CHAPTER 2

Input Devices

1. Video Input Devices

A Computer Vision system senses its environment through a video camera and a digitizer, and like its biological counterpart, the camera converts an optical image to a format that can be processed by a computing device. But unlike the human eye, the typical video camera is a low resolution, noise infected, mechanically fragile and of unproven durability device. While there are cameras that do better than the human eye in one or another aspect, overall the human eye is far superior. There are of course cameras smaller than the eye and cameras more accurate than the eye. But the quality of the images of the small cameras is poor and the size of the high quality cameras is monstrous. On the other hand a video camera can be easily connected through a digitizer to a computer, not to mention that they will not rot if forgotten in a drawer.

1.1. Components of video cameras

A camera is a quite complex device and different people would analyze it in many different ways. A physicist would concentrate on the light refraction, diffraction etc, an optician on the lens technology, an electrical engineer on the sensing device and the signal amplification and a photographer on the aesthetics. A Computer Scientist has three components to study. The control system, the sensing surface geometry and some of the properties of the lens.

1.1.1. Camera control

The control system modifies the parameters of the camera under computer (or manual) command. Most camera parameters can be modified, but focal length, *f*-number, and position and orientation are the most common and most important. The estimation and the strategies for changing these parameters are the subject matter of *Camera Calibration* and *Active Vision*.

1.1.2. Sensing surface

The sensing device is almost invariably a CCD (*Charge Coupled Device*), a monolithic chip that contains all the sensing elements that form the image surface. Typical size for video quality CCD is 1/2" (12.5 mm) diagonal and a resolution around 640×480. Since television standards require a $3:4$ aspect ratio the size of our typical sensing surface is

$$12.50mm \times \frac{4}{\sqrt{3^2+4^2}} = 10.00mm$$

and

$$12.50mm \times \frac{3}{\sqrt{3^2+4^2}} = 7.50mm$$

This $10.00 \times 7.50mm^2$ area contains about 640×480 pixels, each about $15.00 \times 15.00\mu m^2$ in size. These are about the thickness of a human hair and are hardly visible with naked eye. Still they are much bigger than the wavelength (about half a μm) of the visible light

Almost all video cameras have around 480 rows of pixels because this is what the video standard dictates. Very few cameras though have all the 640 pixels per row. The video standard was designed for tubes and scanning beams of electrons and consequently is vague on this detail. Moreover, the image is sensed as analog signal before being digitized. In this analog format the image is transmitted row by row (with some synchronization signals between rows) and each row is an analog waveform with no indication as to where the original pixels were. The only indication about the number of pixels per row is the amount of detail present. So the number of pixels per row depends only on the digitizer that usually samples it at 640 pixels per row to make digitized image compatible with most computer screens that have square pixels (640:480 is a ratio 4:3).

The size of the CCD affects a few things. The price is quite dramatically afftexted because we not only need more silicon for the chip but also a bigger lens and enclosure to go with it. But a bigger chip allows for better light collection ability and thus less camera noise (less graininess). It also allows the pixels to be larger than the *Airy circle* that we discuss in the next subsection.

1.1.3. Camera lenses

There is a large variety of lenses in the market with specifications to fit many applications in research, industry, education, entertainment etc. And of course the number of parameters that specify the quality of a lens is large. The most important of them in Computer Vision are the following two:

1.1.3.1. Focal length The image of an object that is infinitely far away (or at least very far away like the sun) will form at a distance equal to the focal length behind the lens. The longer focal length the higher the “magnifying” power of the lens. The trade names of the lenses according to their magnifying power are *fish eye*, *wide angle*, *normal*, *standard* and *tele* and the term *macro* is used for lenses that can focus on objects that are very close. For half inch video cameras $16mm$ is the standard lens, the one with the most natural feel.

1.1.3.2. f-number The *f*-number is defined as the ratio of the focal length to the diameter of the lens. The smaller the *f*-number the wider the opening of the lens and the more light can come through. So with low *f*-number the image is brighter. There two more things that change with the *f*-number and have to do with perfect lenses. One is the depth

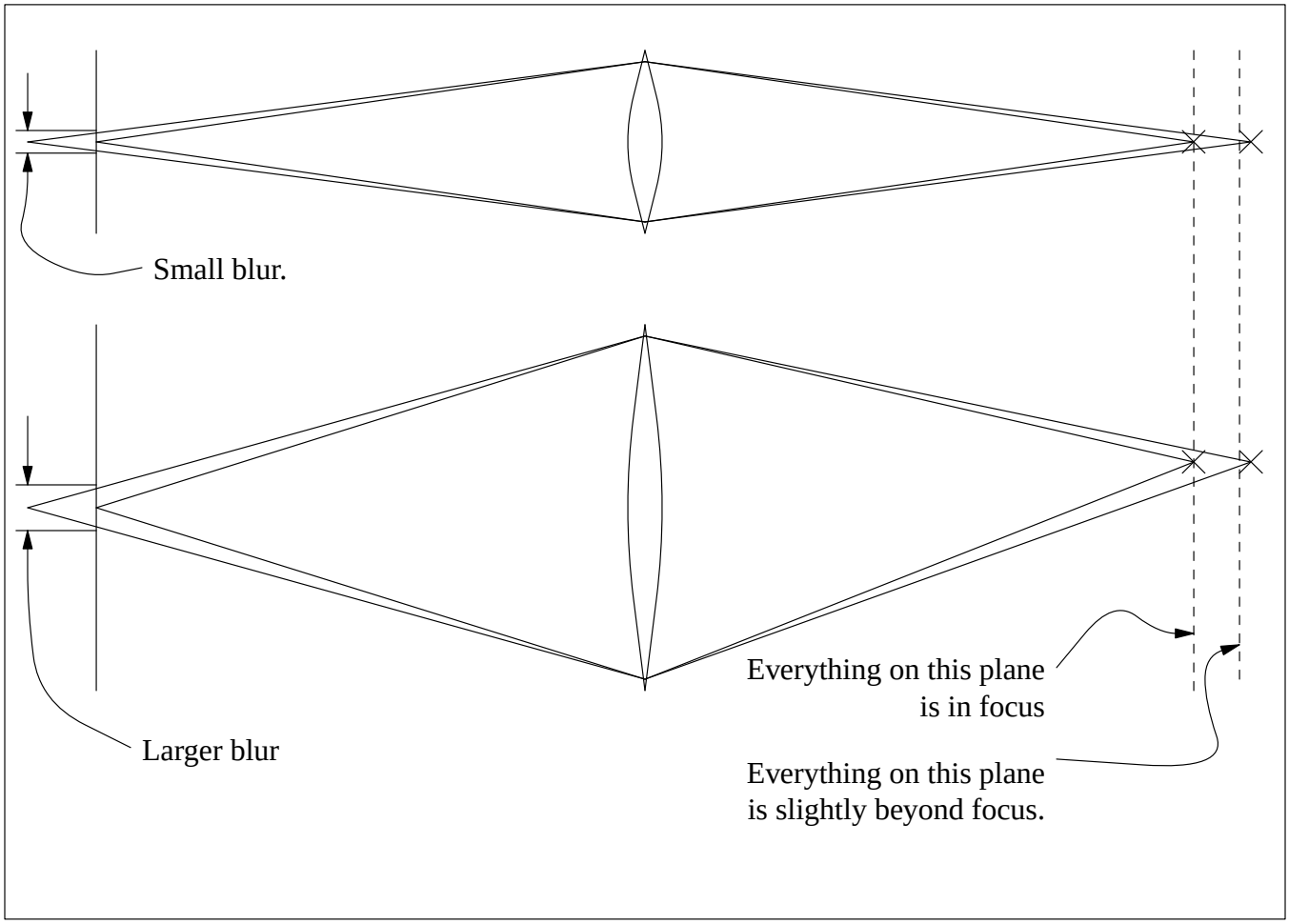

Figure 1.1: Things that are in focus will give rise to sharp images in both lens cameras above. But light sources that are a bit further away will blurry. The bigger the lens the larger the blur and so the camera is more sensitive to movements along the optical axis. The depth of field of the camera with the wide lens is more restricted.

of field. When we focus the lens at an object, say 4 meters away the objects that are a little closer and a little further away are still acceptably focused. If things are focused half a meter closer and half a meter further than the object then the depth of field is one meter. The larger the *f*-number the larger the depth of field and so the lens is more forgiving to inaccuracies in focusing (Fig. 1.1). But there is another effect which is due to diffraction that conspires to create the opposite effect. When the aperture of the lens becomes too small then the diffraction of the light tends to blur the image. The diameter of the blur, called the *Airy circle*, is in the order of the *f*-number times the wavelength.

To understand this we have to see how the lens really works and go beyond the common idea that the lens “bends” the light rays. This is just a convenient abstraction that works most of the time but not always. Light in this context behaves like a wave and the

lens just introduces such phase shifts in the different rays that they interfere with each other in the desired way. If the aim is to focus light on a single point the rays that go to this point will arrive there in phase and interfere constructively and rays that go to all other points will arrive out of phase and interefere destructively. The problem is that a tiny distance away from our intended focus the light rays, while not perfectly in phase, will not be totally out of phase either, allowing some light to reach there and instead of the desired single infinitesimal point we will get a blur.

We briefly examine the phenomenon by looking at a small number of rays. If they are to interefere constructively they have to follow paths that are of equal length and if they are to interefere destructively paths that differ by half a wavelength. Looking at (Fig. 1.2) we can see that the difference between the two paths of the off center rays should be $\lambda/2$ and assuming that distance x is small compared to the diameter and that the f-number is relatively large, this difference is

$$\sqrt{f^2+(d/2+x)^2}-\sqrt{f^2+(d/2-x)^2}\approx\frac{\left(f^2+(d/2+x)^2\right)-\left(f^2+(d/2-x)^2\right)}{2\sqrt{f^2+d/2^2}}\approx$$

$$\frac{x}{\sqrt{\left(\frac{f}{d}\right)^2+\left(\frac{1}{2}\right)^2}}\approx\frac{x}{f\text{-number.}}$$

It is easy to see that the distance $2x = \lambda f$-number and that the diameter of this blur is approximately f-number times the wavelength. Although is a “back of the envelop” calculation the result is fairly accurate. The minimum f-number of most lenses is between 1.2 and 4.8 but the f-number can become about 16, which makes the worst case Airy circle comparable in size with the pixel which is around $15\mu m$.

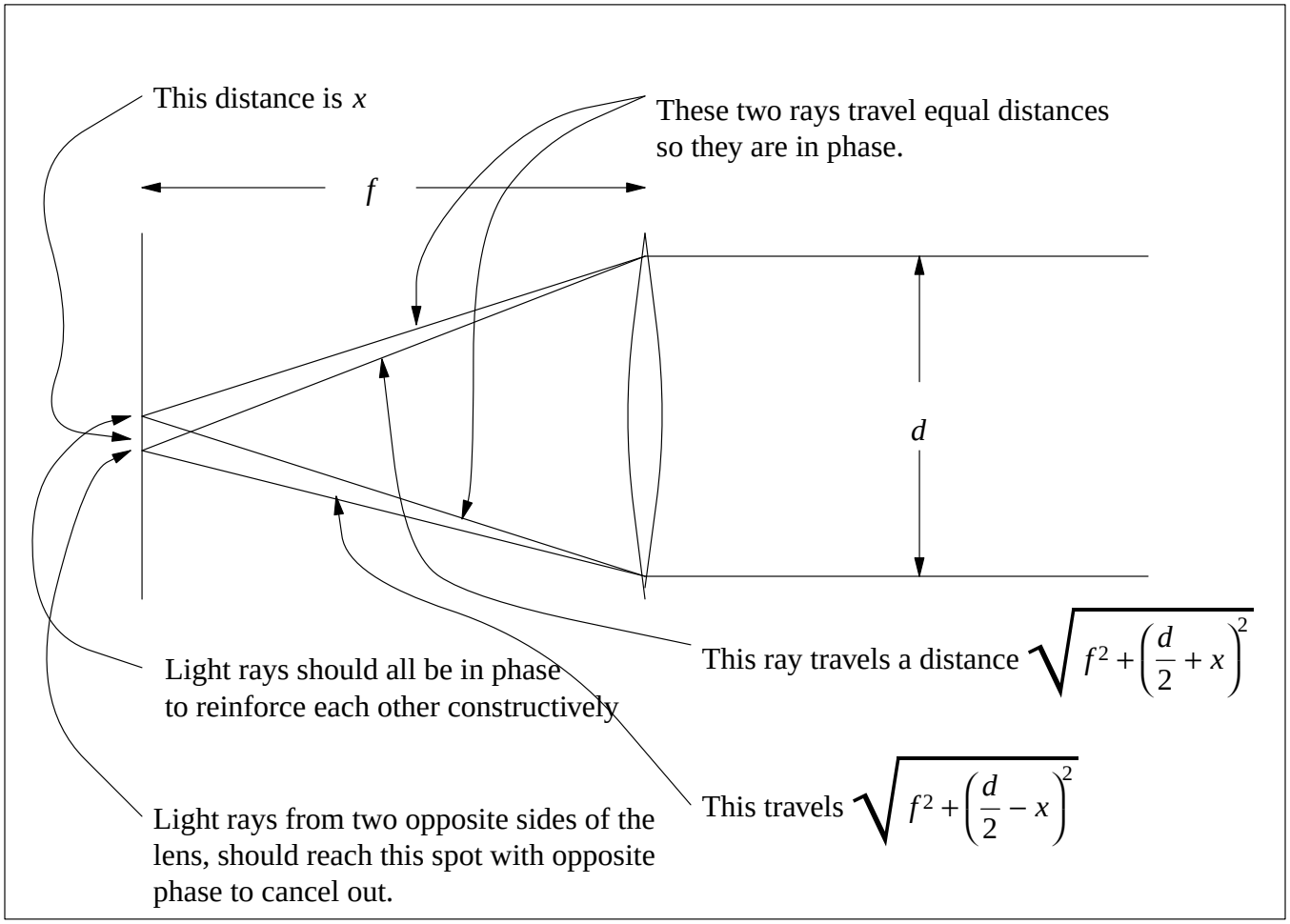

Figure 1.2: A distant point will appear a bright spot because light waves meet in phase and reinforce each other. Right next to it light rays will meet out of phase and cancel out. The minimum distance that this cancellation can happen is the radius of the Airy circle.

1.1.3.3. Aberrations and distortions

Although the focal length and the *f*-number are the most commonly cited numbers lens parameters, there are several others that give information about the quality of the optics and how much they differ from the ideal or *ray-tracing* model. In general we distinguish two kinds of such deviation of a lens from the ideal: the aberrations and the distortions. The first refers to the deviations that result in loss of focus e.g. the image becomes blurred and the later to the loss of geometric fidelity, where the light from a single point focuses on a single point but this point is the wrong point and as a result straight lines are not projected to straight lines. In general the larger the diameter of the lens the more noticeable the aberrations and the shorter the focal length the more noticeable both the distortions and aberrations. We can easily deduce that the lower the *f*-number (e.g. the brighter the lens) the greater the distortions and aberrations.

Lens designers solve the lens myopia in a way very similar to the way doctors solve it for humans: glasses. These glasses take the form of *compound lens*. The design of these lenses is a very difficult optimization problem that does not concern us. Most quality lenses have rather small aberrations

The most common aberrations and distortions are:

1.1.3.3.1. Spherical Most lenses have their surfaces ground to the shape of a patch of a sphere, because this shape is the easiest to construct with simple mechanical means. The main consequence of this is that the rays of light that go through the central region of the lens focus at the nominal focal distance, whereas the ones that pass through the peripheral areas of the lens focus a bit closer. This is a relatively easily corrected aberration.

1.1.3.3.2. Coma The coma aberration makes the image of a small bright point near the edges of the image, look like a bright dot with a comet like halo. It happens because the light rays that go through the center of the lens focus at the ray tracing spot whereas the ones that enter at oblique angles and pass through the periphery of the lens focus at a different distance from the center of the image than the ideal.

1.1.3.3.3. Astigmatism Lens astigmatism shares few things with human astigmatism and should not be confused. The astigmatism of a lens is negligible near the center of the image and increases towards the edges. It is due to the tendency of the rays that emanate from a single 3-D point to focus on two tiny perpendicular linear segments one in front of the other. If we point the lens towards a single tiny distant light source and put the image plane close to the lens the image will be unfocused and look like a round blur. As we move it away the image of the light source becomes clearer. At some point instead of shrinking to a single point it will shrink to a small line. If we continue moving the image plane it will become a round blur again and a little further away it will again focus for the second and last time on a small line perpendicular to the first. The one of the lines will be along the radius and the other normal to that.

1.1.3.3.4. Color Aberration The different wavelengths of light are deflected with different angles when they pass through an optical system like a lens. As a result light of different colors focus in different places reducing the quality of the image. This aberration can be fairly easily eliminated with proper design.

1.1.3.3.5. Radial Distortion The radial distortion does not blur the image by itself but distorts the geometry. The result is that if we point the camera towards a rectangular grid the image will not consist of straight lines but curved. If we have a negative radial distortion the image will be like an inflated balloon and if have a positive it will be like a pincushion. The distortion in most cases can be approximated by $\delta r = k_1 r^3$ where k_1 is the lens distortion factor, r is the radius of a point on the image and δr is the displacement of the point along the radius. We can write it also in vector form (with bold characters denoting vectors) $\delta \mathbf{r} = k_1 r^2 \mathbf{r}$. For higher accuracy one can take more terms of the polynomial e.g. $\delta \mathbf{r} = \mathbf{r} \cdot \sum_i k_i r^{2i}$. The fish eye lens has an extremely pronounced radial distortion and all the wide angle lenses have a quite strong such distortion. The commercial lenses incorporate very few corrections to this distortion for two reasons. One reason is that it is rather expensive and inconvenient because more optical elements are required, arranged in ways that would increase weight and size. Second, it is not that disagreeable to a human when viewed on an already curved television screen. Unfortunately it affects any computer vision application that requires a geometrically accurate image. And while all aberrations can be reduced by decreasing the diameter of the lens, radial distortion can not. Typical values for the radial distortion vary from less than one pixel for high quality tele lenses to several pixels for wide angle lenses.

1.1.3.4. Calculations with the lenses

From the definition of the focal length we know that the image plane should be one focal length away from the lens to properly form the image of an object at infinity. If the object is at distance α in front of the lens and the image plane is at distance β behind, then

$$\frac{1}{\alpha} + \frac{1}{\beta} = \frac{1}{f} \tag{1.1}$$

where f is the focal length.

If the lens is slightly defocused by $\delta\alpha$ then according to figure (Fig. 1.1) and the rule of similar triangles the blur b will be

$$b = \delta\alpha \cdot \frac{d}{\alpha + \delta\alpha} \approx \delta\alpha \cdot \frac{d}{\alpha}$$

and using Eq. (1.1)

$$b = \delta\alpha \cdot \frac{d(\beta - f)}{f\beta} = \delta\alpha \cdot \frac{\beta - f}{\beta f\text{-number}}$$

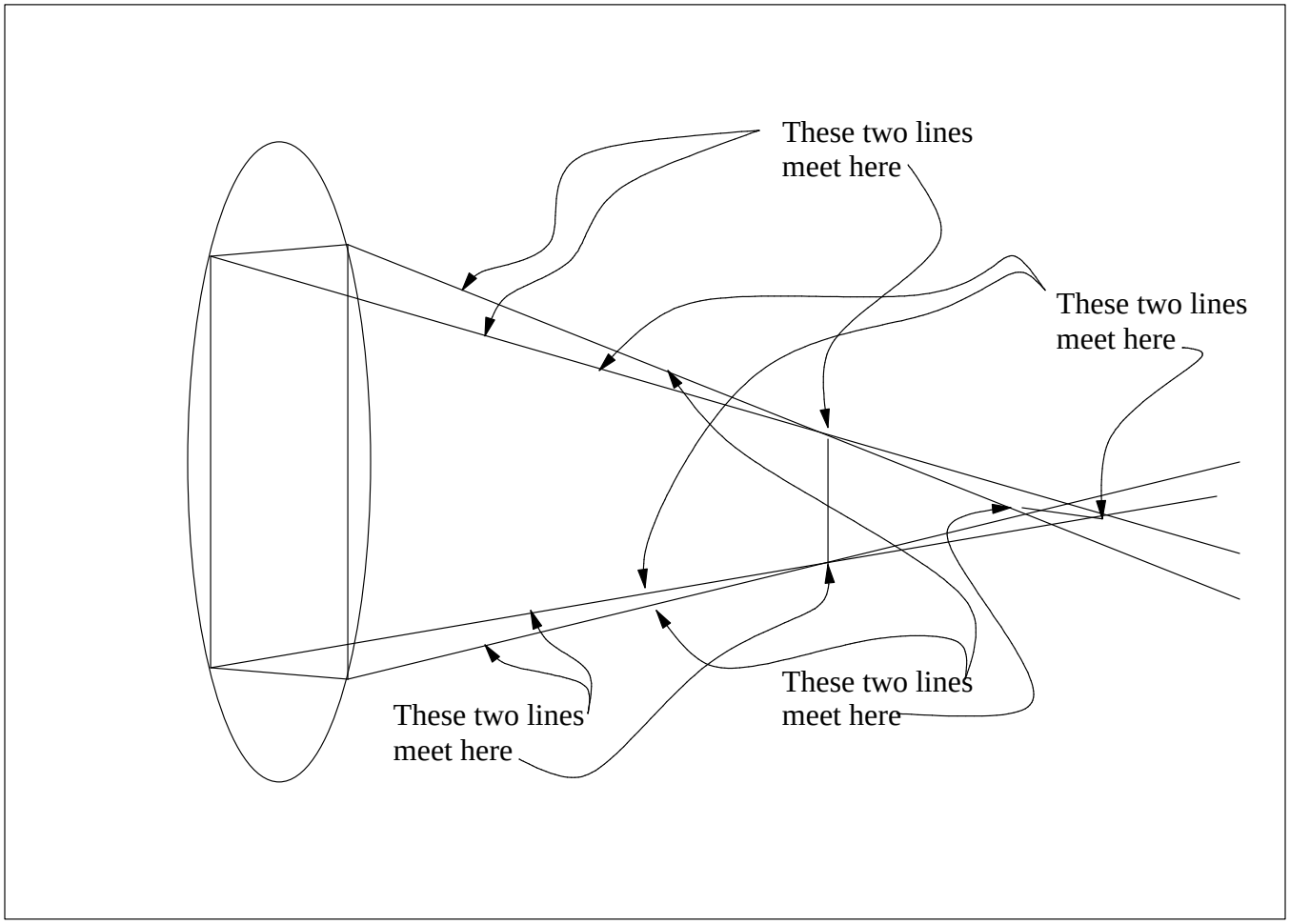

Figure 1.3: Light rays from a point source going through a lens with astigmatism will not focus on a single point but on two little lines one in the tangential and one in the radial direction that are different distances away from the lens.

and since $\beta \gg f$

$$b = \frac{\delta\alpha}{f\text{-number}}$$

Depending on the resolution of our camera (the combined effect of the number of pixels on the sensing surface and the aberrations of the lenses) there is a maximum $b_{\max}$ beyond which the effects of lack of focus will be visible. Given this $b_{\max}$ the depth of focus $\delta\beta$ is

$$|\delta\beta| = \frac{\beta^2}{\alpha^2}\,\delta\alpha = \frac{\beta^2}{\alpha^2}\,b_{\max}\,f\text{-number}$$

where we used the derivative of (Eq. (1.1)) to obtain $\delta\beta$.

1.2. Modeling the Video Camera Geometry

A real camera is an approximation of the pinhole camera which cannot be used due the low light gathering ability and the diffraction effects of the small aperture. Ideally we would prefer to do our calculations with a pinhole camera with an effective focal length of one unit. Since it is impossible to use it without a lens the next best thing is to model the camera and and hide all the details behind a few subroutines, C++ objects, or MediaMath objects. By replacing the camera with an abstract model we can throw in a few other nice features that real cameras lack, like non inverted image and coordinates that are easier to use.

The image is naturally a two dimensional quantity, but we will use three dimensional homogeneous coordinates for the greater versatility they offer. Let's define the coordinate systems we will use and then see how we transform one to the other.

The *Image Coordinate System* is the system we use for images irrespective of the 3-D nature of the imaged objects. This system is used exclusively in image processing and in any vision application that does not involve the 3-D world. The origin or point $[0, 0]$ of the system is the top left corner of the image and x and y increase to the right and down respectively and are usually integers (Fig. 1.4). The third coordinate, which is there merely for the convenience of the homogeneous coordinates, happens to be parallel to the axis of the camera with direction away from the viewer.

The second coordinate system we need is the *Camera Coordinate System*. This system can represent both the image points and 3-D points and it is attached to the camera. Its origin is the nodal point (the center) of the lens, its X and Y axes are parellel to the focal plane and the Z axis points towards the scene. The third (Z) coordinate of an image

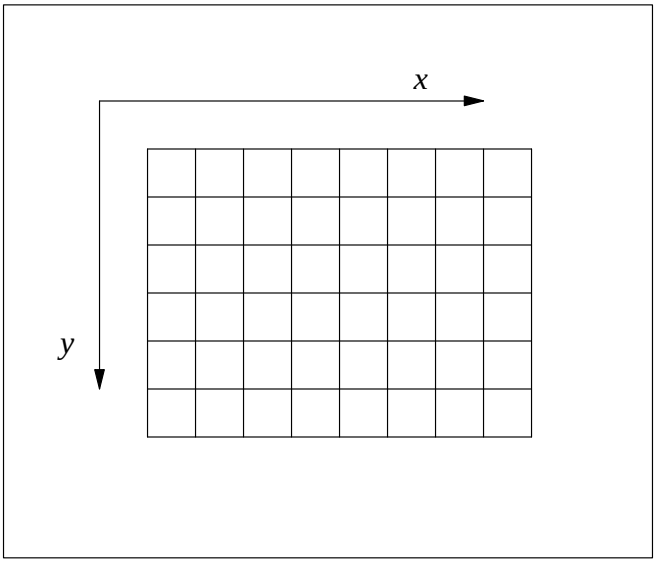

Figure 1.4: The image coordinate system.

point point can be considered either as the homogeneous coordinates supplement or the genuine Z component of a point on the focal plane. If we assume that the focal plane is one unit away from the origin then the two roles are indistinguishable.

The coordinate system is right handed with the focal plane normal to Z. Notice that the right handiness requires us to have the Y axis pointing down, which agrees with the conventions of the image coordinate system. The image is formed by a ray emanating from a point on the 3-D object towards the nodal point (Fig. 1.5). In a real camera we would have to extend this ray beyond the nodal point and get an inverted image behind the camera. Instead we choose to model the focal plane as being in front of the lens, so the image of the point is the intersection between the ray and the focal plane. This is quite different from both regular cameras with a lens or a reflective camera since the image is not inverted but the camera manufacturers are part of the conspiracy and cross wire the electronics. The image that comes out of the camera is more consistent with our unrealistic model than the actual physical model.

In most cases when we work with vectors the coordinate system is easily implied but when it is not we will use a left superscript I or c to specify image or camera coordinate system. Assume that we have a point ${}^{I}p$ defined in the image coordinate system

$$ {}^{I}p = \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} $$

and we want to find its relation with ${}^{c}p$ in the camera coordinate system

$$ {}^{c}p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} $$

In the absence of distortion, the relation involves only a rescaling and a shift. Since transformations involving rescalings and shifts are very common, scientists have given them a name. They call them *affine transformations*. We can find the rescaling factor by comparing the size of the pixel in the two coordinate systems. In the image coordinate system it is 1×1 and in the camera coordinate system it is $\frac{l_v}{r_v} \times \frac{l_h}{r_h}$, where l_h and l_v are the horizontal and vertical dimensions of the sensing surface and r_h and r_v are the corresponding resolutions (usually 640 and 480). We can assume that the focal length f is $f = 1$ and if it is not we can divide all length quantities by f. We also know that the shift is half the sensing surface size in each direction because the $[0, 0]$ point in the image system is in the upper left corner and in the camera system is in the center. So the matrix that relates them is:

$$C = \begin{bmatrix} \frac{r_h}{l_h} & 0 & \frac{r_h}{2} \\ 0 & \frac{r_v}{l_v} & \frac{r_v}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

and its inverse

$$C^{-1} = \begin{bmatrix} \frac{l_h}{r_h} & 0 & -\frac{l_h}{2} \\ 0 & \frac{l_v}{r_v} & -\frac{l_v}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Matrix C is called *Calibration Matrix* and here is why. While the camera and lens manufacturers are supposed to provide us with all the numbers like l_h and r_h to a high degree of precision, they do not always do. This is especially true for the focal length.

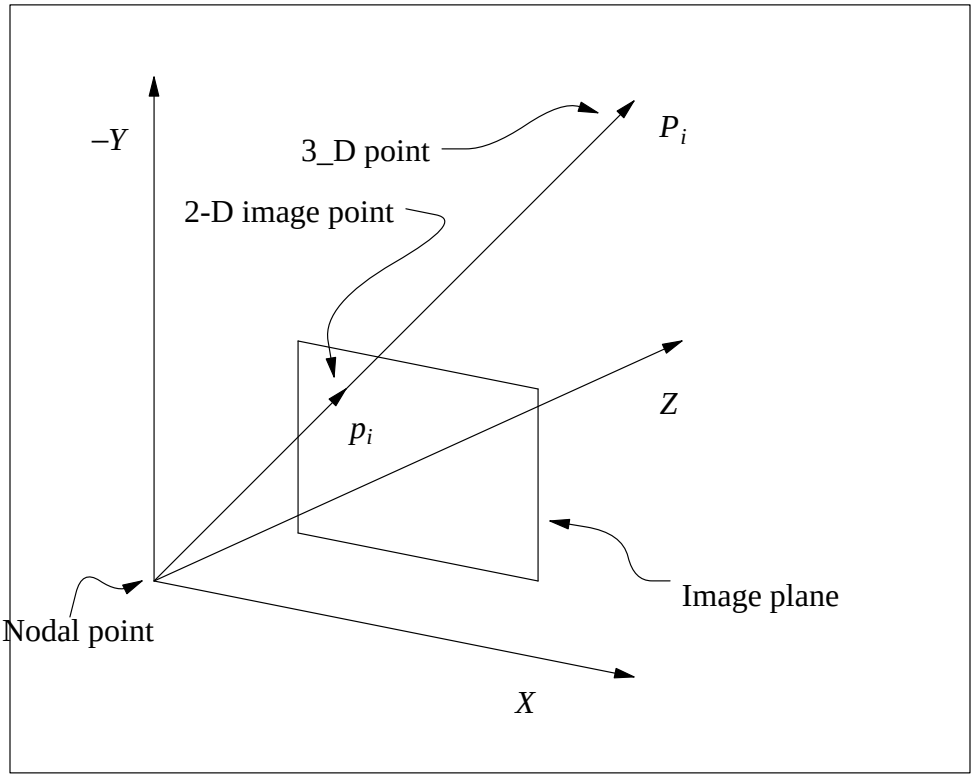

Figure 1.5: The camera coordinate system. By convention we use lower case letters for the image points and upper case letters for the 3-D points.

They also do not guarantee that the lens is directly in front of the center of the CCD. As a result we have to apply a procedure called *Camera Calibration* to calculate the parameters of C. Hence the name.

Also note that we used the convention for the camera coordinate system that X points to the right and Y points down. This is similar to the image coordinate system. We might as well have the X point to the left and Y point up. In this case the $[1,1]$ and $[2,2]$ elements of the calibration matrix would have the opposite sign.

1.2.1. Perspective Projection

Since the projection involves both the 3-D world points and their images we use the camera coordinate system. By convention we use capital letters to represent 3-D points and lower case to represent their images. If a 3-D point is

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

and its projection is

$$p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

then by simple application of the law of similar triangles we have

$$x = \frac{X}{Z}$$
$$y = \frac{Y}{Z}$$

always assuming that the focal length is 1. If we want to write the above equation in vector notation then

$$p = \frac{P}{P \cdot \hat{Z}}$$

where $\hat{Z}$ is the unit vector in Z direction and $\cdot$ is the dot product.

1.2.2. MediaMath Example

When we have these parameters it is easy to construct the inverse calibration matrix and use it as a global variable.

```
Cal_Mati = mk_fmat(3,3,[[l_h/r_h,0,-l_h/2.0]
                        [0,l_v/r_v,-l_v/2.0]
                        [0,0,f_length]]);
Cal_Mati /= f_length;
```

Then every vector in image space can be converted to camera space easily.

```
i_p = mk_fvec(3,[j,i,1]);
```

```
c_p = Cal_Mati*i_p;
```

Consider for instance the the problem where we are already given the zed map *Z_map*, e.g. an image that at every pixel contains the depth or the Z component of the object at the corresponding point in 3-D, and we are asked to rotate all the points in the scene by the Euler angles α, β and γ and move them by a, b and c.

```
/* Construct the rotation and translation vector */
R = R_z(alpha) * R_y(beta) * R_z(gamma);
T = mk_fvec(3,[a, b, c]);
/* Construct the object vector as a generalized vector */
i_p = mk_gvec(3,[x_img(Z_map->vmax,Z_map->hmax),
                 y_img(Z_map->vmax,Z_map->hmax),
                 1]);
c_p = Cal_Mati*i_p;
c_p *= Z_map;
/* Rotate and translate */
new_c_p = R*c_p + T;
```

where instead of manipulating each point inside a double for-loop we consider them as a vector of images.