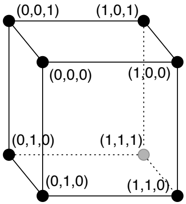
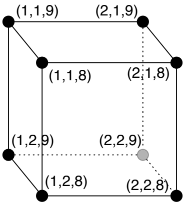
Perspective rendering

As application of change of basis, we show how to synthesize a camera view from a set of points in three dimensions, taking into account perspective.
The math will be useful in next lab, where we will go in the opposite direction, removing perspective from a real image.

We start with the points making up a wire cube:



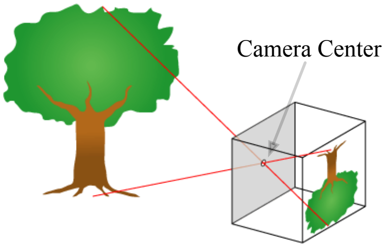
For reasons that will become apparent, we translate the cube, adding $(1, 1, 8)$ to each point.



How does a camera (or an eye) see these points?

Simplified camera model

Simplified model of a camera:

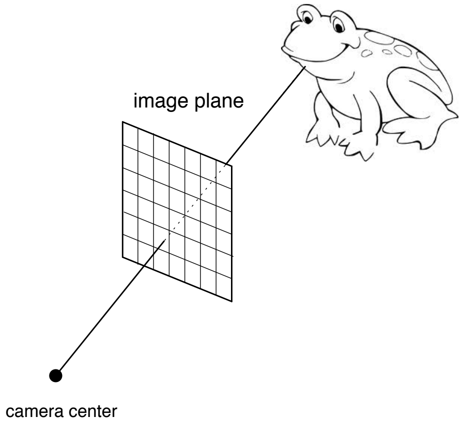


- There is a point called the *camera center*.
- There is an image sensor array in the back of the camera.
- Photons bounce off objects in the scene and travel through the camera center to the image sensor array.
- A photon from the scene only reaches the image sensor array if it travels in a straight line through the camera center.
- The image ends up being inverted.

Even more simplified camera model

Even simpler model to avoid the inversion:

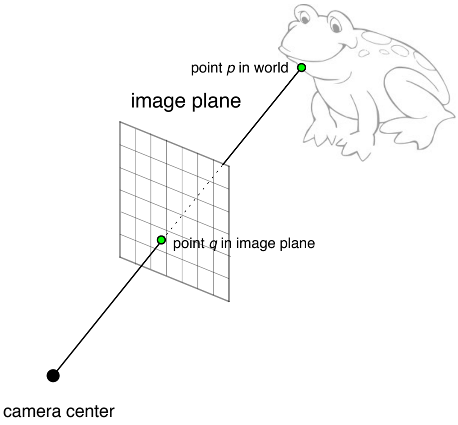
- The image sensor array is between the camera center and the scene.
- The image sensor array is located in a plane, called the *image plane*.
- A photon from the scene is detected by the sensor array only if it is traveling in a straight line towards the camera center.
- The sensor element that detects the photon is the one intersected by this line.
- Need a function that maps from point $\mathbf{p}$ in world to corresponding point $\mathbf{q}$ in image plane



Even more simplified camera model

Even simpler model to avoid the inversion:

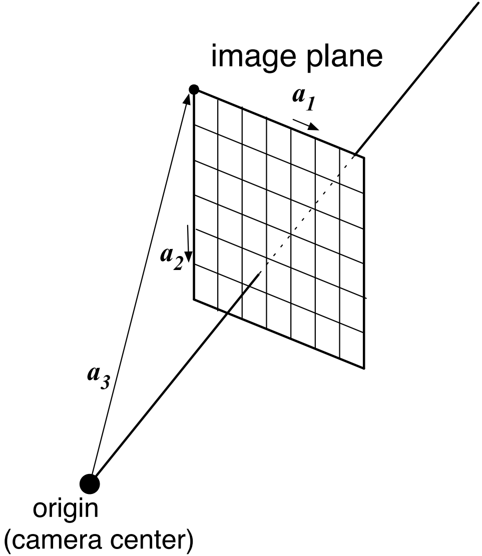
- The image sensor array is between the camera center and the scene.
- The image sensor array is located in a plane, called the *image plane*.
- A photon from the scene is detected by the sensor array only if it is traveling in a straight line towards the camera center.
- The sensor element that detects the photon is the one intersected by this line.
- Need a function that maps from point **p** in world to corresponding point **q** in image plane



Camera coordinate system

Camera-oriented basis helps in mapping from world points to image-plane points:

- The origin is defined to be the camera center. (That's why we translated the wire-frame cube.)
- The first vector $\mathbf{a}_1$ goes horizontally from the top-left corner of a sensor element to the top-right corner.
- The second vector $\mathbf{a}_2$ goes vertically from the top-left corner of a sensor element to the bottom-left corner.
- The third vector $\mathbf{a}_3$ goes from the origin (the camera center) to the top-left corner of sensor element (0,0).



From world point to camera-plane point

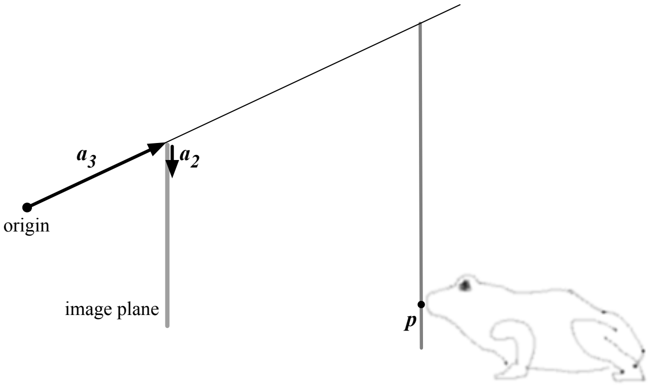
Side view (we see only the edge of the image plane)

- Have a point **p** in the world
- Express it in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$
- Consider corresponding point **q** in image plane.
- Similar triangles $\Rightarrow$ coordinates of **q**

Summary: Given coordinate representation (x_1, x_2, x_3) in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$,

coordinate representation of corresponding point in image plane is $(x_1/x_3, x_2/x_3, x_3/x_3)$.

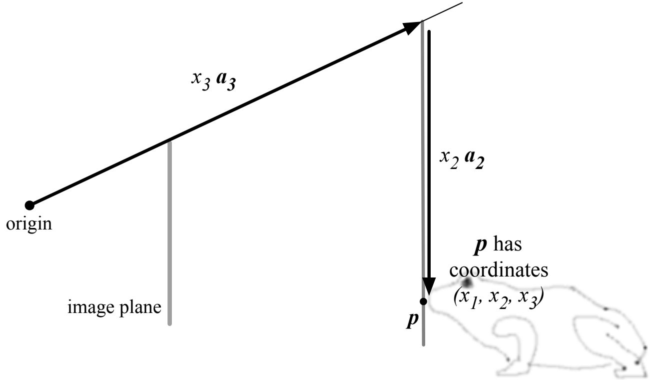
I call this *scaling down*.



From world point to camera-plane point

Side view (we see only the edge of the image plane)

- Have a point $\mathbf{p}$ in the world
- Express it in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$
- Consider corresponding point $\mathbf{q}$ in image plane.
- Similar triangles $\Rightarrow$ coordinates of $\mathbf{q}$



Summary: Given coordinate representation (x_1, x_2, x_3) in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$,

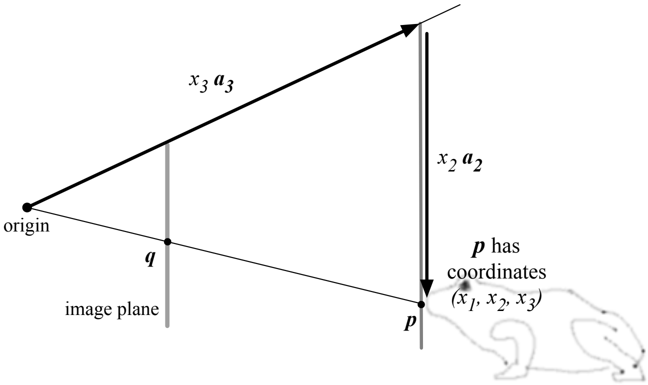
coordinate representation of corresponding point in image plane is $(x_1/x_3, x_2/x_3, x_3/x_3)$.

I call this *scaling down*.

From world point to camera-plane point

Side view (we see only the edge of the image plane)

- Have a point $\mathbf{p}$ in the world
- Express it in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$
- Consider corresponding point $\mathbf{q}$ in image plane.
- Similar triangles $\Rightarrow$ coordinates of $\mathbf{q}$



Summary: Given coordinate representation (x_1, x_2, x_3) in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$,

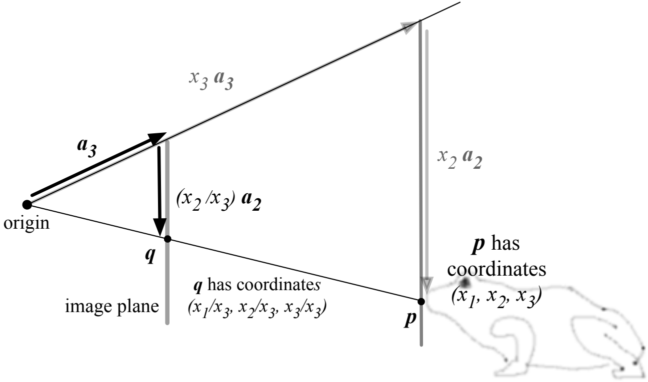
coordinate representation of corresponding point in image plane is $(x_1/x_3, x_2/x_3, x_3/x_3)$.

I call this *scaling down*.

From world point to camera-plane point

Side view (we see only the edge of the image plane)

- Have a point $\mathbf{p}$ in the world
- Express it in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$
- Consider corresponding point $\mathbf{q}$ in image plane.
- Similar triangles $\Rightarrow$ coordinates of $\mathbf{q}$



Summary: Given coordinate representation (x_1, x_2, x_3) in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$,

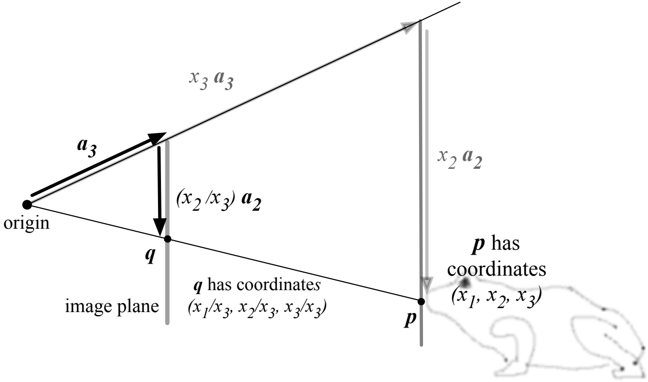
coordinate representation of corresponding point in image plane is $(x_1/x_3, x_2/x_3, x_3/x_3)$.

I call this *scaling down*.

From world point to camera-plane point

Side view (we see only the edge of the image plane)

- Have a point $\mathbf{p}$ in the world
- Express it in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$
- Consider corresponding point $\mathbf{q}$ in image plane.
- Similar triangles $\Rightarrow$ coordinates of $\mathbf{q}$



Summary: Given coordinate representation (x_1, x_2, x_3) in terms of $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$,

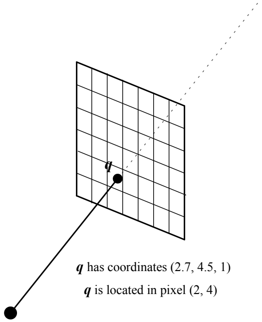
coordinate representation of corresponding point in image plane is $(x_1/x_3, x_2/x_3, x_3/x_3)$.

I call this *scaling down*.

Converting to pixel coordinates

Converting from a point (x_1, x_2, x_3) in the image plane to pixel coordinates

- Drop third entry x_3 (it is always equal to 1)



From world coordinates to camera coordinates to pixel coordinates

Write basis vectors of camera coordinate system using world coordinates

For each point $\mathbf{p}$ in the wire-frame cube,

- find representation in $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$
- scale down to get corresponding point in image plane
- convert to pixel coordinates by dropping third entry x_3