High Dynamic Range Imaging
EPL 656

Digital Cameras

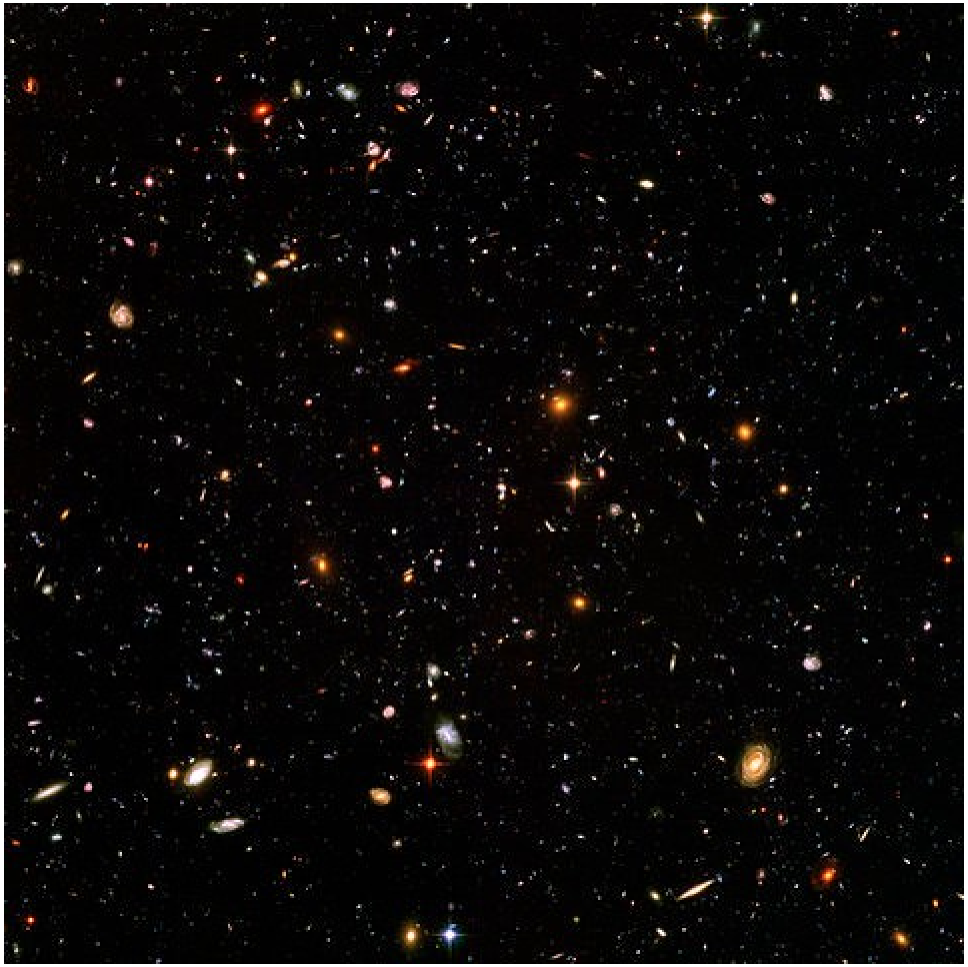
- High quality cameras today: DSLR (digital single lens reflex)
- Light enters though the lens and passes through a hole called aperture
- Mechanical shutter opens to let the light in
- Light is recorded on the image sensor
- Sensor value proportional to
  - How long the shutter stayed open (exposure time)
  - How open was the aperture (aperture value)

Exposure

1.6s 1/2s 1/10s 1/100s

- We can achieve similar results using aperture control

Hubble Telescope



Exposure time: 96 hours!

Camera Sensor Capabilities

- JPEG images: 24 bits per pixel
  - 8 bit per color channel (red-green-blue)
  - Pixel values: 0-255 for each color
- Camera sensor: 48 bits per pixel
  - Stored as RAW images
  - Actual range 0-10000 ~ per channel
  - Photographers work on these images
- Real World:
  - 0-1000000000
  - High dynamic Range (HDR)
  - How do we capture this?

Low Dynamic Range

High Dynamic range

Merging LDR Images

- Main idea:
  - No exposure/aperture setting can capture full dynamic range
  - Capture multiple images at different settings and merge them
- Keep aperture constant
  - Aperture affects depth of field
  - Hard to merge images if focus point is different
  - Change only exposure time
- Need to be able to compare pixel values taken at different exposures
  - The following is true for DSLR cameras:
  - **Pixel value = (incoming light) * (camera function) * (exposure time)**
  - For RAW images the camera function is a constant and can be ignored
  - This relation allows us to compare pixel values taken at different exposures

HDR Creation

- Simple method: Select the best exposure for each pixel
  - For each pixel location x,y get pixel values from all LDR images
  - Select the one closest to the middle of the allowed range
    - For 8-bit(0-255) images : 127
  - Divide by exposure time (this increases the dynamic range!)
- Works but its prone to noise

HDR Creation

- Debevec Method:
  - For each pixel x,y:
    - Loop over all LDR images and get pixel x,y
    - Weight pixels using a weighting function
      - Same idea as before
      - Saturated pixels get smaller weights
    - Multiply by weight and divide by exposure value
      - Weighted pixel = pixel*weight/exposure
    - Average all pixels together
- Handles noise better because of the averaging