

# Lecture 10: Camera

DHBW, Computer Graphics

Lovro Bosnar

1.3.2023.

# Syllabus

- 3D scene
    - Object
    - Light
    - Camera
  - Rendering
  - Image and display
- Camera
    - Real camera system
    - Image formation
    - Pinhole camera model
    - Virtual pinhole camera
- 

# Where is camera?



<https://sciencebehindpixar.org/pipeline/sets-and-cameras>

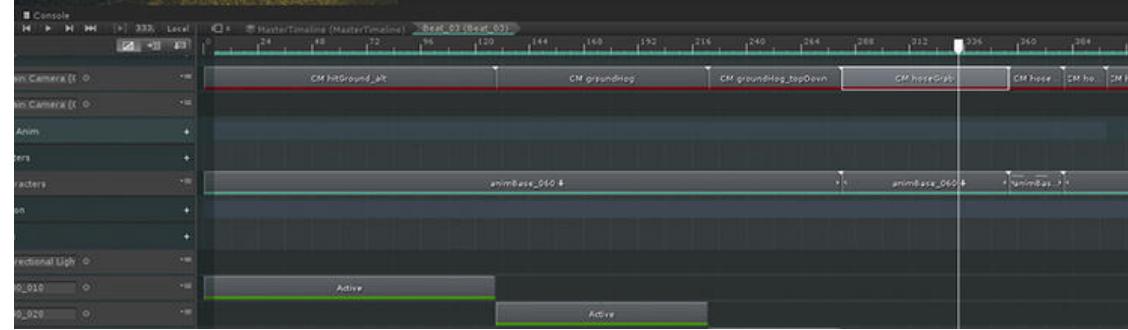
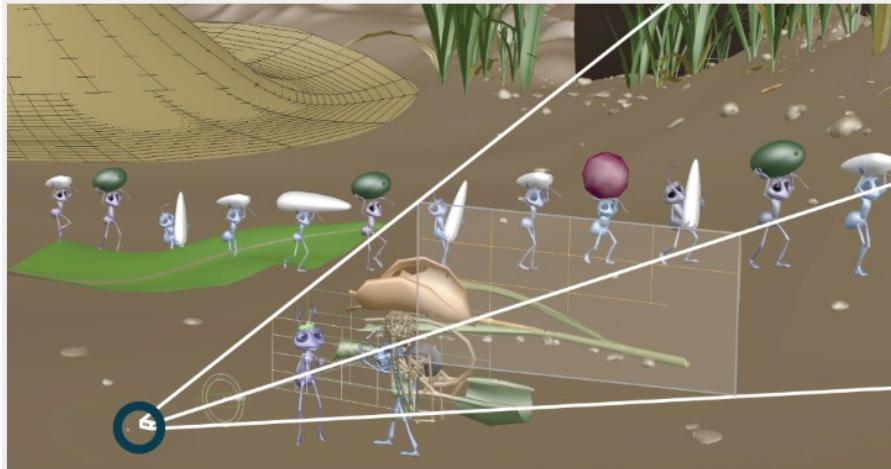


<https://www.half-life.com/en/alyx/>



<https://unity.com/solutions/film-animation-cinematics>

# Where is camera?



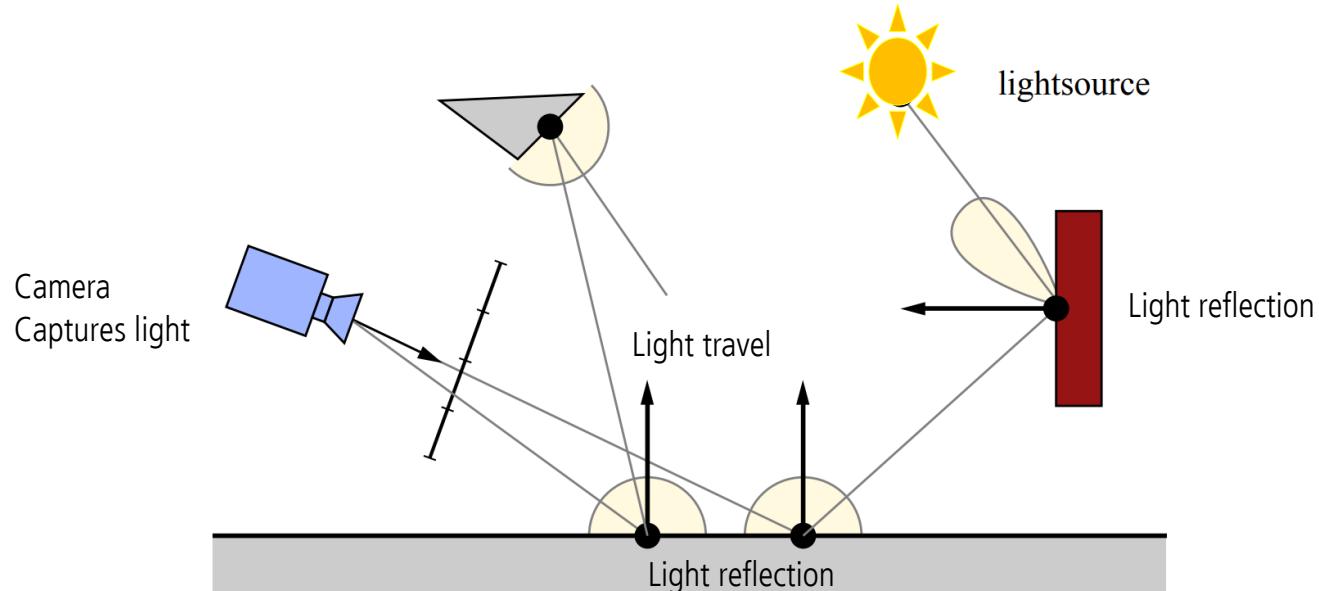
# Where is camera?

- In rendered images, camera is not seen – it defines what we see.



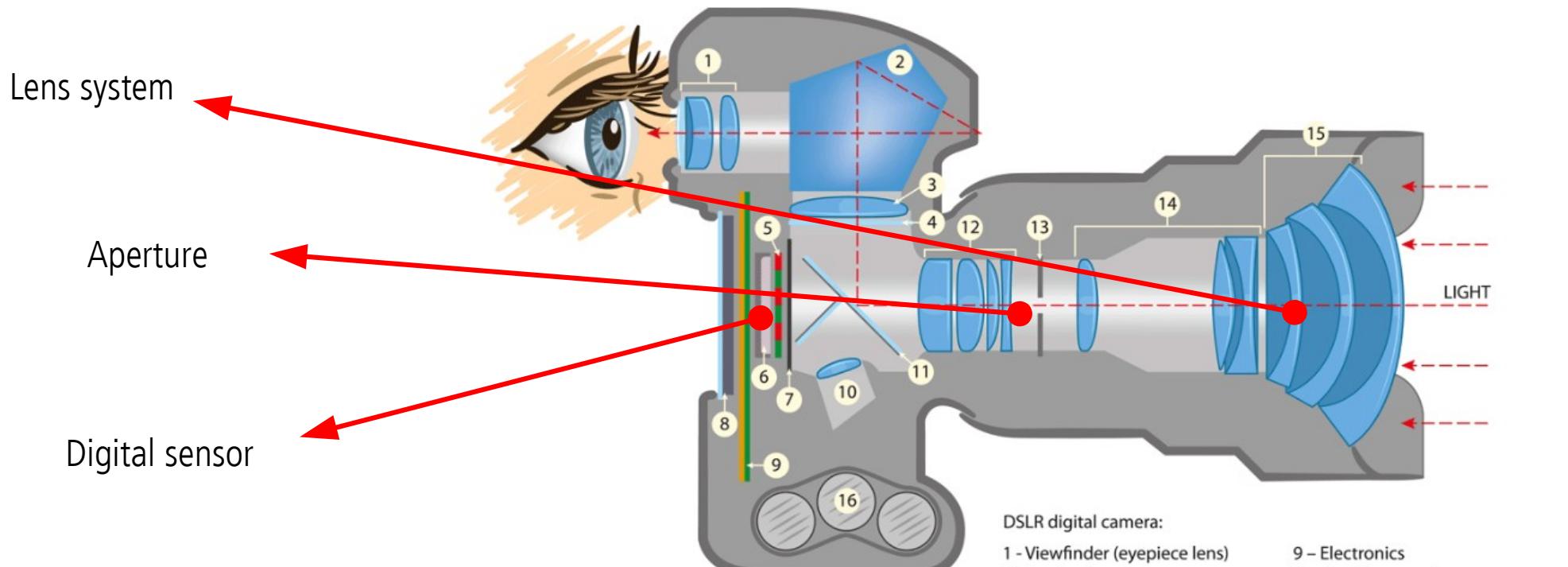
# Camera in rendering

- Camera defines a portion of visible scene
- Light falling on camera sensor forms an image
- To find light contributing to pixel colors, rays are traced from camera to 3D scene
  - Backward tracing



# Real camera systems

# Real camera systems



DSLR digital camera:

- |                                |                                     |
|--------------------------------|-------------------------------------|
| 1 - Viewfinder (eyepiece lens) | 9 - Electronics                     |
| 2 - Pentaprism                 | 10 - Autofocus system               |
| 3 - Focusing screen            | 11 - Reflex and relay mirror        |
| 4 - Condenser lens             | 12 - Focusing elements              |
| 5 - Color and infrared filter  | 13 - Aperture                       |
| 6 - Digital sensor             | 14 - Zoom elements                  |
| 7 - Shutter                    | 15 - Front light gathering elements |
| 8 - Display                    | 16 - Batteries                      |

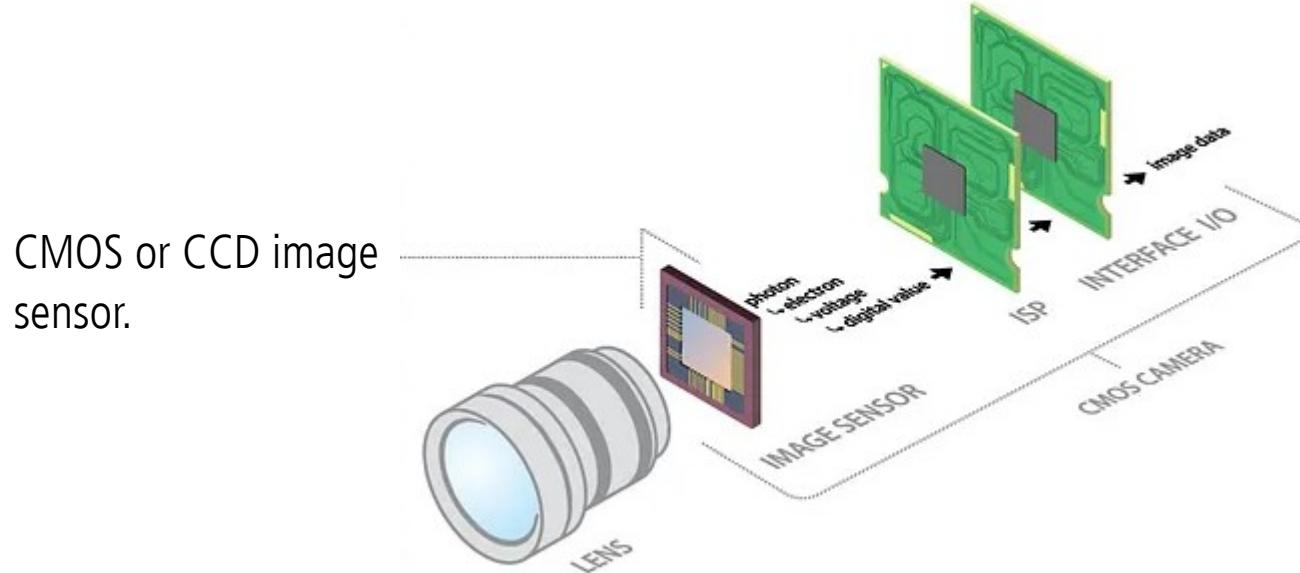
Basic Parts of a DSLR Camera

# Real camera systems

- Camera contains many small discrete **sensors**: CMOS/CCD
  - Each measures incoming **radiance** values and converts it to **color** signal
  - Radiance: light energy per ray
- Sensors are placed into light-proof enclosure with small opening - **aperture**
  - Such setup restricts light where light can enter and strike sensor
- **Lens** is placed at the aperture to focus the light so each sensor receives light from only a small set of incoming directions

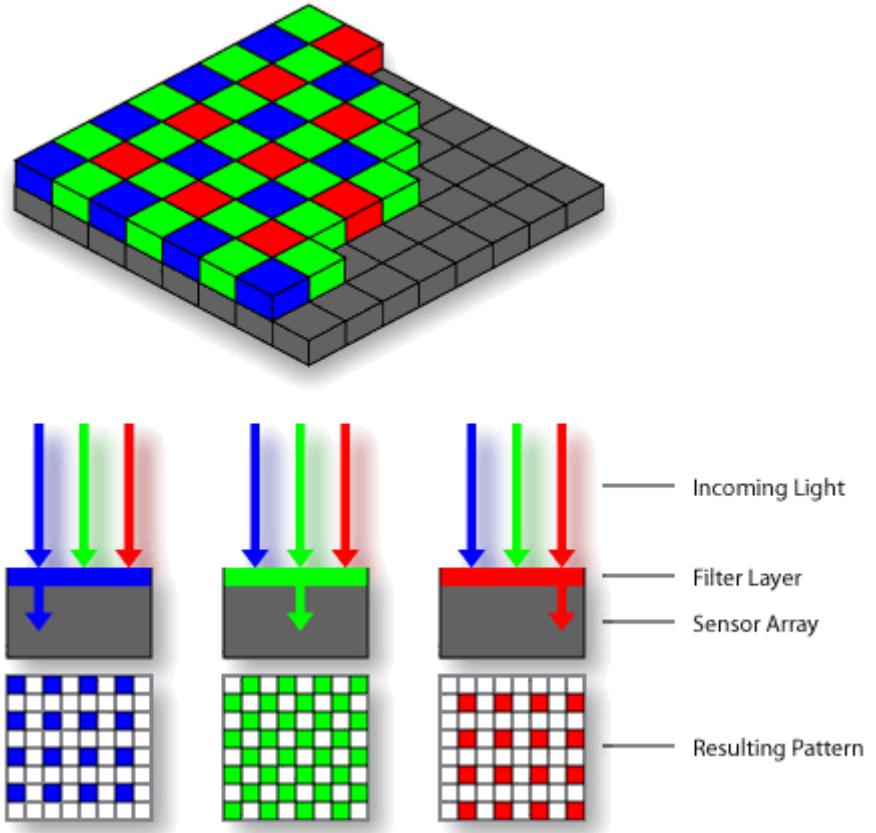
# Camera image sensor

- Sensor converts incident light energy into analog signal (e.g., voltage)
- Array of sensors are used to form an image



# Camera image sensor

- Each sensor records incoming light energy per second
- Only one signal over whole cell area is captured
- Sensor measures only R,G,B wavelengths
- Sensor cell can be seen as a pixel

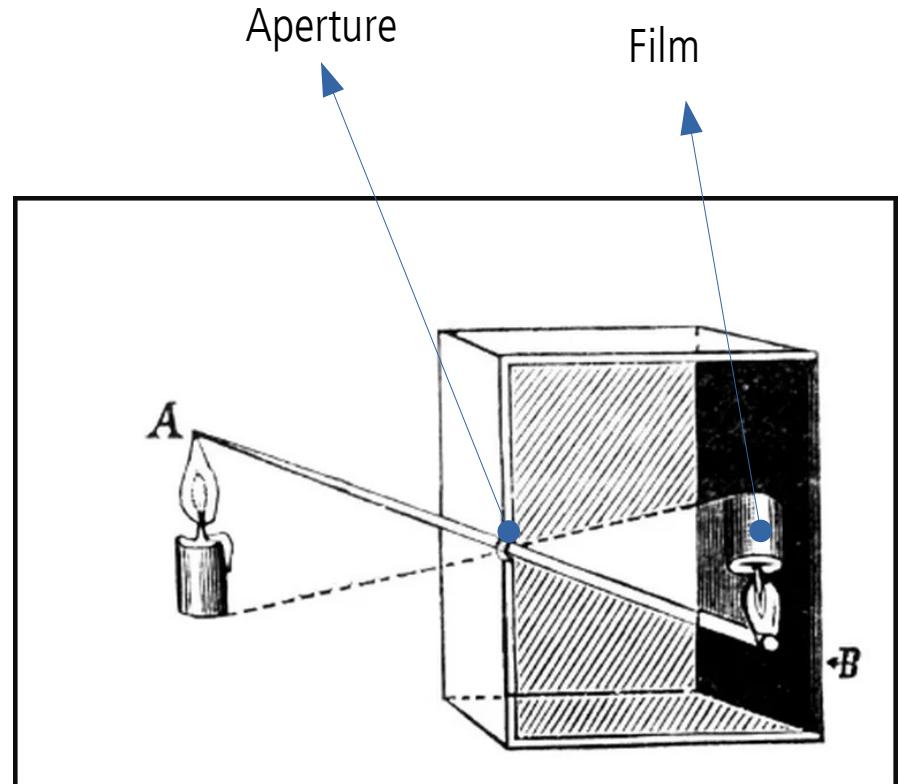


[https://simple.wikipedia.org/wiki/Bayer\\_filter](https://simple.wikipedia.org/wiki/Bayer_filter)

# Image formation

# Camera obscura

- Lightproof box with:
  - Black interior (to prevent light reflection)
  - Very small opening: **aperture**
  - Light-sensitive **film** opposite to aperture
- When aperture is opened, film is exposed to light
  - Light passing through the hole forms an inverted image of the external scene on the film
  - Time which aperture must be opened to capture image on film is called **exposure**
- A foundation for modern photographic cameras



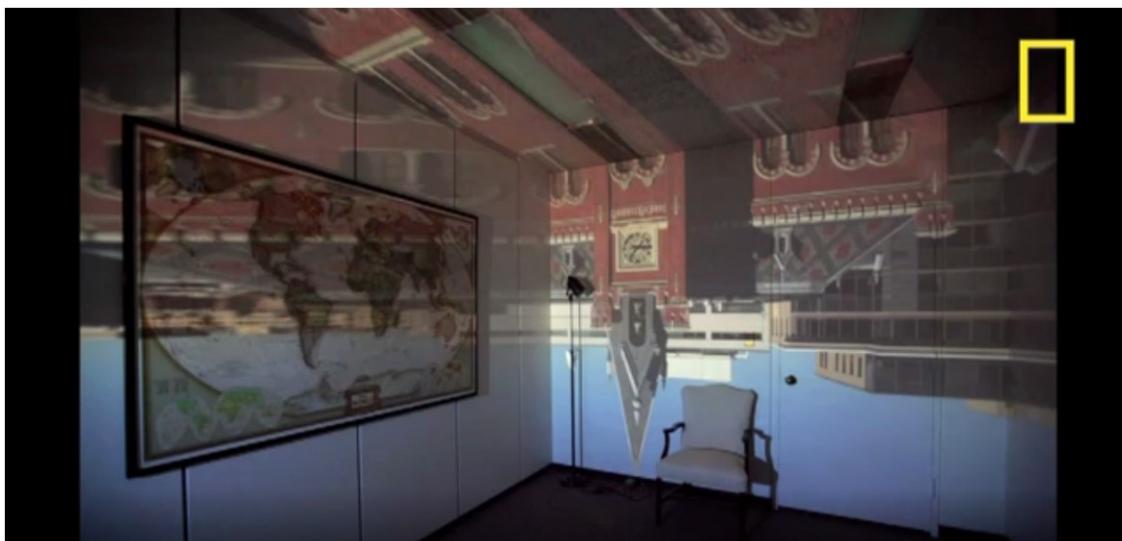
[https://www.inquirer.com/philly/entertainment/20160806\\_Camera\\_Obscura\\_at\\_Fleisch\\_er\\_throws\\_new\\_light\\_on\\_photography.html](https://www.inquirer.com/philly/entertainment/20160806_Camera_Obscura_at_Fleisch_er_throws_new_light_on_photography.html)

# Camera obscura = Dark room



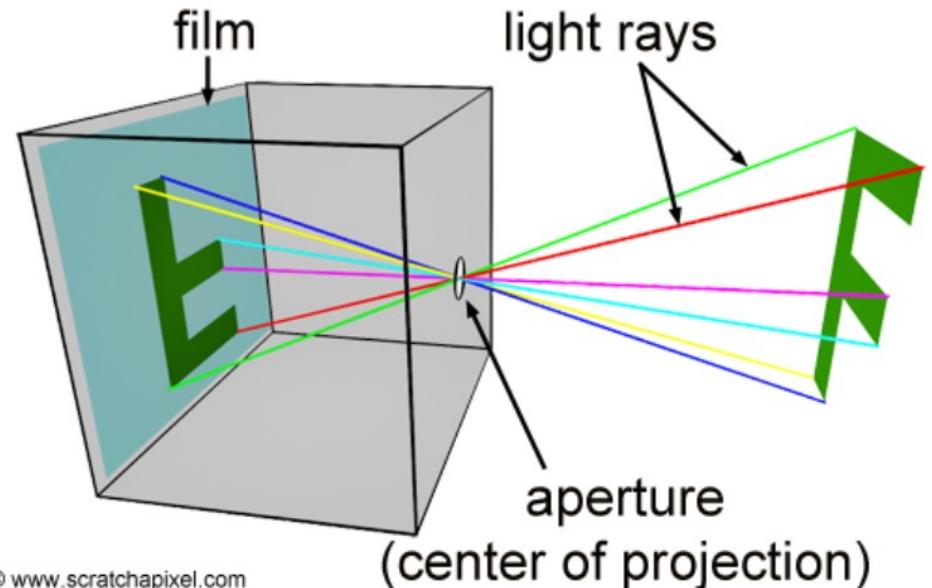
View from the window.

Room made into camera obscura projecting view from the window on the wall.



# Pinhole camera

- Main elements:
  - **Aperture**: center of projection: all rays entering converge here and diverge from it to the other side
  - **Film**: light hitting the film forms the image
- Image formation
  - Aperture is opened to expose film to light
  - Ray reflected from the world in point P enters the camera and intersects film in one point
  - Each point in the visible portion of the scene corresponds to a single point on the film



© www.scratchapixel.com

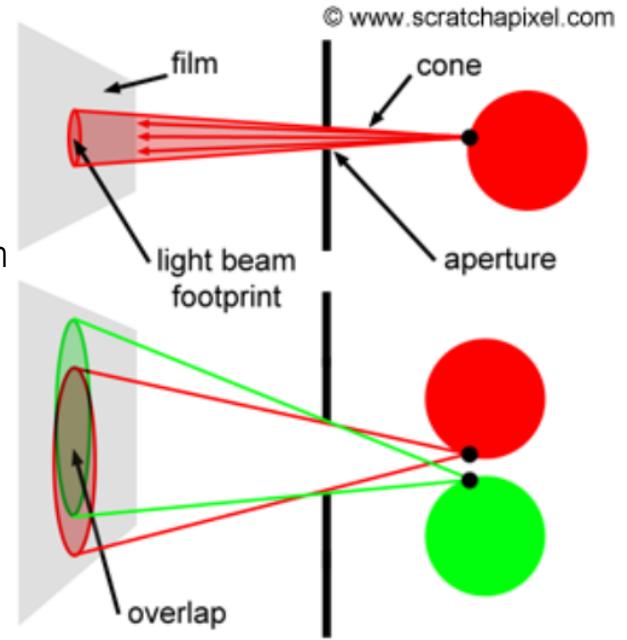
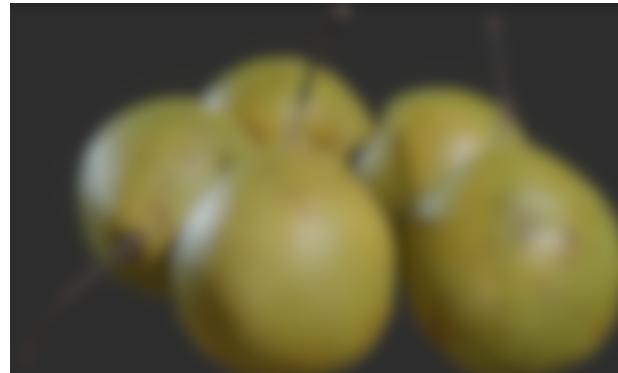
<https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera>

# Aperture size

- **Ideal pinhole:** aperture is so small that only one ray passes through it
  - Formed image is **sharp** since each point of the object maps to the one point of the film
- **Aperture has certain size**
  - Angle of cone of rays is determined by the size of aperture
  - Smaller cone → **sharper image**
  - Larger aperture → **blurred image**



[https://polyhaven.com/a/food\\_pears\\_asian\\_01](https://polyhaven.com/a/food_pears_asian_01)

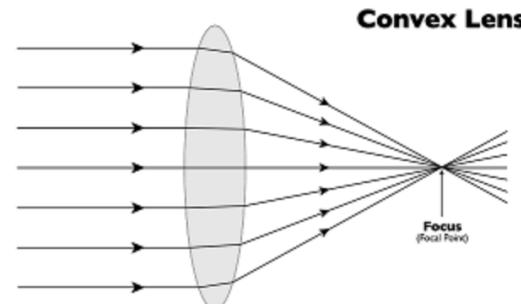
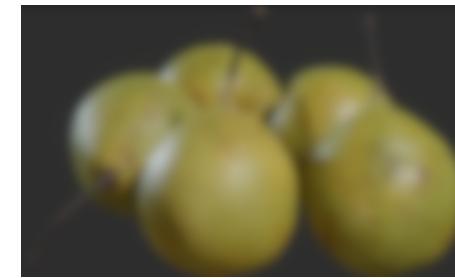


Cones are overlapping and single point appears multiple times on film  
→ blur (out of focus).

<https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera>

# Aperture and Lenses

- Small aperture: sharp image, but long exposure times needed leading to motion blur
- Large aperture: blurred image
- Solution is to use lens in front of aperture
  - Rays entering camera are gathered (converged) and focused to one point on a film plane
  - Aperture can now be larger enabling smaller exposure time while blur is canceled with lens



# Depth of field

- **Lens camera** introduce **depth of field**: distance between nearest and the farthest object from the scene that appears sharp in the formed image
- **Pinhole cameras have infinite depth of field**
  - Because of this, images rendered with pinhole cameras are completely sharp and thus look artificial



Depth of field: <https://www.adobe.com/uk/creativecloud/photography/discover/bokeh-effect.html>

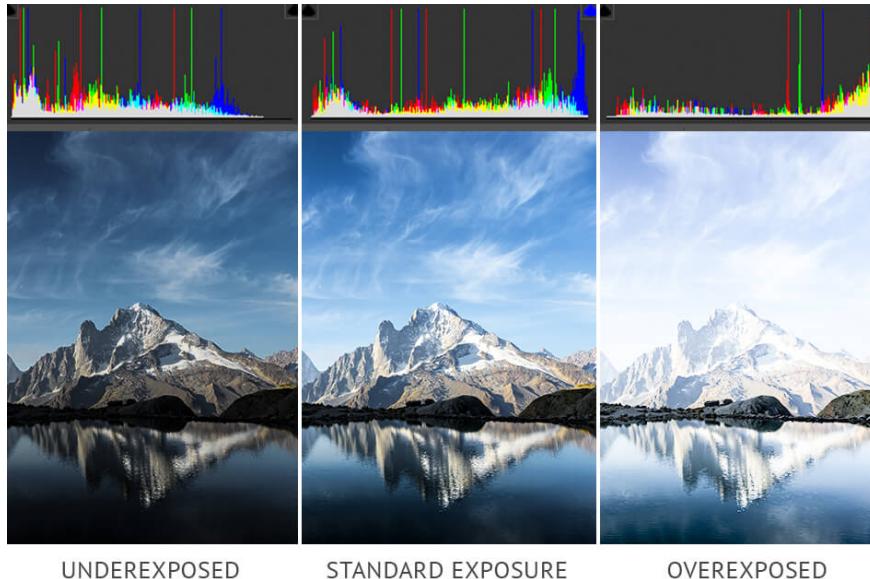
# Bokeh



Lights out of focus are aesthetically pleasing → bokeh effect.

# Exposure

- Time in which the film is exposed to light is called **exposure** time
  - Determines image brightness
- Amount of visible details in lit and shadowed areas: **dynamic range**
  - Real world and 3D scenes can have high difference in dynamic range, but display devices are limited in dynamic range



<https://fixthephoto.com/what-is-exposure-composition.html>

# Exposure

- If aperture is small, longer time is needed for image to form on a film
  - In this case, **motion blur** can appear for objects which are not perfectly still



Earth rotation causes motion blur in long-exposure photos



[https://en.wikipedia.org/wiki/Motion\\_blur](https://en.wikipedia.org/wiki/Motion_blur)

# Exposure

- Exposure is controlled with:
  - Aperture size (f-stop) → depth of field
  - Shutter speed → motion blur
  - Sensitivity (ISO) (gain) → noise



<https://photonify.com/3-steps-for-adjusting-the-depth-of-field-on-your-camera/>

<https://petapixel.com/exposure-triangle/>

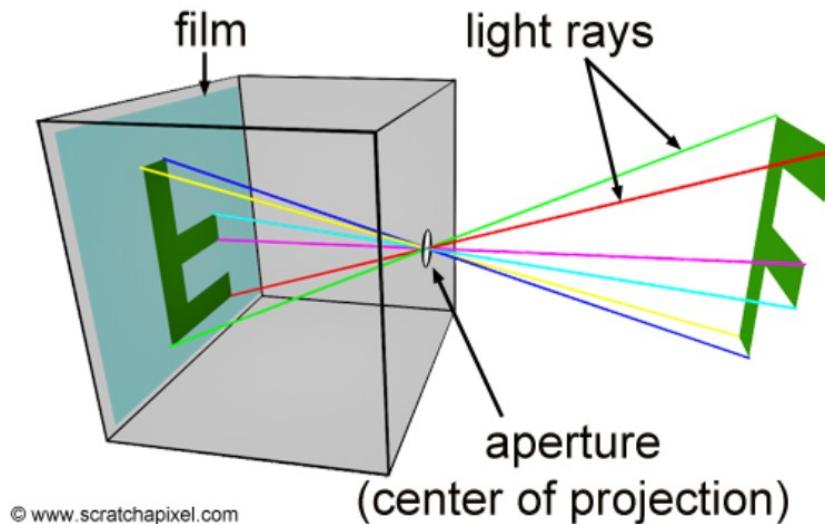
<https://actioncamera.blog/2017/02/22/the-exposure-triangle/>

<https://www.shutterbug.com/content/give-photos-dramatic-perspective-motion-and-blur-simple-photoshop-technique-video>

# Pinhole camera model

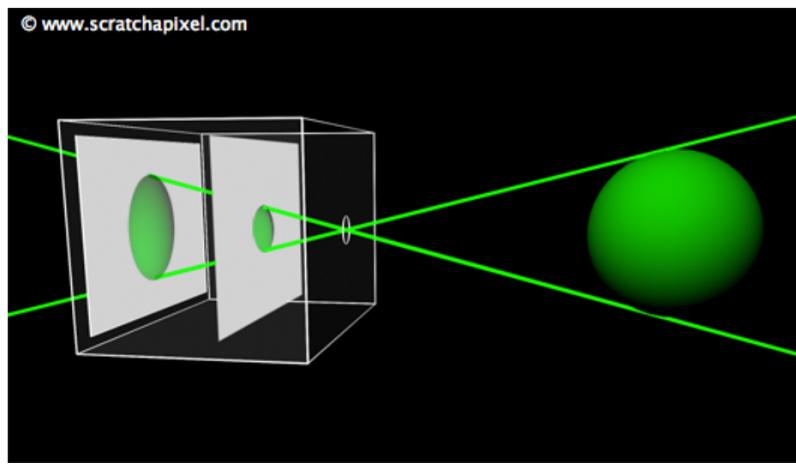
# Pinhole camera model

- Complex lens system is simplified: **pinhole camera**
- Main elements:
  - **Film** (image plane)
  - **Aperture** (small opening)
- Parameters:
  - Focal length
  - Angle of view
  - Film size

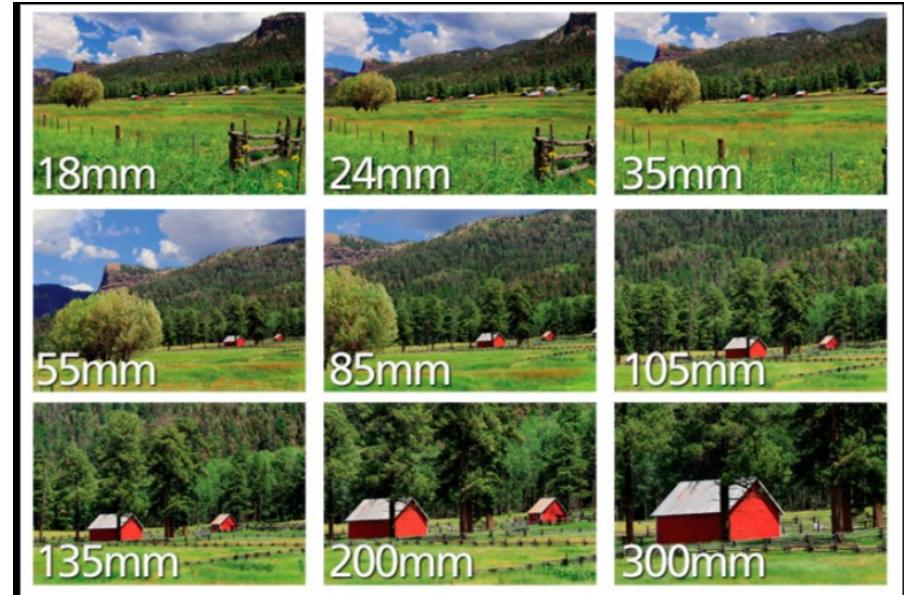


# Focal length

- Distance of film plane from aperture defines amount of scene that we see
  - Moving image plane (film plane) closer to aperture effectively performs **zoom out**
  - Moving film plane away from aperture effectively performs **zooming in**
- This parameter is called **focal length** or **focal distance**

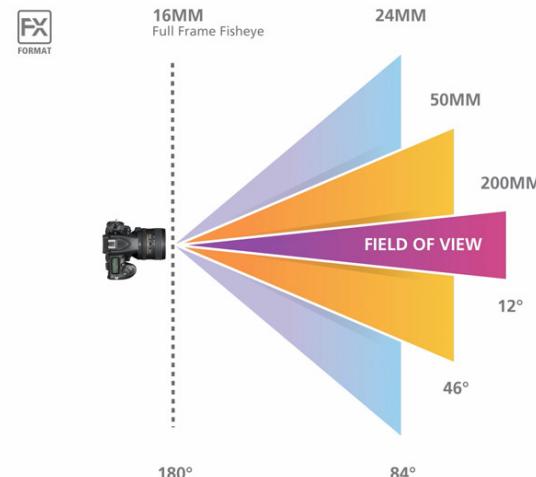
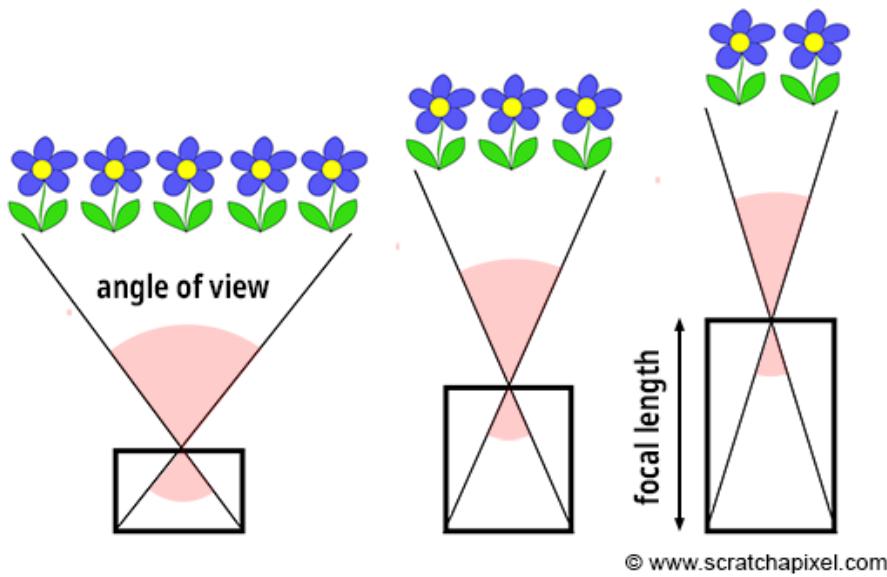


<https://www.nikonusa.com/en/learn-and-explore/a/tips-and-techniques/understanding-focal-length.html>



# Angle of view

- Angle of triangle apex defined with aperture and film edges is called angle of view (AOV) or field of view (FOV)
- Changing focal length, changes AOV/FOV



Focal length and corresponding field of view  
<https://www.nikonusa.com/en/learn-and-explore/a/tips-and-techniques/understanding-focal-length.html>

# Focal length and angle of view

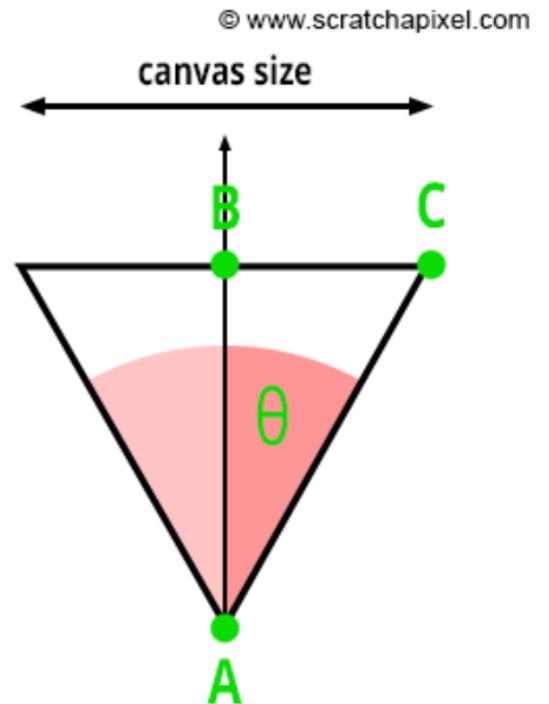
- Relation between focal length ( $f = AB$ ) and angle of view ( $\theta$ )
- Longer focal lengths result in smaller FOV (more zoom)
- Short focal length results in larger FOV (less zoom)

$$\tan(\theta) = \frac{BC}{AB}$$

$$BC = \tan(\theta) * AB$$

$$\text{Canvas Size} = 2 * \tan(\theta) * AB$$

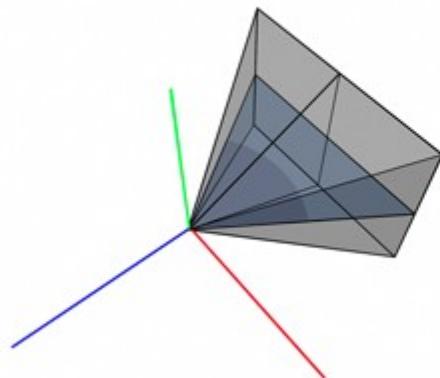
$$\text{Canvas Size} = 2 * \tan(\theta) * \text{Distance to Canvas} .$$



# Field of view

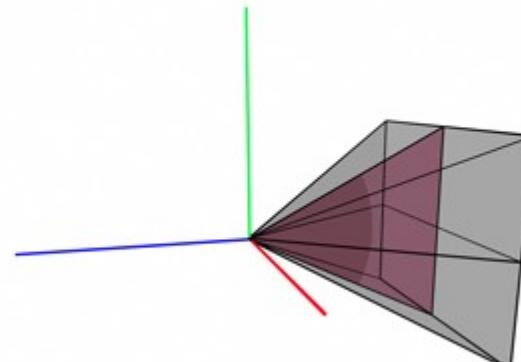
- Triangle defined with aperture and film edges is a pyramid and we distinguish **horizontal and vertical angle (field) of view (AOV/FOV)**
- Using fixed film aspect ratio, horizontal field of view determines vertical field of view and vice versa

Horizontal Field of View



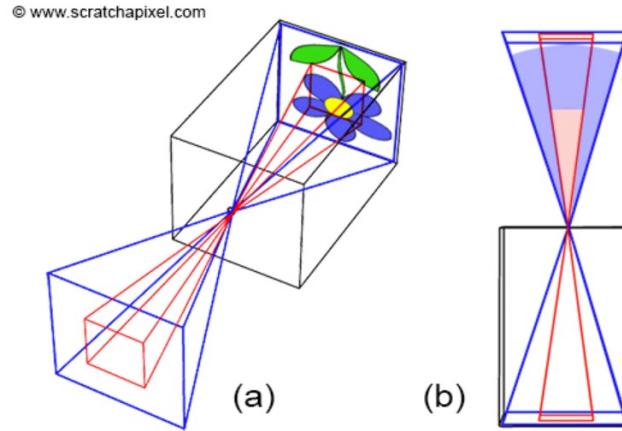
© www.scratchapixel.com

Vertical Field of View



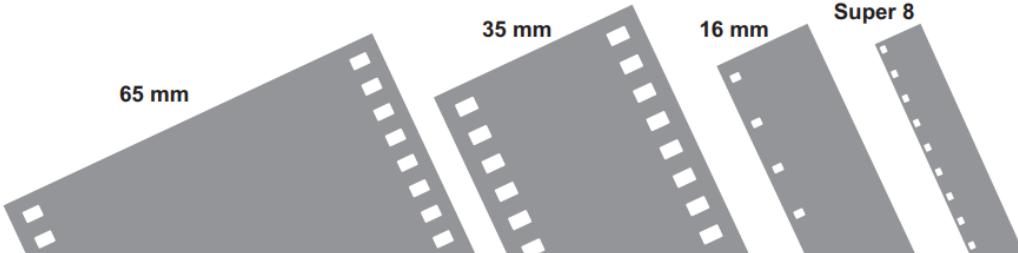
# Film size

- Angle of view depends on film (image plane) size
  - Film has horizontal and vertical dimensions
  - Using fixed film aspect ratio, horizontal dimension determines vertical dimension
- Smaller area of film size implies smaller angle of view and vice versa.
  - Capturing the same extent of the scene with larger film requires adjusting focal length



# Film size

- Larger film formats were developed enabling more details and better image quality
- In digital cameras, film is replaced with sensor, thus we talk about sensor size
  - Sensor size has the same role as film size



Gauge refers to the width of film. Four are common: 65mm, 35mm, 16mm, Super 8.

[http://www.theodoropoulos.info/attachments/076\\_kodak05\\_Film\\_Types\\_and\\_Formats.pdf](http://www.theodoropoulos.info/attachments/076_kodak05_Film_Types_and_Formats.pdf)

35 mm / 135	120 / Medium Format	4x5 / Large Format
 36 mm 24 mm	 4.5 cm 6 cm 645	 5 inch 4 inch 6 cm 6 cm 6 cm 7 cm 67

<https://www.learnfilm.photography/guide-to-medium-format-photography/>

1920x1080

# Film size and image resolution

- **Image resolution:** width  $\times$  height pixels
- **Size of sensor and number of pixels on sensor are independent parameters**
  - Angle of view doesn't depend on number of pixels on image sensor
- Image quality depends on:
  - Image sensor size
  - Number of pixels on sensor (resolution)
- Higher resolution images will have more details



20% of 1920x1080



# Image resolution and aspect ratio

- Image aspect ratio (device aspect ratio) is calculated from image resolution (width and height):

$$\text{Image (or Device) Aspect Ratio} = \frac{\text{width}}{\text{height}}$$

- Film and display devices (computer screens, television) have standardized aspect ratios

- 4:3

- Old television systems and computer monitors (e.g., resolution 640x480)
  - Default for digital cameras; 35mm films

- 5:3 and 1.85:1

- Often used for film

- 16:9

- Aspect ratio used by HD TV, monitors and laptops (e.g., resolution: 1920x1080)



<https://photographylife.com/aspect-ratio>

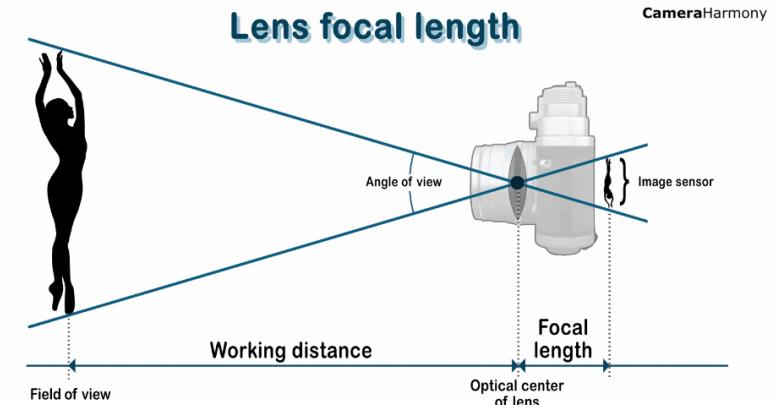
<https://www.studiobinder.com/blog/aspect-ratio/>

# Film and image aspect ratio\*

- In computer graphic camera model, sensor (film/canvas) aspect ratio can be different from the image (device) aspect ratio.
  - Difference leads to stretching the image in either  $x$  or  $y$  directions
- Solution: canvas aspect ratio is often directly computed from the image aspect ratio
  - Example, if image resolution is 640x480, canvas aspect ratio will be set to 4:3.

# Pinhole camera parameters: recap

- Parameters:
  - Focal length (focal distance)
  - Angle of view (field of view)
  - Film size
  - With two parameters we can always infer the third one.
- Angle of view is determined by:
  - Focal length: longer focal length → narrower angle of view
  - Film size: larger film size → wider angle of view



# Virtual pinhole camera

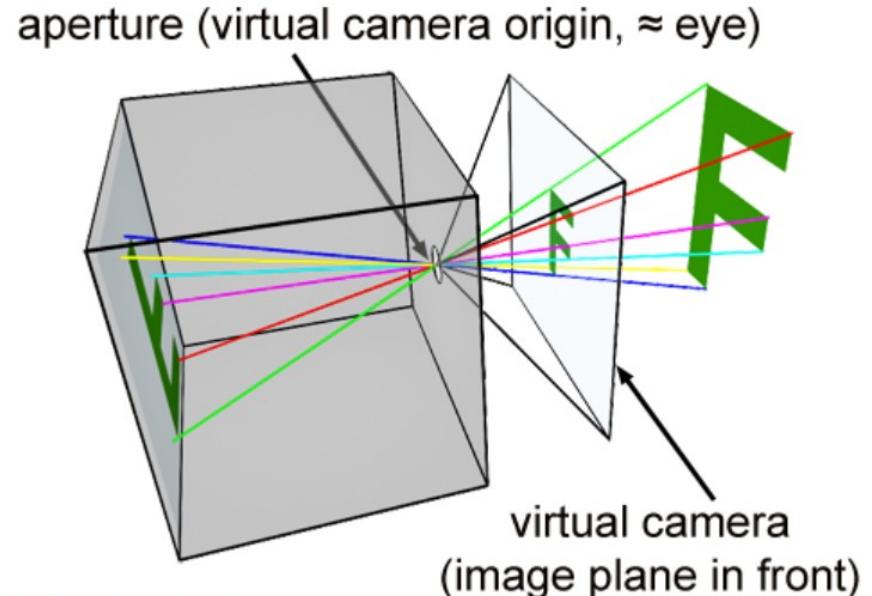
Camera model for rendering

# Virtual pinhole camera

- Virtual pinhole camera model in rendering is used to deliver image similar to those produced by a real camera
  - Size and shape of real and rendered object must be the same
  - Combining rendered and live action footage must be possible
  - Expressive and artistic possibilities must be enabled
- Pinhole camera model is used in almost all production software
  - Godot: [https://docs.godotengine.org/en/stable/classes/class\\_camera.html](https://docs.godotengine.org/en/stable/classes/class_camera.html)
  - Unity: <https://docs.unity3d.com/ScriptReference/Camera.html>
  - Blender: <https://docs.blender.org/manual/en/latest/render/cameras.html>
  - Houdini: <https://www.sidefx.com/docs/houdini/nodes/obj/cam.html>

# Virtual pinhole camera

- In graphics, pinhole camera geometry is not directly simulated
- Virtual pinhole camera simulates only directions from which sensors (pixel) receive light
- Virtual image plane is placed in front of aperture and rays are traced from aperture to each pixel in virtual image plane
  - Camera rays → color of virtual image plane pixels

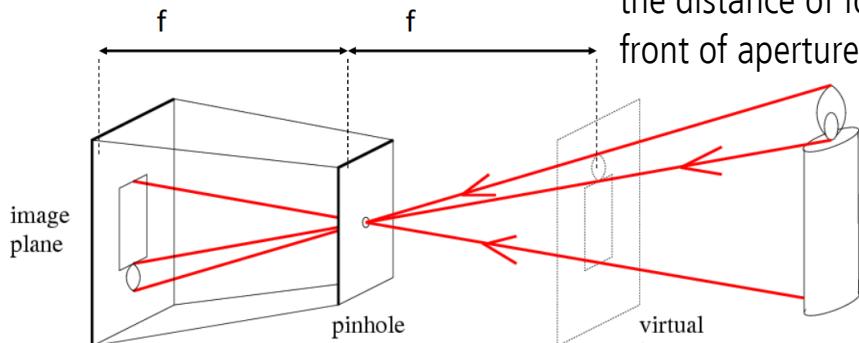


© www.scratchapixel.com

<https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera/virtual-pinhole-camera-model>

# Virtual pinhole camera

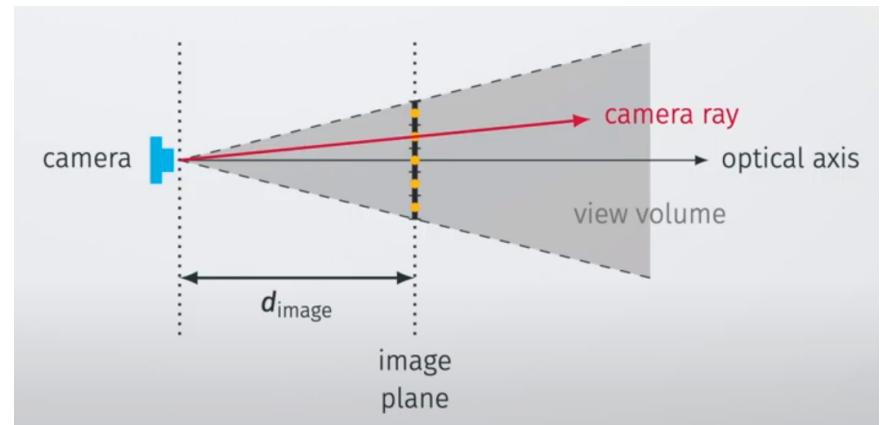
- Following rays from aperture to image plane pixels and into the scene forms image on image plane
- This results **perspective projection**



Real pinhole camera produces images which are upside down

Simulated pinhole camera produces images which are correctly orientated

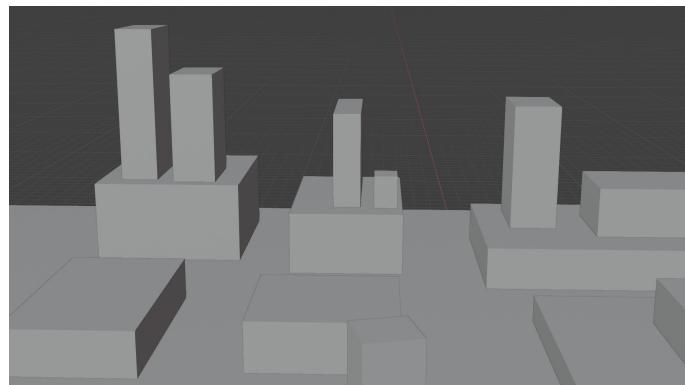
Virtual image plane is placed on the distance of focal length ( $f$ ) in front of aperture



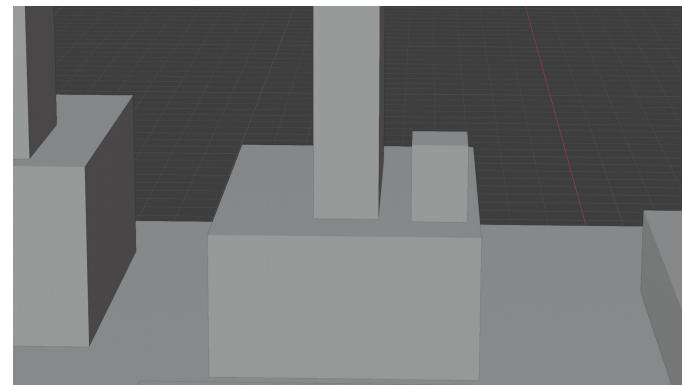
Distance from camera to image plane is focal length

# Viewing frustum

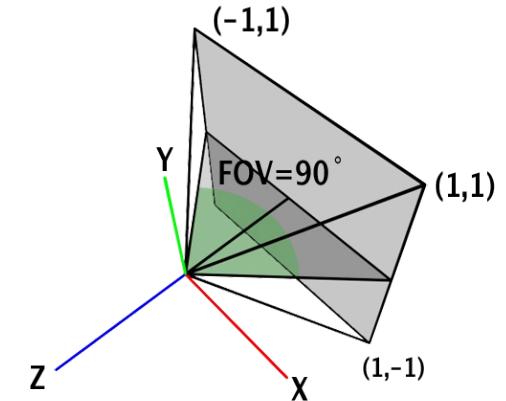
- Viewing frustum defines portion of scene visible to camera
- Viewing frustum of pinhole camera is defined with:
  - Camera origin (eye, aperture)
  - Field (angle) of view: film size and focal length



Focal length 50mm

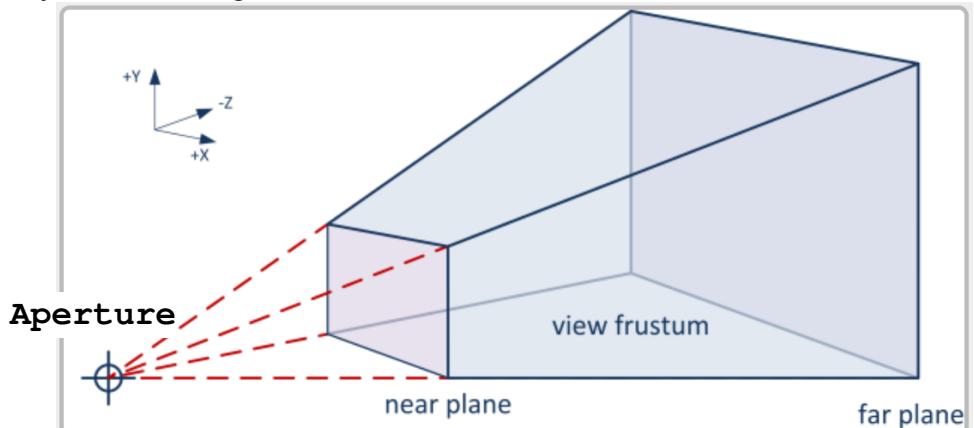


Focal length 120mm



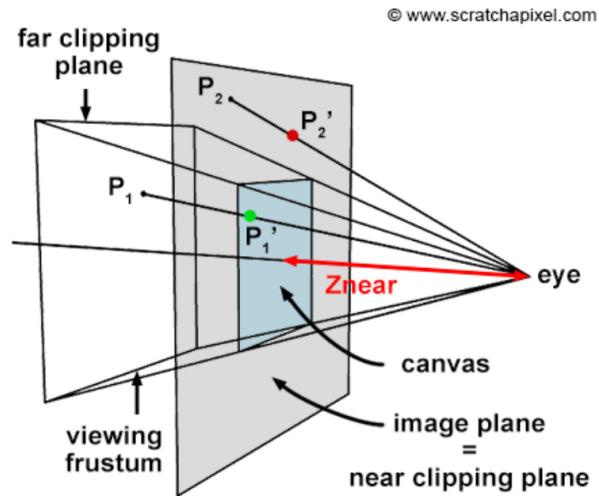
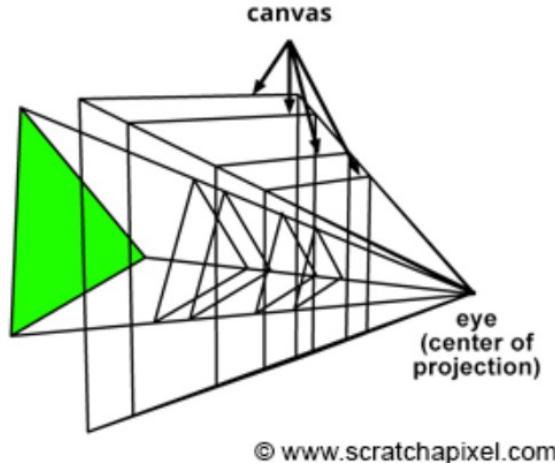
# Near and far clipping planes

- Virtual pinhole camera additionally has near and far clipping planes
  - Virtual planes placed in front of camera and parallel to image plane.
  - Location of clipping planes is measured along camera's line of sight (camera's local z axis)
- Objects in 3D scene closer than near or further than far plane are not visible in image plane.
- Near and far clipping planes are needed for resolving precision issues
  - Precision can be lost if distance between closest and furthest object is too large



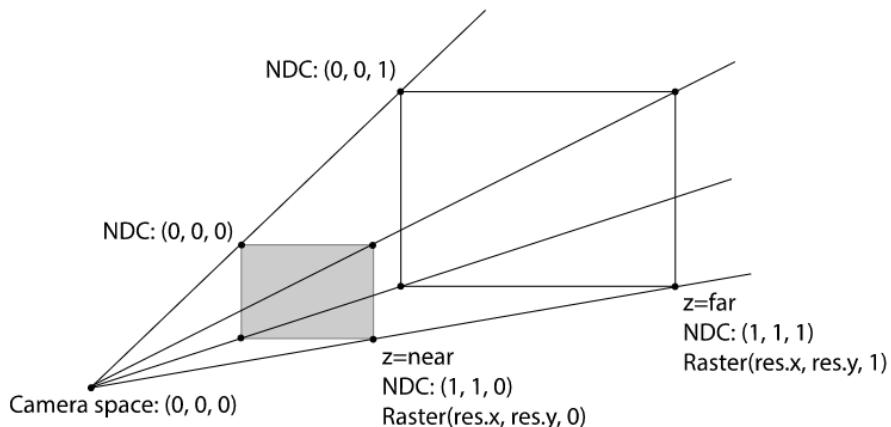
# Image plane and clipping planes

- Image plane is often placed at near clipping plane for computation simplicity
  - Image plane can be placed anywhere on camera's line of sight (camera's local z axis) between near and far clipping planes
  - Size of image plane depends on distance from eye.



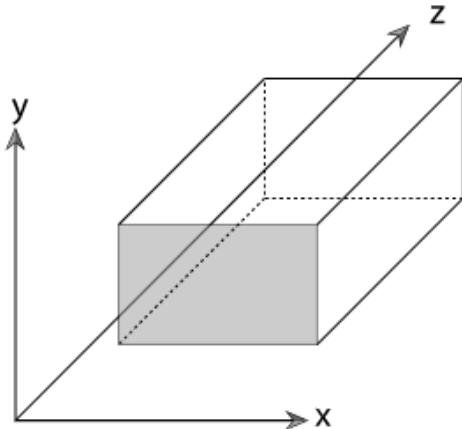
# Perspective camera

- Most often used from film and game rendering
- Rays are generated from one point (eye, aperture) through virtual image plane
- **Simulates foreshortening effect:** more distant objects appear smaller



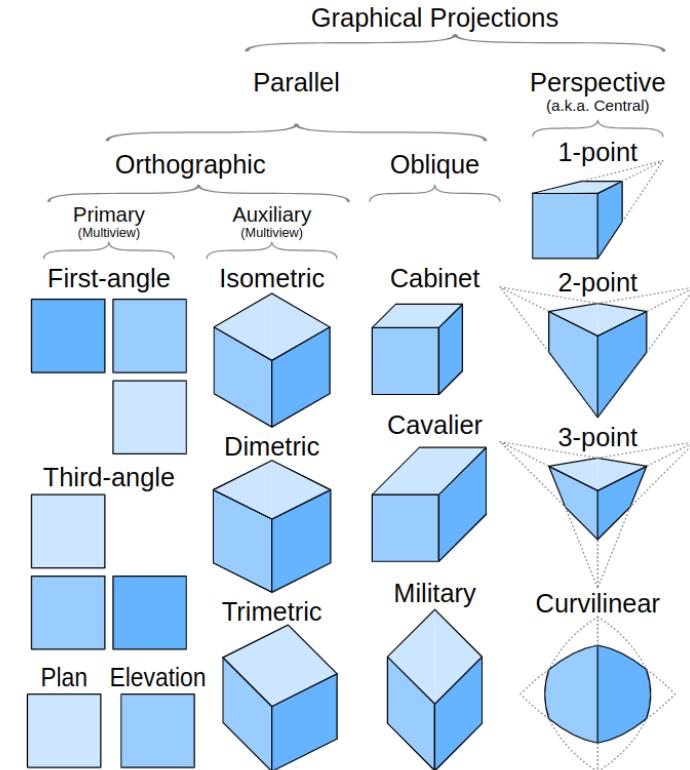
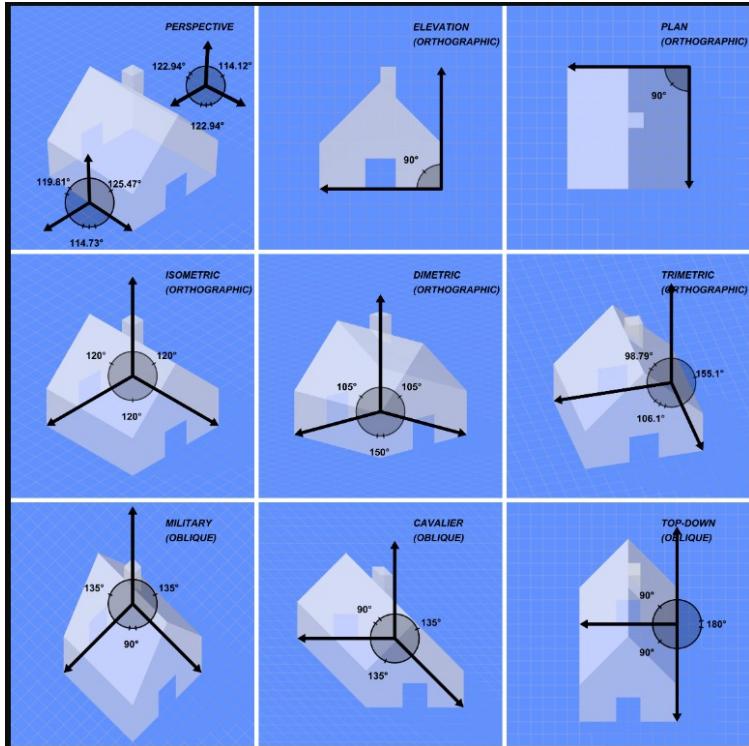
# Orthographic camera

- Useful for architecture, engineering, design, isometric games, etc.
- Rays are generated parallel through virtual image plane
- Objects are of same size regardless of distance



[https://store.steampowered.com/app/1573390/Lilas\\_Sky\\_Ark/](https://store.steampowered.com/app/1573390/Lilas_Sky_Ark/)

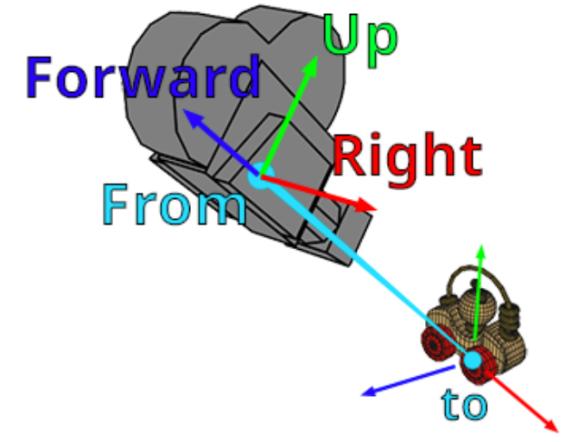
# Other camera projection types



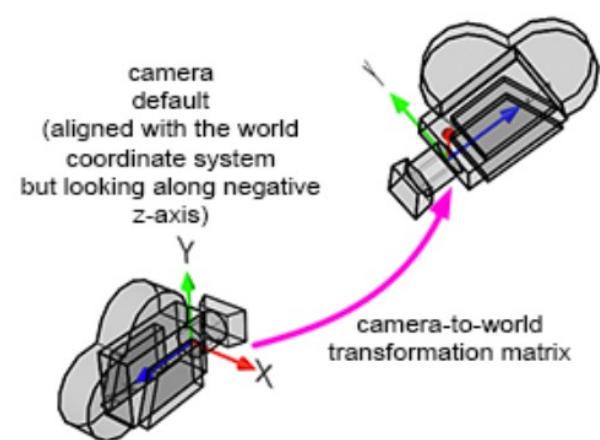
[https://en.wikipedia.org/wiki/Orthographic\\_projection](https://en.wikipedia.org/wiki/Orthographic_projection)

# Camera orientation and position

- Camera orientation is defined with Forward, Up, Right vectors.
- Camera orientation and position can be defined with **look-at** notation:
  - Point in space where camera is positioned (**From**)
  - Point in space where camera is looking (**To**)
- **Look-at matrix**
  - Forward, Up, Right can be calculated based on From, To
  - Forward, Up, Right, From define look-at matrix
  - Defines complete transformation needed for moving camera in 3D scene



[www.scratchapixel.com](http://www.scratchapixel.com)



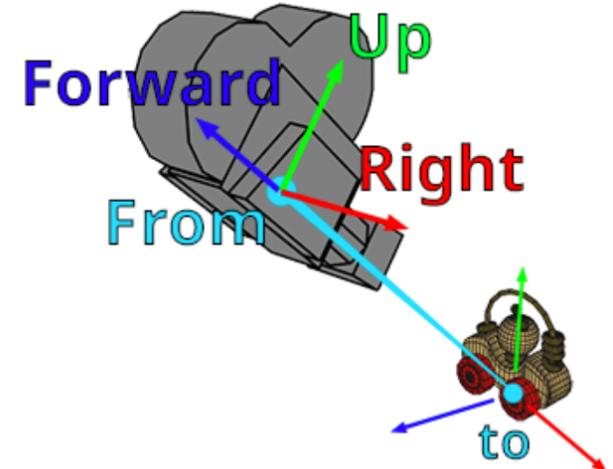
# Look at matrix

- 4x4 transformation matrix → transformation of camera from its local (camera) space to world space in 3D scene: **camera-to-world** and its inverse **world-to-camera** matrix

Forward = normalize(From - To)

Right = crossProduct(randomVec, Forward)

Up = crossProduct(forward, right)



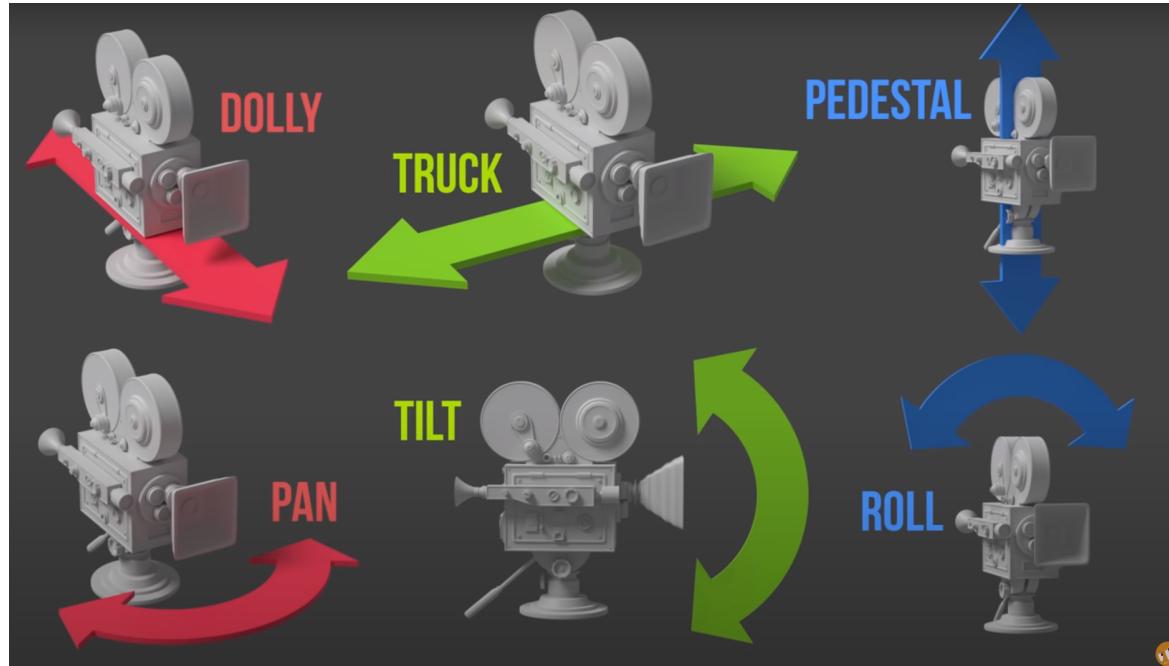
[www.scratchapixel.com](http://www.scratchapixel.com)

$Right_x$	$Right_y$	$Right_z$	0
$Up_x$	$Up_y$	$Up_z$	0
$Forward_x$	$Forward_y$	$Forward_z$	0
$From_x$	$From_y$	$From_z$	1

RandomVec = (0, 1, 0) or other if Forward is close to (0, 1, 0) or (0, -1, 0)

# Moving camera\*

- Translations: moving in axis directions
- Rotations: rotation around axis

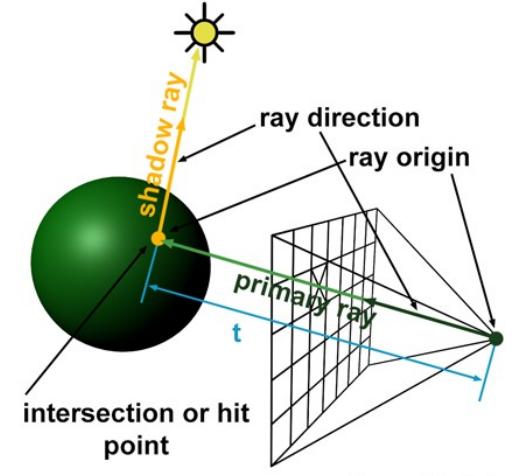


# Virtual pinhole camera image formation

- How image is formed using virtual pinhole camera depends on rendering techniques:
  - Ray-tracing-based rendering
  - Rasterization-based rendering

# Camera in ray-tracing

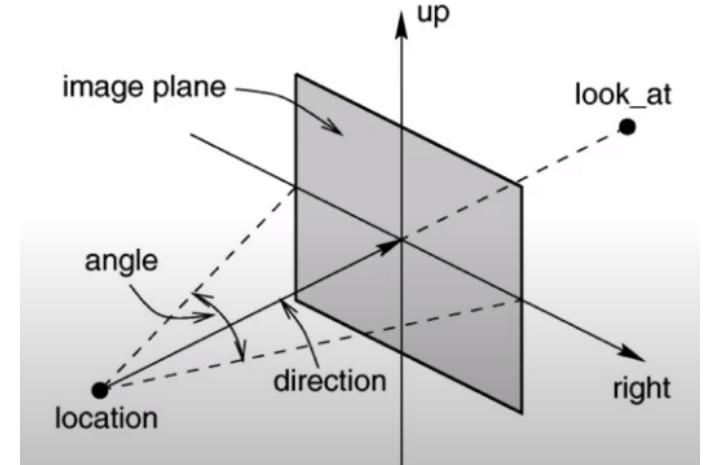
- **Image centric approach:** generate ray for each pixel of virtual image plane
  - Ray is generated using camera eye position and pixel position in camera image plane.
  - Ray is traced into the 3D scene where it intersects the objects – **ray casting** (foundational method for ray-tracing).
  - Color that is computed from intersecting the object is assigned to pixel



© www.scratchapixel.com

# Generating rays

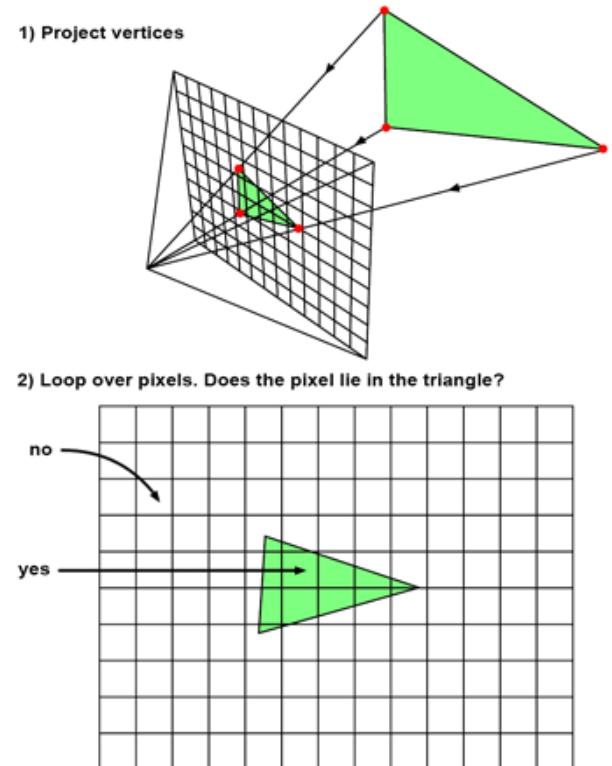
- Assume: image plane is  $[-1, 1] \times [-1, 1]$
- Rays  $r(t)$  can be generated if:
  - Image resolution  $r_x \times r_y$  is given
  - Position of eye (aperture)  $e$  is given
  - Vectors Forward, Right, Up are given
  - Focal length  $f$  or angle of view  $\alpha$  is given



$$r(t) = e + t * (f * \text{Forward} + (i + 0.5) / r_x * \text{Right} + (j + 0.5) / r_y * \text{Up})$$

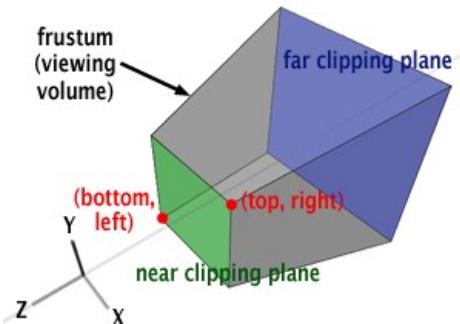
# Camera in rasterization

- Rasterization-based rendering projects objects (triangles) from 3D scene onto image plane and for each pixel finds a position of projected object
  - Object centric approach
- The process of finding pixel coordinates on image plane of 3D point is done as follows:
  - 3D points from world space are transformed to camera space: **world-to-camera matrix**
  - **Projection matrix** is applied on transformed point
  - Resulting points are convert to raster space (image plane with pixels)

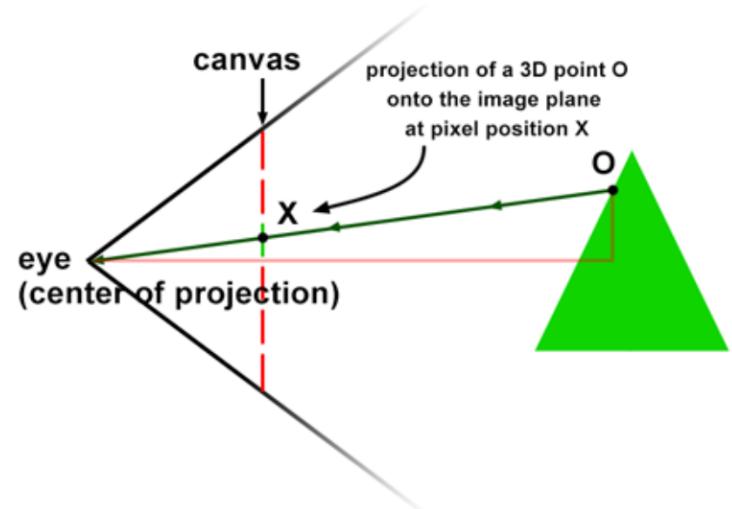


# Projection matrix

- In rasterization, camera is used to define perspective projection matrix.
- General perspective projection matrix is defined with viewing frustum **near** (n), **far** (f), **left** (l), **right** (r), **top** (t) and **bottom** (b)

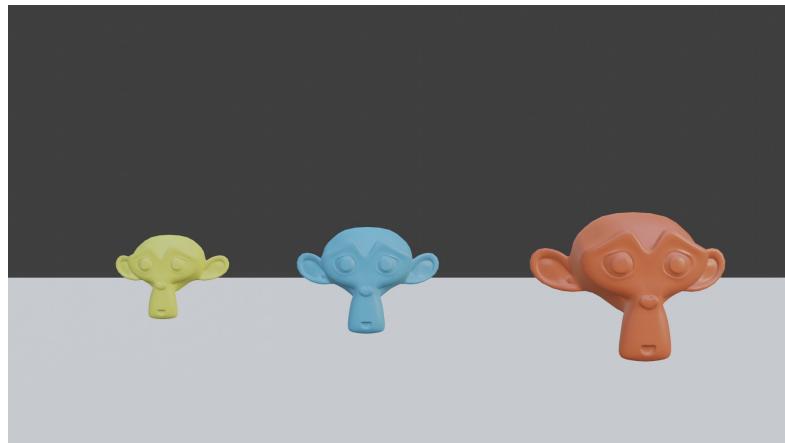


$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$



# Virtual pinhole camera notes

- Virtual pinhole camera has infinitesimally small aperture
  - Image will be perfectly sharp
  - Additional methods are needed from simulating lens effects such as depth of field
  - Lens system introduce focus plane in 3D scene where objects are in sharp focus and blurred otherwise



Perfectly sharp image



Depth of field

# Virtual pinhole camera notes

- Virtual pinhole camera has infinitesimally small aperture → light energy is carried instantaneously by ray
- Time in image plane is exposed to light doesn't affect amount of light falling on image plane
  - Exposure is simulated in already rendered image or adding it as parameter to lights

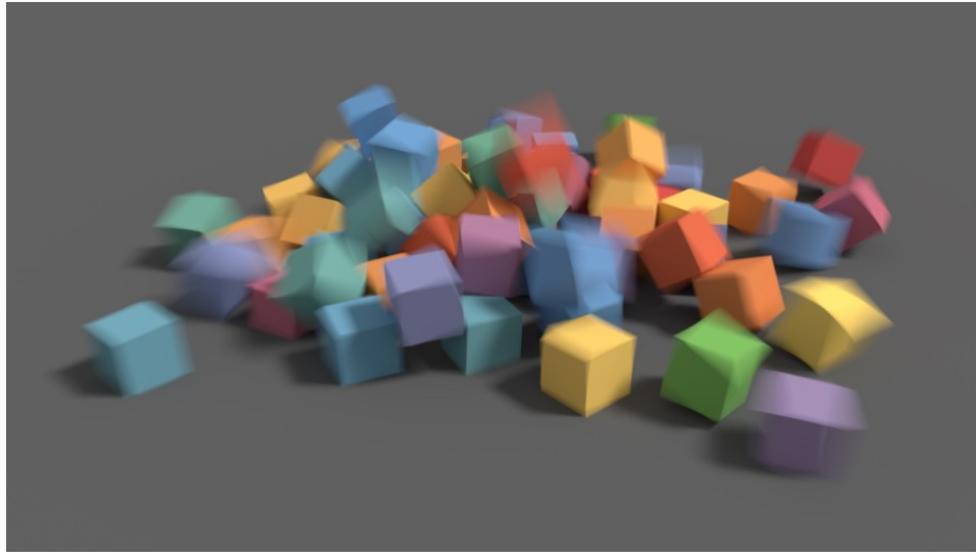


Different exposures of the same render:

[https://docs.blender.org/manual/en/latest/render/color\\_management.html](https://docs.blender.org/manual/en/latest/render/color_management.html)

# Virtual pinhole camera notes

- In simulated camera, light transport is considered instant, therefore, simulation of motion blur requires additional simulation
- Motion blur must be additionally simulated using **shutter speed**: time in which camera door is opened to allow incoming light



Motion Blur

[https://docs.blender.org/manual/en/latest/render/cycles/render\\_settings/motion\\_blur.html](https://docs.blender.org/manual/en/latest/render/cycles/render_settings/motion_blur.html)

# More into topic

- More about camera film, aspect ratio:

<https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera/how-pinhole-camera-works-part-2>

- Simulating camera lenses:

- [https://www.pbr-book.org/3ed-2018/Camera\\_Models/Realistic\\_Cameras](https://www.pbr-book.org/3ed-2018/Camera_Models/Realistic_Cameras)

- <https://developer.nvidia.com/gpugems/gpugems/part-iv-image-processing/chapter-23-depth-field-survey-techniques>

# Summary questions

- [https://github.com/lorentzo/IntroductionToComputerGraphics/tree/main/lectures/9\\_light](https://github.com/lorentzo/IntroductionToComputerGraphics/tree/main/lectures/9_light)

# Literature

- <https://github.com/lorentzo/IntroductionToComputerGraphics>
- ScrachAPixel lesson:  
<https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera>