of that point, $\vec{p}$. The transformation is nonlinear, due to the scaling by $X_{3,c}$ in equation (4).

A Note on Units

So far we have written the focal length f in meters. But note that only the terms fs_x and fs_y appear in the intrinsic calibration matrix,

$$M_{in} = \begin{pmatrix} fs_x & 0 & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{pmatrix},$$

where $s_{x,y}$ are in the units of horizontal/vertical pixels per meter (and $o_{x,y}$ are in pixels).

Instead of meters, it is common to measure f in units of pixel width, that is, replace fs_x by f. In which case the intrinsic calibration matrix becomes

$$M_{in} = \begin{pmatrix} f & 0 & o_x \\ 0 & fa & o_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{7}$$

where $a = s_y/s_x$ is the (unitless) aspect ratio of a pixel ($0 < a < 1$ if the pixels are rectangular and flat, $a = 1$ if the pixels are square, and $a > 1$ rectangular and tall).

Homogeneous Coordinates

The projective transform becomes linear when written in the following homogeneous coordinates,

$$\begin{aligned} \vec{X}_w^h &= c(\vec{X}_w^T, 1)^T, \\ \vec{p}^h &= d\vec{p} = d\,(p_1, p_2, 1)^T. \end{aligned}$$

Here c, d are arbitrary nonzero constants . The last coordinate of these homogeneous vectors provide the scale factors. It is therefore easy to convert back and forth between the homogeneous forms and the standard forms.

The mapping from world to pixel coordinates can then be written as the *linear* transformation,

$$\vec{p}^h = M_{in} M_{ex} \vec{X}_w^h. \tag{8}$$

Essentially, the division operation in perspective projection is now implicit in the homogeneous vector $\vec{p}^h$. The division is simply postponed until $\vec{p}^h$ is rescaled by its third coordinate to form the pixel coordinate vector $\vec{p}$.

Due to its linearity, equation (8) is useful in many areas of computational vision.

Example: Lines Project to Lines

As a first application of the perspective projection equation (8), consider a line in 3D written in homogeneous coordinates, say

$$\vec{X}^{\,h}(s) \;=\; \begin{pmatrix} \vec{X}^{\,0} \\ 1 \end{pmatrix} + s \begin{pmatrix} \vec{t} \\ 0 \end{pmatrix}.$$

Here $\vec{X}^{\,0}$ is an arbitrary 3D point on the line expressed in world coordinates, $\vec{t}$ is a 3D vector tangent to the line, and s is the free parameter for points along the line. To avoid special cases, we assume that the line does not pass through the center of projection, and the tangent direction $\vec{t}$ has a positive inner-product with the optical axis (more on this below). By equation (8), the image the point of $\vec{X}^{\,h}(s)$ is

$$\vec{p}^{\,h}(s) \;=\; M\vec{X}^{\,h}(s) \;=\; \vec{p}^{\,h}(0) + s\vec{p}^{\,h}_t,$$

where $M = M_{in}M_{ex}$ is a 3×4 matrix, $\vec{p}^{\,h}(0) = M((\vec{X}^0)^T, 1)^T$, and $\vec{p}^{\,h}_t = M(\vec{t}^{\,T}, 0)^T$. Note $\vec{p}^{\,h}_t$ and $\vec{p}^{\,h}(0)$ are both constant vectors, independent of s. Therefore the image of the 3D line, in pixel coordinates, is

$$\vec{p}(s) \;\equiv\; \frac{1}{p^h_3(s)}\vec{p}^{\,h}(s) \;=\; \frac{1}{\alpha(s)}\vec{p}^{\,h}(0) + \frac{s}{\alpha(s)}\vec{p}^{\,h}_t, \qquad (9)$$

where $\alpha(s) = p^h_3(s)$. Using equations (1) and (7) we find

$$\alpha(s) = p^h_3(0) + \beta s, \quad \text{for } \beta = p^h_{t,3} = \vec{e}_3^{\,T} M_{ex}(\vec{t}^{\,T}, 0)^T, \qquad (10)$$

where $\vec{e}_3^{\,T} = (0, 0, 1)$. The condition that the inner-product of $\vec{t}$ and the direction of the optical axis is positive is equivalent to $\beta > 0$.

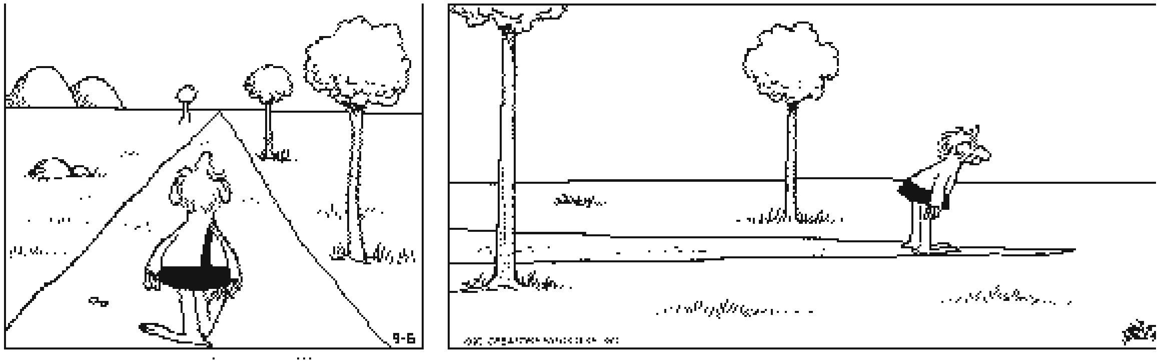
Note that equation (9) shows that $\vec{p}(s)$ is in the plane spanned by two constant 3D vectors. It is also in the image plane, $p_3 = 1$. Therefore it is in the intersection of these two planes, which is a line in the image. That is, lines in 3D are imaged as lines in 2D. (Although, in practice, some lenses introduce “radial distortion”, which we discuss later.)

One caveat on eqn (9) is that some of these points may be behind the principal plane (and therefore behind the camera). Using equations (1) and (7) it follows that $X_{c,3}(s)$, the Z-component of the point on the line written in camera coordinates, is equal to the third component $\vec{p}^{\,h}(s)$, which we denoted by $\alpha(s)$ above. Thus the point is in front of the principal plane if and only if $\alpha(s) > 0$ (and in front of the lens if $\alpha(s) > c$ for some constant $c > 0$.)

Since $\beta > 0$ we have from (10) that $1/\alpha(s) \to 0$ and $s/\alpha(s) \to 1/\beta$ as $s \to \infty$. Therefore, from (9), the image points $\vec{p}(s) \to (1/\beta)\vec{p}^{\,h}_t$ as $s \to \infty$. Note that this limit point is a constant image point dependent only on the tangent direction $\vec{t}$.

In fact, in homogeneous world coordinates, the 4D vector $(\vec{t}^{\,T}, 0)^T$ is the point at infinity in the direction $\vec{t}$. The perspective projection of this point is simply $\vec{p}^{\,h}_t = M(\vec{t}^{\,T}, 0)^T$, which is homogeneously equivalent to the limit of the image points we derived above. The next example explores this fact further.

Example: Parallel Lines Project to Intersecting Lines



Next consider a set of parallel lines in 3D, say

$$\vec{X}_k^h(s) \;=\; \begin{pmatrix} \vec{X}_k^0 \\ 1 \end{pmatrix} + s \begin{pmatrix} \vec{t} \\ 0 \end{pmatrix}.$$

Here all these lines have the same tangent direction $\vec{t}$, and hence are parallel in 3D (both in the world and camera coordinates).

To eliminate special cases, we again assume that none of these lines passes through the center of projection, and $\vec{t}$ has a positive inner-product with the direction of the optical axis (i.e., $\beta > 0$, with β defined as in equation (10)).

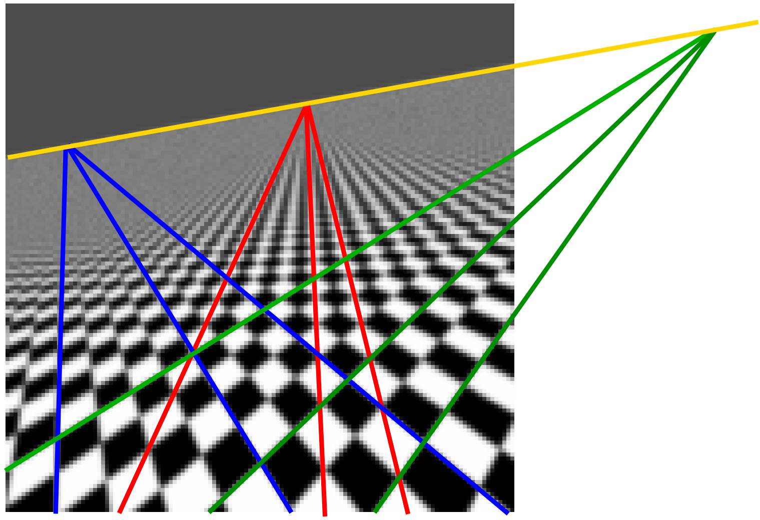
Then from the previous example we know that, as $s \rightarrow \infty$, the perspective projections of the points $\vec{X}_k^h(s)$ all converge to the same image point, namely $\vec{p}_t^h = M(\vec{t}^T, 0)^T$.

Thus the images of the parallel 3D lines $\vec{X}_k^h(s)$ all intersect at the image point $\vec{p}_t^h$. Moreover, it can be shown from equations (9) and (10) that, under the natural condition that we only form the image of points on the 3D line which are in front of the principal plane (i.e., $X_{c,3}(s) = \alpha(s) > 0$), the projected points on the image line segments converge *monotonically* to $\vec{p}_t^h$. That is, in the image, the projected line segments all appear to terminate at $\vec{p}_t^h$. (For example, note the sides of the road in the left figure above. Although, as the right figure shows, we can always be surprised.)

In summary, the common termination point for the images of parallel lines in 3D is the perspective projection of the 3D tangential direction $\vec{t}$. It is referred to as the *vanishing point*.

Example: The Horizon Line

As another exercise in projective geometry, we consider multiple sets of parallel lines, all of which are coplanar in 3D. We show that the images of each parallel set of lines intersect and terminate at a point on the horizon line in the image.



Consider multiple families of parallel lines in a plane, where each family of lines has the tangent direction $\vec{t}_j$ in 3D. From the previous analysis, the j^{th} family must co-intersect at the image point (in homogeneous coordinates)

$$\vec{p}_j^{\,h} = M(\vec{t}_j^{\,T}, 0)^T.$$

Since the tangent directions are all assumed to be coplanar in 3D, any two distinct directions provide a basis. That is, assuming the first two directions are linearly independent, we can write

$$\vec{t}_j = a_j \vec{t}_1 + b_j \vec{t}_2,$$

for some constants a_j and b_j. As a result, we have

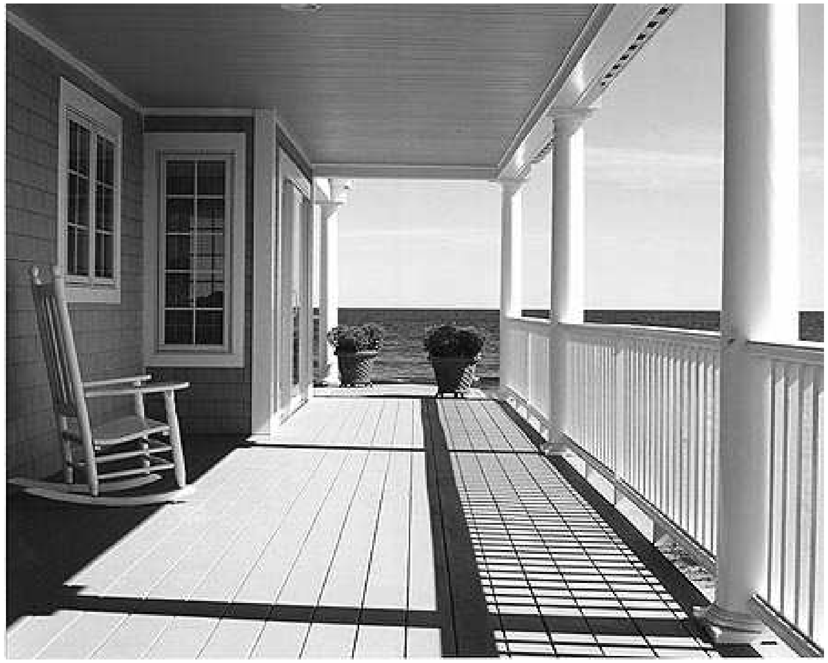
$$\vec{p}_j^{\,h} = M([a_j \vec{t}_1 + b_j \vec{t}_2]^T, 0)^T = a_j \vec{p}_1^{\,h} + b_j \vec{p}_2^{\,h}$$

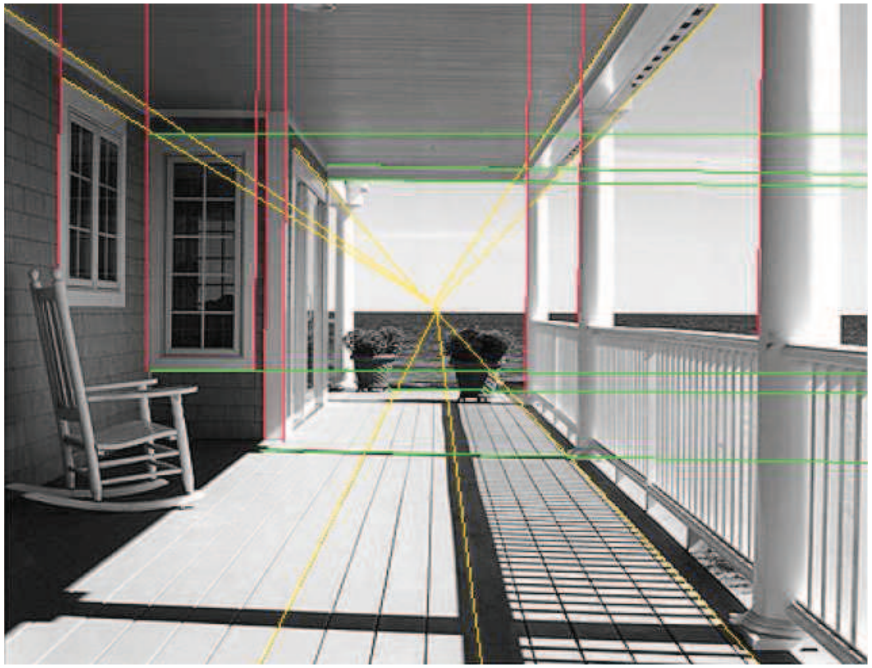
Dividing through by the third coordinate, $p_{j,3}^h$, we find the point of intersection of the j^{th} family of lines is at the image point

$$\vec{p}_j = \left(\frac{1}{p_{j,3}^h}\right) \vec{p}_j^{\,h} = \left(\frac{a_j p_{1,3}^h}{p_{j,3}^h}\right) \vec{p}_1 + \left(\frac{b_j p_{2,3}^h}{p_{j,3}^h}\right) \vec{p}_2 = \alpha_j \vec{p}_1 + \beta_j \vec{p}_2.$$

From this equation it follows that $\alpha_j + \beta_j = 1$. (Hint, look at the last row in this vector valued equation.) Hence the image point $\vec{p}_j$ is an affine combination of the two image points $\vec{p}_1$ and $\vec{p}_2$. Therefore the horizon must be the line in the image passing through $\vec{p}_1$ and $\vec{p}_2$, which is what we wanted to show.

Example: 3D Sets of Parallel Lines





Many man-made environments have a wealth of rectangular solids. The surface normals for the planes in these structures are restricted to just three orthogonal directions (ignoring signs). This means that there are three horizon lines, one for each surface normal.

It is also relatively common (with a good carpenter) to have 3D lines on these surfaces which have three mutually orthogonal tangent directions $\vec{t}_k$, $k = 1, 2, 3$. An example of such lines is shown on the right, with each family in a different colour. (But I suspect one of these sketched lines does not correspond to an edge in the scene with one of the three selected tangential directions, can you identify which one?)

Sketch the lines and the three vanishing points for the (corrected) sets of lines. You can select visible edges in the image to add further lines to these three sets. Also sketch the three horizon lines for the three sets of parallel planes. In both cases use a suitable notation for vanishing points and horizon lines that are far outside the image boundary.

It turns out that the resulting information is suitable for both determining the focal length of the camera (assuming square pixels) and reconstructing a scaled 3D model for the major planar surfaces of the porch. See single-view metrology, say Szeliski, Sec. 6.3.3.

Optical Distortion

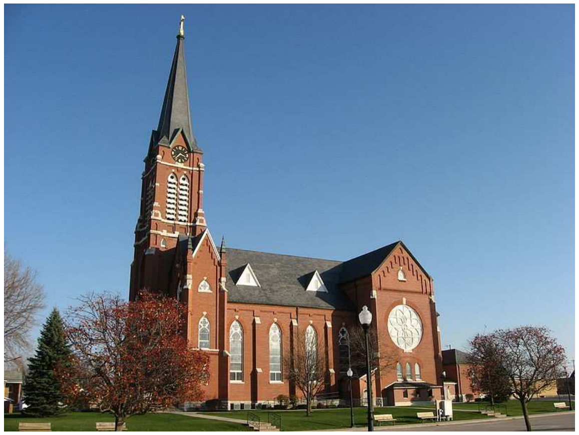
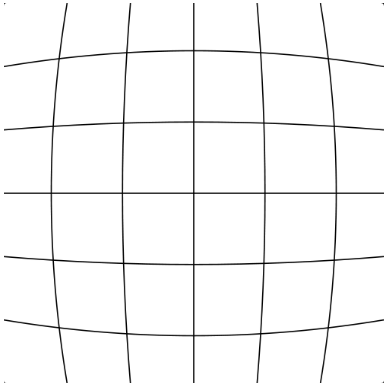
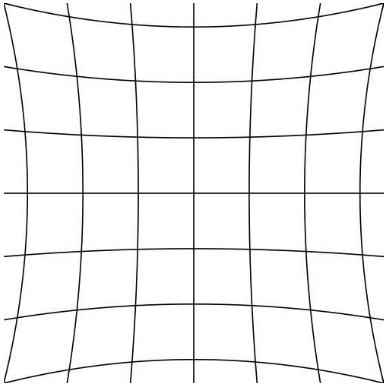


Image with barrel distortion.



Barrel distortion of square grid.



Pincushion distortion.
Images from Wikipedia.

Imagine printing an image on a thin rubber sheet. For many cameras, this image is a spatially distorted version of a perfect perspective transformation of the scene (e.g., top-left). This spatial distortion can be corrected by warping (i.e., applying a variable stretching and shrinking to) the rubber sheet.

This correction can be done algorithmically by first estimating a parametric warp from sample image data (perhaps simply one image containing many straight lines). Often a radial distortion suffices. The overall process is called calibrating the *radial distortion*. (See Wikipedia, Distortion (Optics).)

This warp can then be applied to any subsequent image acquired by that camera; effectively unwarping it to provide a new image which is a close approximation to perfect perspective projection.

Lenses

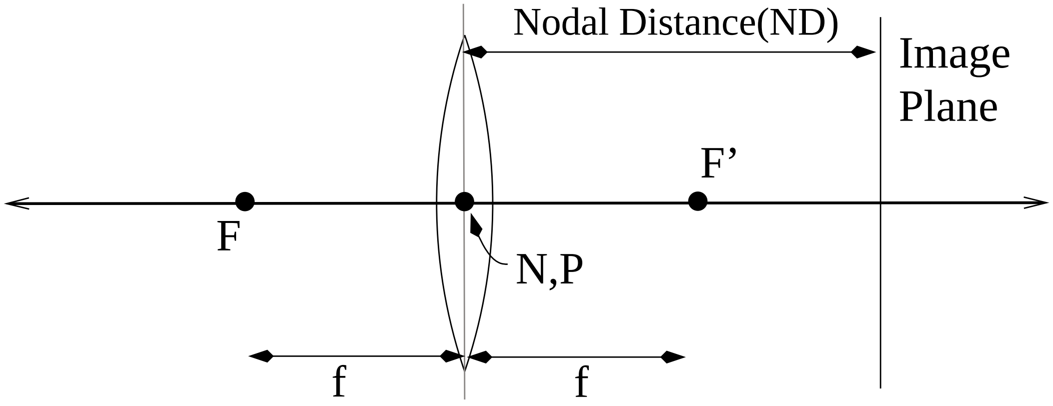
Finally we discuss a more detailed model of lenses, namely the thin lens model.

This model replaces the pinhole camera model, and is essential for:

- relating the optical properties of a lens, such as its focal length, to the parameter f (that we also called "focal length") in the pinhole camera model,
- characterizing the defocus of an image as a function of the depth of an object,
- understanding the critical optical blur which is performed before the image is sampled.

Thin Lens: Cardinal Points

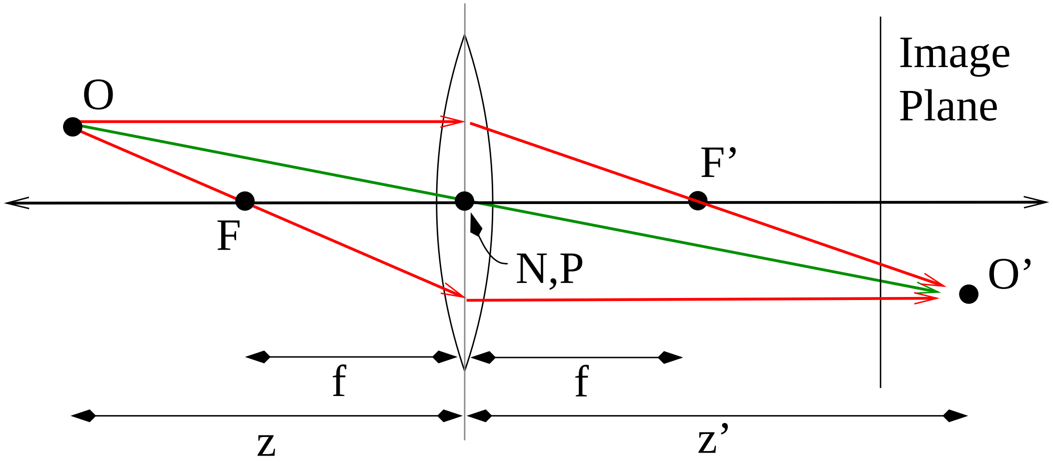
The thin lens model provides a more general model for a camera's lens than a simple pinhole camera. It allows defocus to be modelled.



- A cylindrically symmetric lens can be geometrically modelled by three pairs of *cardinal points* on the optical axis, namely the *focal*, *nodal*, and *principal points*.
- Here we consider a thin lens, with the same material (such as air) on either side of the lens.
- For this case, the nodal and principal points all agree (denoted, N,P above), and are often called the *center of projection*.
- The plane perpendicular to the optical axis containing P is called the *principal plane*.
- The *focal points* F and F' are a distance f away from N. Here f is called the *focal length* of the lens.

Thin Lens: Principal Rays

The cardinal points provide a geometric way to determine where a world point, $\vec{\mathcal{O}}$, will be focussed.



The point $\vec{\mathcal{O}}$ is focussed at $\vec{\mathcal{O}}'$ given by the intersection of (any two of the) three *principal rays*:

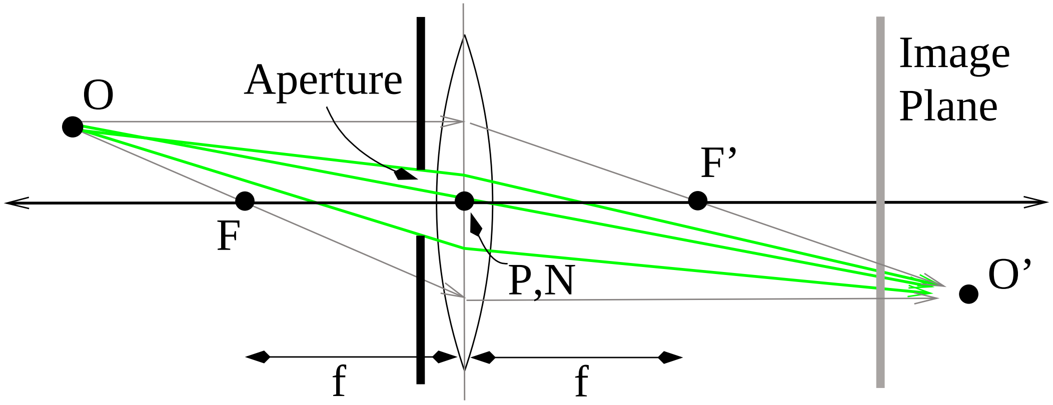
- A ray from $\vec{\mathcal{O}}$ passing straight through the nodal point N of the lens.
- The two rays that are parallel to the optical axis on one side of the principal plane, and pass through the front or rear focal points (F and F') on the opposite side of the lens.

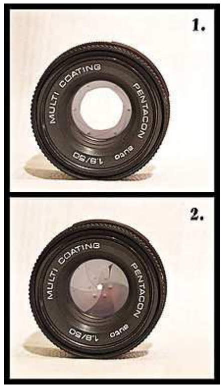
All rays from $\vec{\mathcal{O}}$ which pass through the lens are focussed at $\vec{\mathcal{O}}'$ (behind the image plane shown above).

The *lens equation* $\frac{1}{f} = \frac{1}{z} + \frac{1}{z'}$ follows from this construction, where are z and z' be the distances of $\vec{\mathcal{O}}$ and $\vec{\mathcal{O}}'$ to the principal plane.

Thin Lens: Aperture and F-number

A lens aperture can be modelled using an occluder placed within the principal plane.





From Wikipedia.

The aperture itself is the hole in this occluder. Let D denote the *aperture diameter*.

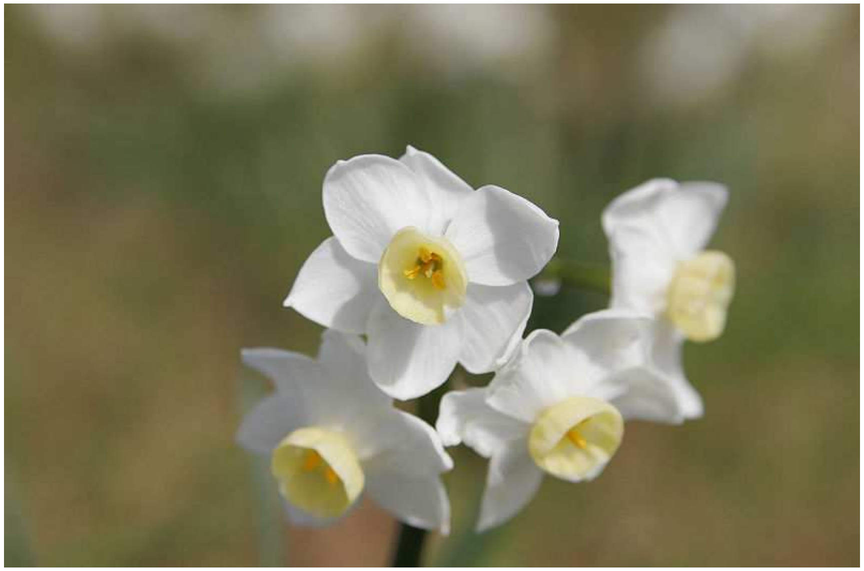
The f-number (or f-stop) of a lens is given by the ratio f/D.

For the defocussed situation shown above, the point source $\mathcal{O}$ is imaged to a small region in the image plane (i.e., the projection of the aperture plus an additional blur region due to diffraction effects). The size of this projected region is proportional to D, and therefore inversely proportional to the f-number.

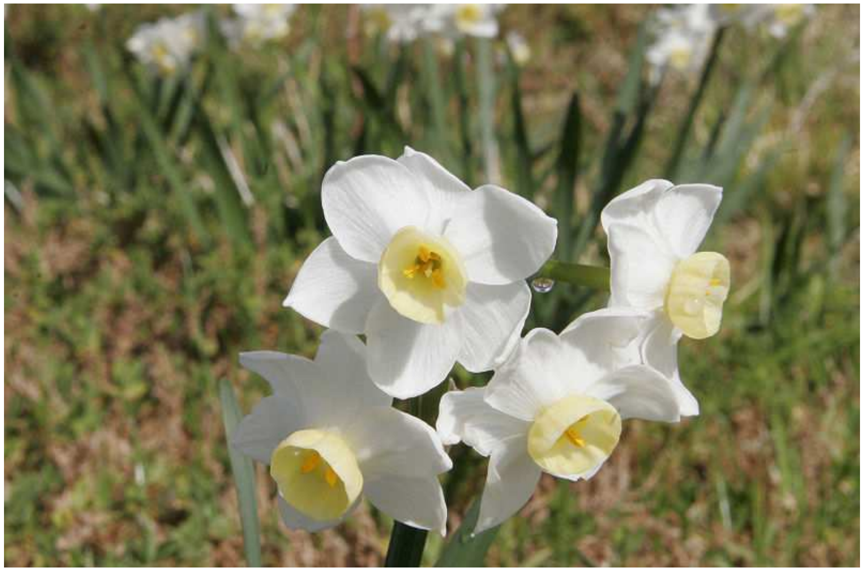
As the f-number increases (i.e., D decreases), the lens behaves more like a pinhole camera, although, due to diffraction the blur radius never decreases to zero.

Thin Lens: Depth of Field

The depth of field is the distance between the nearest and furthest objects in the scene that appear acceptably in focus. That is, they are blurred by no more than a small fixed diameter.



F-number: 5 (i.e., “f/5” on the lens)

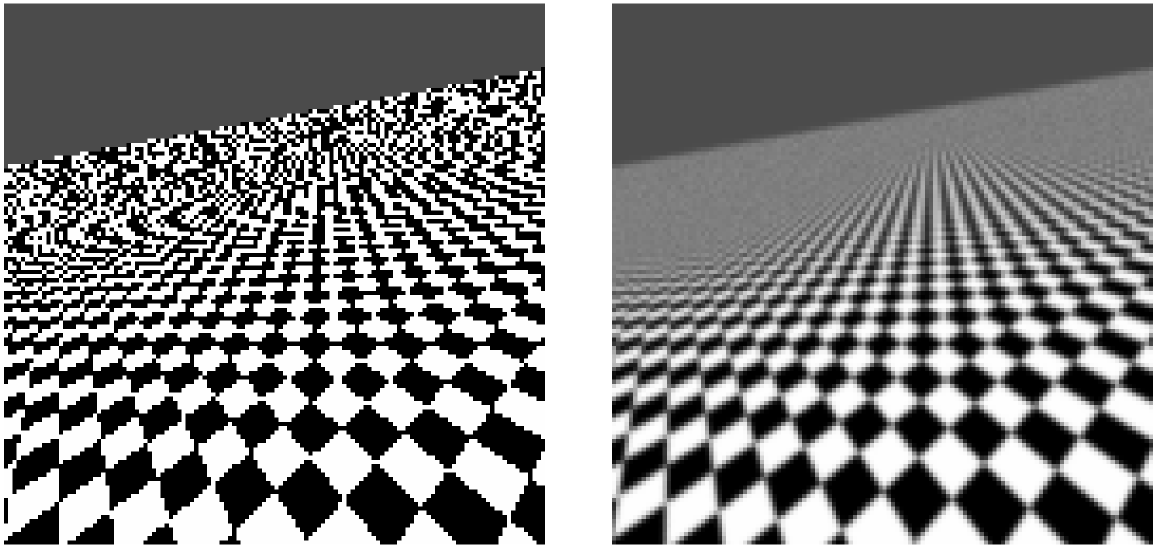


F-number: 32 (i.e., “f/32”)

Since the size of the blurred region is inversely proportional to the f-number, a larger f-number provides a larger depth of field. This is illustrated by the image pair above (from Wikipedia, depth of field).

Optical Blur, Sensor Elements, and Aliasing

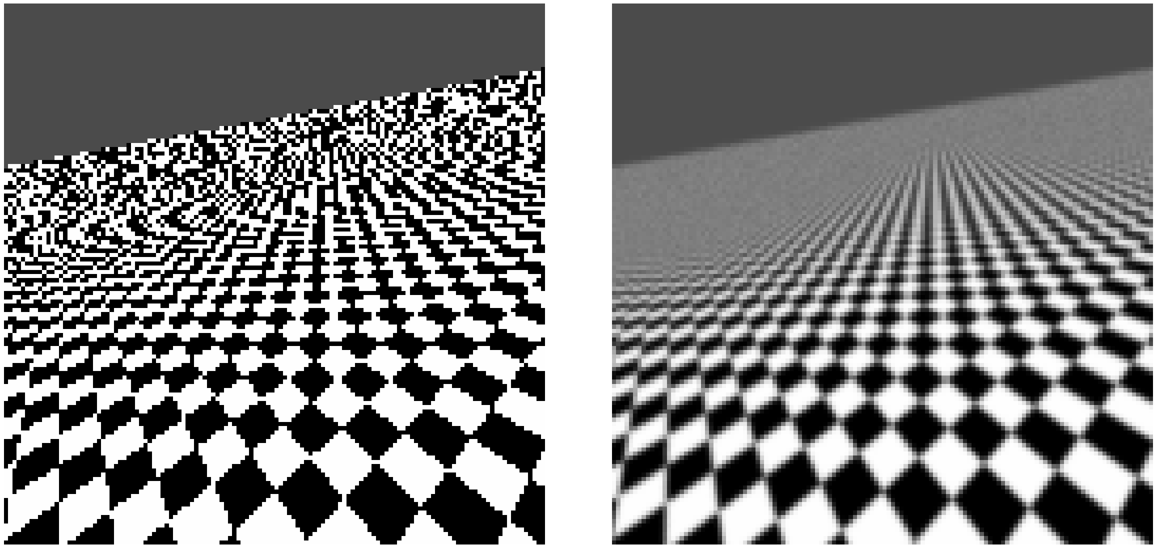
Due to diffraction effects and the physical area of the light sensing elements (e.g., individual CCD sensors), the incident light sensed by any camera has been spatially averaged over a small region in the image plane. This (analogue) averaging plays a critical role in image formation.



A perspective image of an infinite checkerboard is rendered by a pinhole camera model (above left). Due to the point sampling, the checks in the distance appear distorted. This is called "aliasing". Given a more appropriate model for the analogue optical blur this aliasing is eliminated (above right).

Resampling and Aliasing

Downsampling an image refers to reducing the number of pixels. E.g., downsampling by 2 uses every second pixel in every second row. (This is also called decimation.) Before downsampling, care must be taken that aliasing isn't introduced in the downsampled image.



Resampling Rule of Thumb. One can safely resample an image by K, in each direction x and y, only if the original image is smooth enough that, any point in the original image can be approximated (say using bilinear interpolation) given only the 4 nearest downnsampled neighbours.

Otherwise the image should first be blurred (next lecture), then downsampled.

Other Issues in Image Projection and Formation

Intrinsic Calibration refers to a procedure to estimate the intrinsic parameters to the camera, namely the parameters of the intrinsic calibration matrix M_{in} (as, say, given in equation (7)), along with the radial distortion parameters for the camera.

Extrinsic Calibration refers to estimating the extrinsic calibration matrix M_{ext}, with respect to some predetermined world coordinate frame. (For both types of calibration, see the Camera Calibration Toolbox for Matlab, by Jean-Yves Bouguet.)

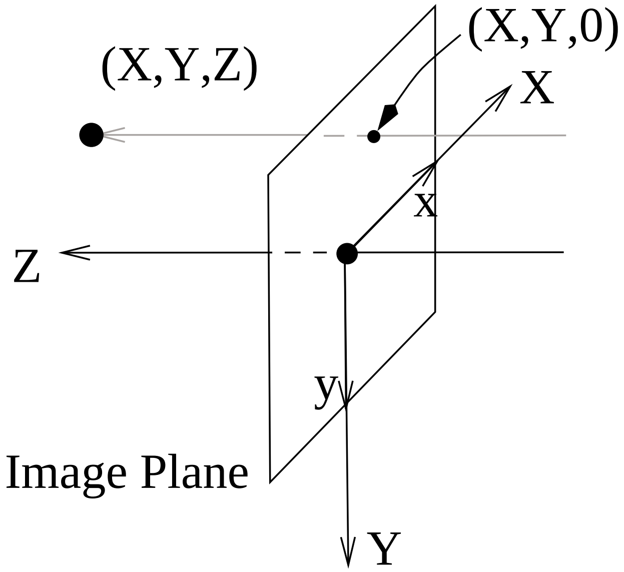
Radiometry, Reflection and Colour. In order to synthesize an image we also require some understanding of the measurement of light (i.e., radiometry), and reflectance (i.e., the interaction of light with surfaces). See the additional readings on the course homepage for more information. Here we will largely ignore these topics since firstly, we have enough on our plate already, and secondly, these topics overlap with other courses (i.e., CSC320 and CSC418).

Digital Image Formation. A good overview is in Szeliski, Sec. 2.3.

Image Noise arises from most of the steps of digital image formation. In this course we will restrict ourselves to simple noise models. Noise will be a constant companion from here on.

Aside: Orthographic Projection

Scaled orthographic projection provides a linear approximation to perspective projection, which is applicable for a small object far from the viewer and close to the optical axis.



Given a 3D point (X, Y, Z), the corresponding image location under scaled orthographic projection is

$$\begin{pmatrix} x \\ y \end{pmatrix} = s_0 \begin{pmatrix} X \\ Y \end{pmatrix}$$

Here s_0 is a constant scale factor; orthographic projection uses $s_0 = 1$.

There are several other alternative approximations to perspective projection.