

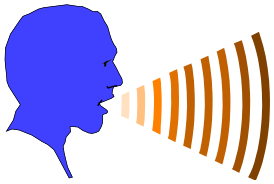
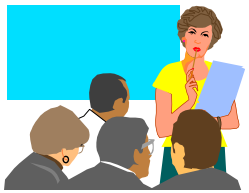



Image and Video Processing

Digital Video Basics

Yao Wang
Polytechnic School of Engineering, New York University

Outline

- What is video? From analog to digital
- Temporal frequency caused by motion
- Frequency response of HVS and its impact on necessary video sampling rates
- Sampling theorem for video capture and display
- Moving object detection by frame difference
- Moving object tracking by block matching

What is Video?

- Real-world scene is a 3D signal changing in time F(X,Y,Z,t) (a 4D continuous space signal)
  - (X,Y,Z) are 3D spatial coordinate, t is time
- Video is a projection of the dynamic scene onto 2D camera plane f(x,y,t) (a 3D continuous space signal)
  - (x,y) is the projection of 3D point (X,Y,Z) onto 2D image plane
  - For given t, f(x,y,t) is a 2D frame (or image)
- Digital video samples the 3D domain (x,y,t) to form a 3D discrete space signal f(m,n,k)
  - Frame rate (Temporal Resolution): number of frames/s (fps or Hz)
  - Frame size (Spatial Resolution): number of pixels/frame
- A color video has 3 color values at each pixel (m,n,k)
- In the discussion below, f(m,n,k) refers to the luminance value of the pixel

Fourier transforms over multidimensional space

- Continuous space signals
  - Defined for all real values
  - Continuous space FT (CSFT)
- Discrete space signals
  - Defined only on integers
  - Discrete space FT (DSFT)

- Delta function
- Linear Time/shift Invariance
- Convolution

Continuous Space Signals

- K-D Space Signals $\psi(\mathbf{x}), \mathbf{x} = [x_1, x_2, \ldots, x_K] \in R^k$

- Convolution $\psi(\mathbf{x}) * h(\mathbf{x}) = \int_{R^k} \psi(\mathbf{x}-\mathbf{y}) h(\mathbf{y}) d\mathbf{y}$

- Example function
  - Delta function

$$\delta(\mathbf{x}) = \begin{cases} \infty, & \mathbf{x} = \mathbf{0}, \\ 0, & \text{otherwise}, \end{cases} \quad \text{and} \quad \int_{\mathcal{R}^K} \delta(\mathbf{x})\, d\mathbf{x} = 1.$$

$$\psi(\mathbf{x}) * \delta(\mathbf{x} - \mathbf{x}_0) = \psi(\mathbf{x} - \mathbf{x}_0),$$

Continuous Space Fourier Transform (CSFT)

- Forward transform $$\Psi_c(\mathbf{f}) = \int_{R^k} \psi(\mathbf{x}) \exp(-j2\pi \mathbf{f}^T \mathbf{x}) d\mathbf{x}$$

- Inverse transform $$\psi(\mathbf{x}) = \int_{R^k} \Psi_c(\mathbf{f}) \exp(j2\pi \mathbf{f}^T \mathbf{x}) d\mathbf{f}$$

- The transform can be done separately in each dimension!
- If the signal is separable, then CSFT of is the product of the 1D CSFT of each one dimensional signal!

- Convolution theorem

$$\psi(\mathbf{x}) * h(\mathbf{x}) \Leftrightarrow \Psi_c(\mathbf{f}) H_c(\mathbf{f})$$
$$\psi(\mathbf{x}) h(\mathbf{x}) \Leftrightarrow \Psi_c(\mathbf{f}) * H_c(\mathbf{f})$$

Continuous Space Systems

- General system over K-D continuous space

$$\phi(\mathbf{x}) = T(\psi(\mathbf{x})), \mathbf{x} \in R^k$$

- Linear and Space-Shift Invariant (LSI) System

$$\alpha_1\phi_1(\mathbf{x}) + \alpha_2\phi_2(\mathbf{x}) = T(\alpha_1\psi_1(\mathbf{x}) + \alpha_2\psi_2(\mathbf{x}))$$
$$T(\psi(\mathbf{x}+\mathbf{x}_0)) = \phi(\mathbf{x}+\mathbf{x}_0)$$

- LSI systems can be completely described by its impulse response

$$h(\mathbf{x}) = T(\delta(\mathbf{x}))$$
$$\phi(\mathbf{x}) = \psi(\mathbf{x}) * h(\mathbf{x}) \Leftrightarrow \Phi_c(\mathbf{f}) = \Psi_c(\mathbf{f})H_c(\mathbf{f})$$

Discrete Space Signals

- K-D Space Signals $\psi(\mathbf{n}), \mathbf{n} = [n_1, n_2, \ldots, n_K] \in Z^K$
- Convolution $\psi(\mathbf{n}) * h(\mathbf{n}) = \sum_{\mathbf{m} \in Z^K} \psi(\mathbf{n} - \mathbf{m}) h(\mathbf{m})$
- Example function
  - Delta function

$$\delta(\mathbf{n}) = \begin{cases} 1, & \mathbf{n} = \mathbf{0}, \\ 0, & \text{otherwise}. \end{cases}$$

Discrete Space Fourier Transform (DSFT)

- Forward transform

$$\Psi_d(\mathbf{f}) = \sum_{\mathbf{n}\in R^K} \psi(\mathbf{n}) \exp(-j2\pi \mathbf{f}^T \mathbf{n})$$

$\Psi_d(\mathbf{f})$ is periodic in each dimension with period of 1

Fundamental period: $I^K = \{\mathbf{f}, f_k \in (-1/2, 1/2)\}$

- Inverse transform

$$\psi(\mathbf{n}) = \int_{\mathbf{f}\in I^K} \Psi_d(\mathbf{f}) \exp(j2\pi \mathbf{f}^T \mathbf{n}) d\mathbf{f}$$

- Convolution theorem

$$\psi(\mathbf{n}) * h(\mathbf{n}) \Leftrightarrow \Psi_d(\mathbf{f}) H_d(\mathbf{f})$$

$$\psi(\mathbf{n}) h(\mathbf{n}) \Leftrightarrow \Psi_d(\mathbf{f}) * H_d(\mathbf{f})$$

DSFT of Digital Video

- Forward transform

$$\Psi_d(f_x, f_y, f_t) = \sum_{m,n,k} \psi(m,n,k) \exp(-j2\pi(f_x m + f_y n + f_t k))$$

$\Psi_d(f_x, f_y, f_t)$ is periodic in each dimension with period of 1

Fundamental period: $I^K = \left\{ \mathbf{f}, f_k \in (-1/2, 1/2) \right\}$

- Separability of the transform: transform along each column, then along each row, then along the time axis for each pixel location

$$\Psi_d(f_x, f_y, f_t) = \mathbb{F}_{1,k}\left\{ \mathbb{F}_{1,n}\left\{ \mathbb{F}_{1,m}\left\{ \psi(m;n,k) \right\}(f_x) \right\}(f_y) \right\}(f_t)$$

- If the signal is separable, 3D DSFT is the product of 3 1D DSFT

$$\psi(m,n,k) = \psi_1(m)\psi_2(n)\psi_3(k) \longleftrightarrow \Psi_d(f_x, f_y, f_t) = \Psi_1(f_x)\Psi_2(f_y)\Psi_3(f_t)$$

Frequency domain characterization of video signals

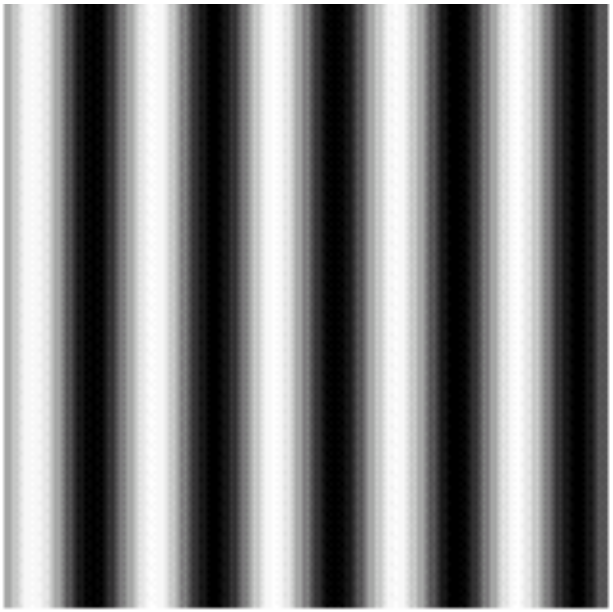
- What is spatial frequency
- What is temporal frequency
- Temporal frequency caused by motion

Spatial Frequency

- Spatial frequency measures how fast the image intensity changes in the image plane
- Spatial frequency can be completely characterized by the variation frequencies in two orthogonal directions (e.g horizontal and vertical)
  - f_x: cycles/horizontal unit distance
  - f_y : cycles/vertical unit distance
- It can also be specified by magnitude and angle of change

$$f_m = \sqrt{f_x^2 + f_y^2}, \quad \phi = \mathrm{atan}(f_y / f_x)$$

Illustration of Spatial Frequency



$f(x,y)=\sin(10\pi x)$

$f_x=5, f_y=0, f_m=5, \phi=0$

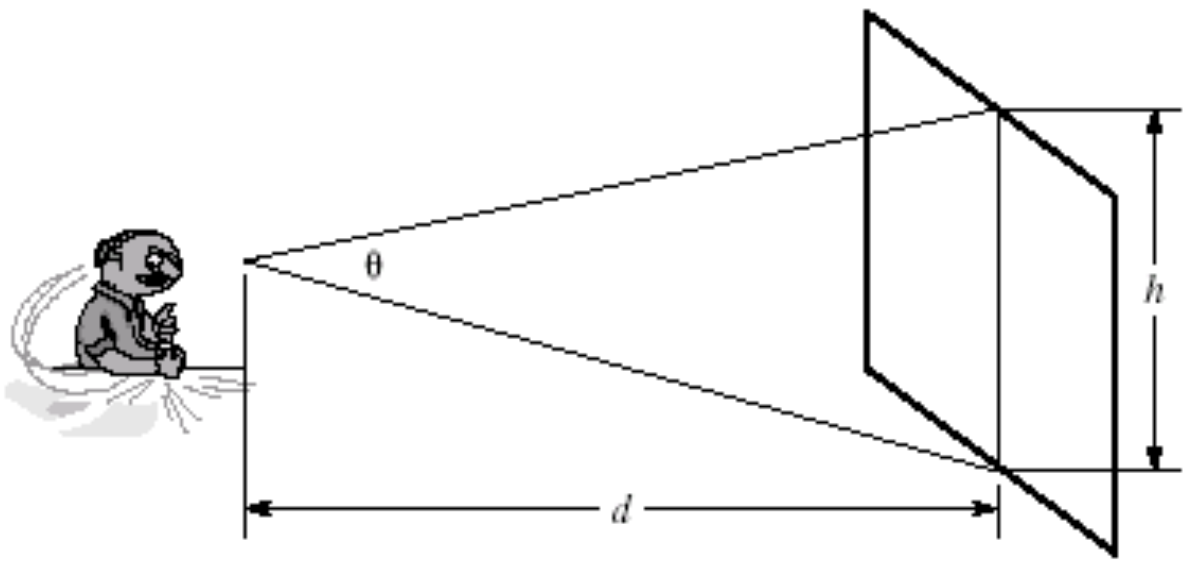
$f(x,y)=\sin(10\pi x-20\pi y)$

$f_x=5, f_y=-10, f_m=\sqrt{125}, \phi=\mathrm{atan}(-2)$

Freq. measued with image-width=image-height=unit-length

Angular Frequency

Perceived spatial frequency (cycles/viewing-angle or cpd) depends on viewing distance



$$\theta = 2\arctan(h/2d)(\text{radian}) \approx 2\text{h}/2\text{d}(\text{radian}) = \frac{180}{\pi}\frac{h}{d}(\text{degree})$$

$$\text{f}_\theta = \frac{\text{f}_s}{\theta} = \frac{\pi}{180}\frac{d}{h}\text{f}_s(\text{cycle/degree})$$

Temporal Frequency

- Temporal frequency measures temporal variation (cycles/s)
- In a video, the *temporal frequency* is spatially varying: each point in space has its own temporal frequency
- Non-zero temporal frequency can be caused by camera or object motion

- Start simple: single object with constant velocity

Temporal Frequency caused by Linear Motion

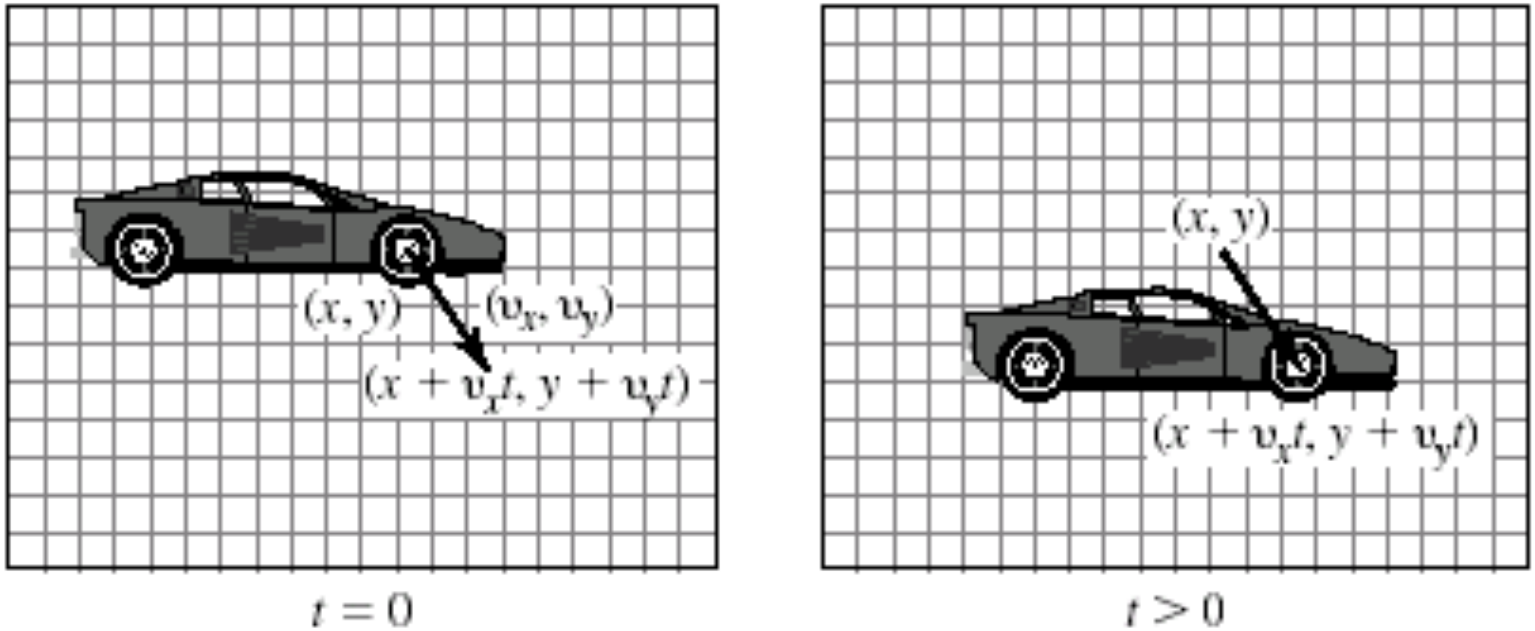


Figure 2.3 Illustration of the constant intensity assumption under motion. Every point (x, y) at $t = 0$ is shifted by $(v_x t, v_y t)$ to $(x + v_x t, y + v_y t)$ at time t, without change in color or intensity. Alternatively, a point (x, y) at time t corresponds to a point $(x - v_x t, y - v_y t)$ at time zero.

Relation between Motion, Spatial and Temporal Frequency

Consider an object moving with speed (v_x, v_y). Assume the image pattern at $t = 0$ is $\psi_0(x, y)$, the image pattern at time t is

$$\psi(x, y, t) = \psi_0(x - v_x t, y - v_y t)$$

$$\Leftrightarrow$$

$$\Psi(f_x, f_y, f_t) = \Psi_0(f_x, f_y)\delta(f_t + v_x f_x + v_y f_y)$$

Relation between motion, spatial, and temporal frequency:

$$f_t = -(v_x f_x + v_y f_y)$$

See Proof in Textbook [Wang2002]

The temporal frequency of the image of a moving object depends on motion as well as the spatial frequency of the object.

Example: A plane with vertical bar pattern (fy=0), moving vertically (vx=0), causes no temporal change (ft=0); But moving horizontally (vx!=0), it causes fastest temporal change (ft= -fx*vx)

Illustration of the Relation

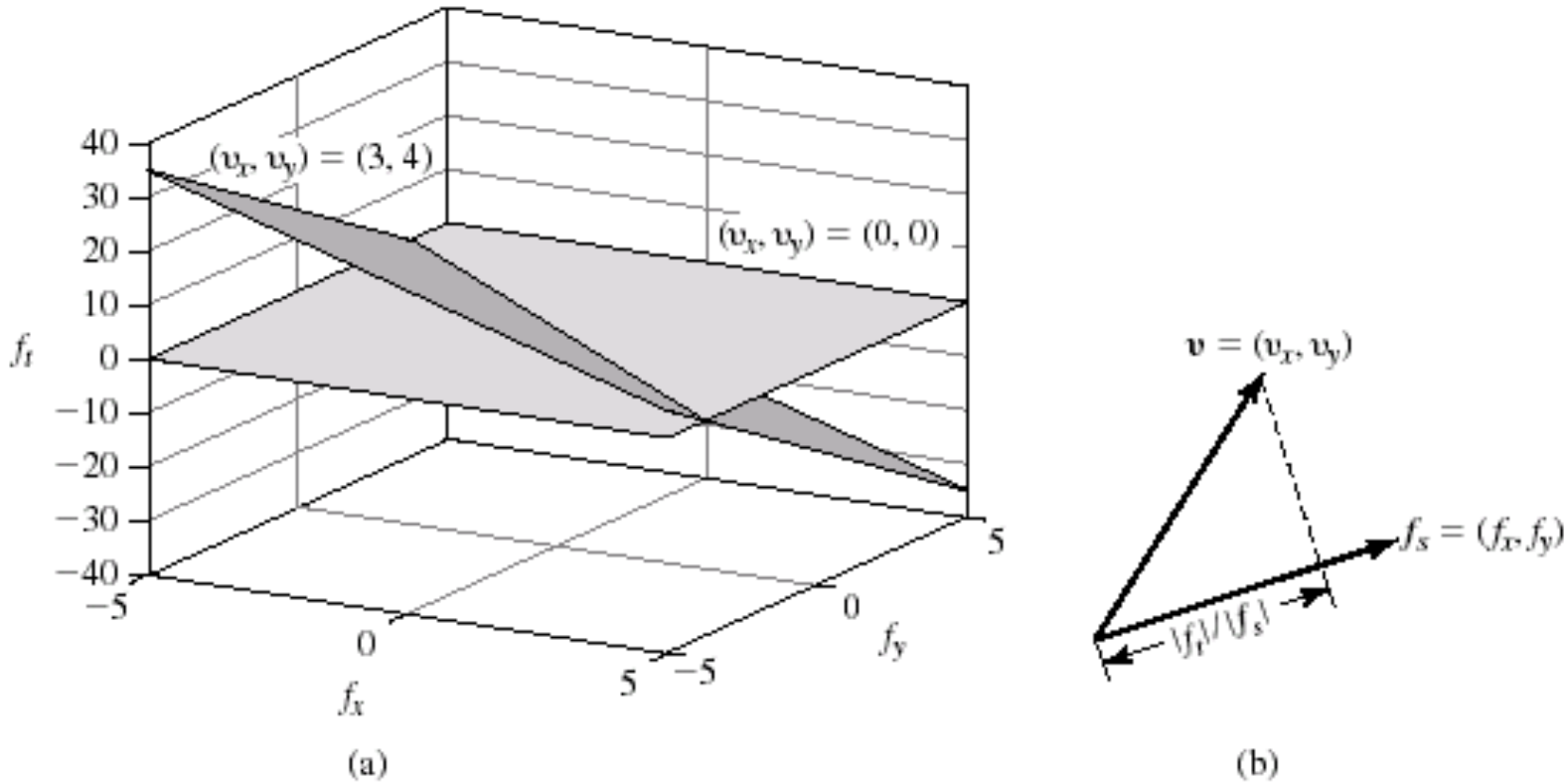


Figure 2.4 Relation between spatial and temporal frequencies under linear motions. (a) The spatiotemporal frequency plane in the (f_x, f_y, f_t) space, corresponding to two different velocity vectors; (b) the temporal frequencies is equal to the projection of the velocity onto the spatial gradient.

If the motion direction and spatial frequency are orthogonal, there will be no apparent motion, or ft= 0.

Frequency Response of the HVS (Human Sensitivity to Different Frequency Components)

- Temporal frequency response and flicker
- Spatial frequency response
- Spatio-temporal response
- Smooth pursuit eye movement

Spatial Contrast Sensitivity Function

- Display the following signal (vertical sinusoidal bars)

$$\psi(x,y,t) = B(1 + m\cos 2\pi f x)$$

- B brightness, f=spatial frequency, m modulation level
- Given f, what is minimum modulation level at which sinusoidal grating is visible?
- $1/m_{min}$ at a given frequency f is the *spatial contrast sensitivity at f*
- Contrast sensitivity function is also known as the Modulation Transfer Function (MTF) of the human eye

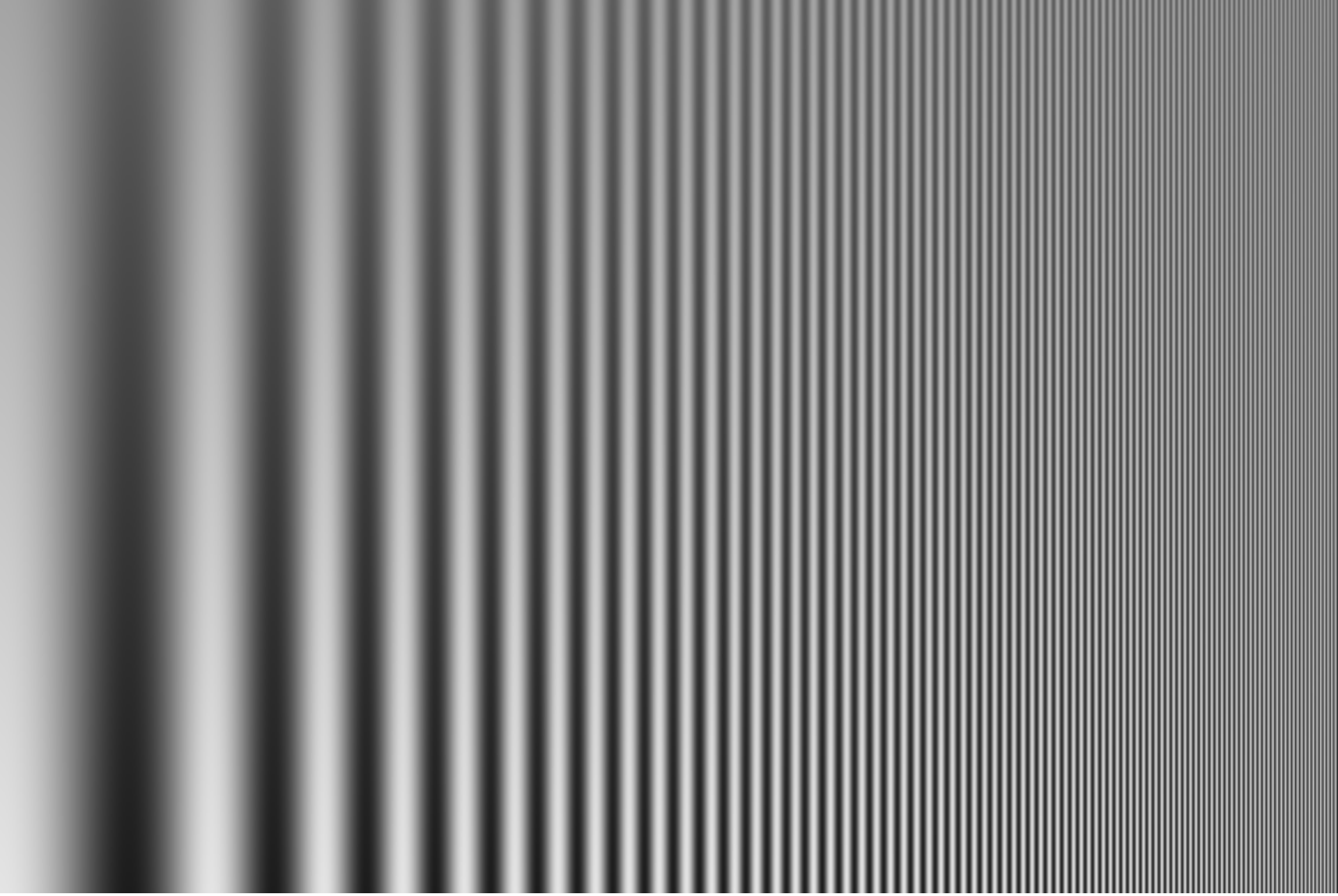


Image provided by Amy Reibman

Spatial Frequency Response of HVS

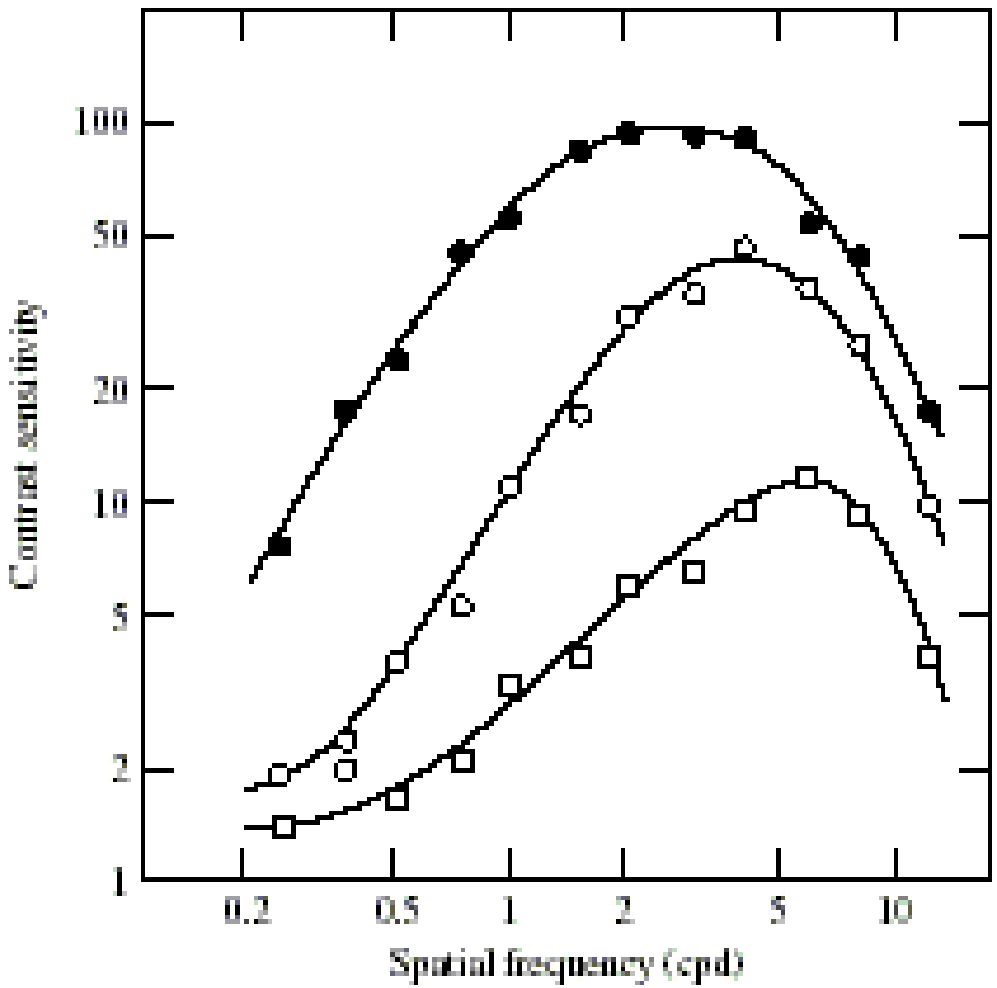


Figure 2.6 The spatial frequency response of the HVS, obtained by a visual experiment. The three curves result from different stabilization settings used to remove the effect of saccadic eye movements. Filled circles were obtained under normal, unstablized conditions; open squares, with optimal gain setting for stabilization; open circles, with the gain changed about 5 percent. Reprinted from D. H. Kelly, Motion and vision. I. Stabilized images of stationary gratings, *J. Opt. Soc. Am.* (1979), 69: 1266–74, by permission of the Optical Society of America.

HVS is most sensitive around 3-5 cpd, and is sensitive up to 30 cpd.

Temporal Contrast Sensitivity Function

- Display the following signal

$$\psi(t,x,y) = B(1 + m\cos 2\pi ft)$$

- B brightness, f =temporal frequency, m modulation level
- What is minimum modulation level at which sinusoidal grating is visible?
- $1/m_{min}$ at a given frequency f is the *temporal contrast sensitivity at f*

Temporal Frequency Response of HVS (depends on mean display brightness)

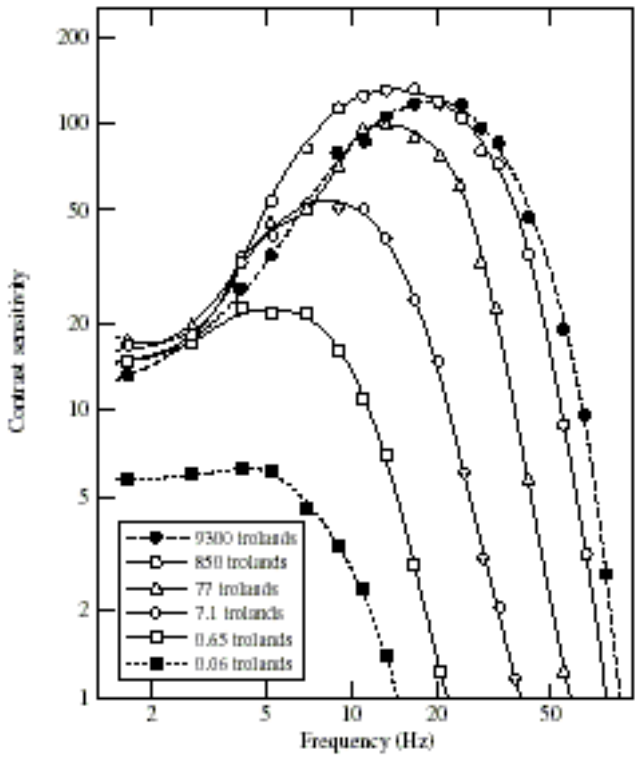


Figure 2.5 The temporal frequency response of the HVS obtained by a visual experiment. Different curves represent the responses obtained with different mean brightness levels, B, measured in trolands. The horizontal axis represents the flicker frequency f, measured in Hz. Reprinted from D. H. Kelly, Visual responses to time-dependent stimuli. I. Amplitude sensitivity measurements, *J. Opt. Soc. Am.* (1961) 51:422–29, by permission of the Optical Society of America.

Critical flicker frequency: The lowest frame rate at which the eye does not perceive flicker.

Provides guideline for determining the frame rate when designing a video system.

Critical flicker frequency depends on the mean brightness of the display:

60 Hz is typically sufficient for watching TV.

Watching a movie needs lower frame rate than TV because movie theater uses less bright display

HVS is most sensitive around 10-20 Hz, and is sensitive up to 75 Hz.

Spatio-Temporal Contrast Sensitivity

- Repeat the previous experiment using the following signal, determine minimal m for each f_x and f_y

$$\psi\left(x,y,t\right)=B\left(1+m\cos(2\pi f_x x)\cos(2\pi f_t t)\right)$$

Spatiotemporal Response

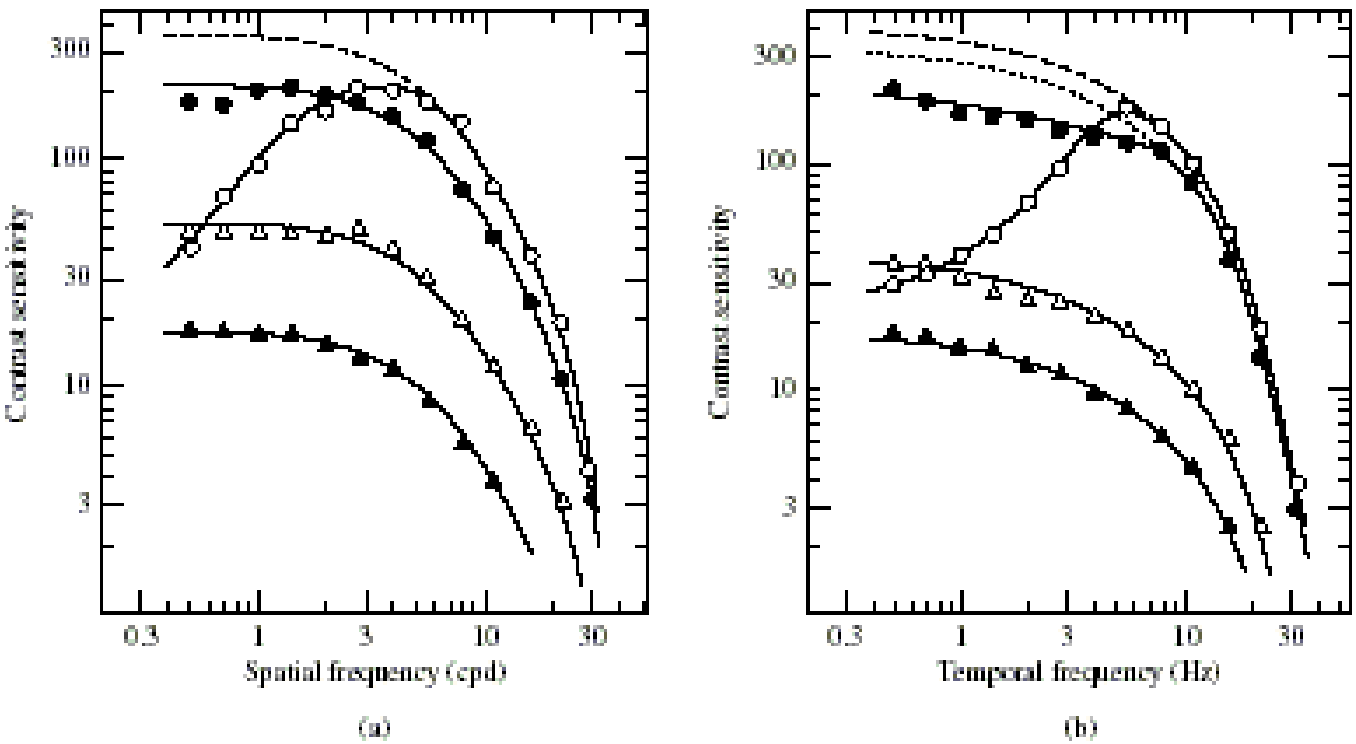


Figure 2.7 Spatiotemporal frequency response of the HVS. (a) Spatial frequency responses for different temporal frequencies of 1 Hz (open circles), 6 Hz (filled circles), 16 Hz (open triangles), and 22 Hz (filled triangles). (b) Temporal frequency responses for different spatial frequencies of 0.5 cpd (open circles), 4 cpd (filled circles), 16 cpd (open triangles), and 22 cpd (filled triangles). Reprinted from J. G. Robson, Spatial and temporal contrast sensitivity functions of the visual systems, *J. Opt. Soc. Am.* (1966), 56:1141–42, by permission of the Optical Society of America.

The reciprocal relation between spatial and temporal sensitivity was used in TV system design:

Interlaced scan provides tradeoff between spatial and temporal resolution.

Smooth Pursuit Eye Movement

- Smooth Pursuit: the eye tracks moving objects
- Net effect: reduce the velocity of moving objects on the retinal plane, so that the eye can perceive much higher raw temporal frequencies than indicated by the temporal frequency response.

Temporal frequency caused by object motion when the object is moving at (v_x, v_y):

$$f_t = -(v_x f_x + v_y f_y)$$

Observed temporal frequency at the retina when the eye is moving at $(\tilde{v}_x, \tilde{v}_y)$:

$$\tilde{f}_t = f_t + (\tilde{v}_x f_x + \tilde{v}_y f_y)$$

$$\tilde{f}_t = 0 \text{ if } \tilde{v}_x = v_x, \tilde{v}_y = v_y$$

No SPEM

SPEM
2 deg/s

SPEM
10 deg/s

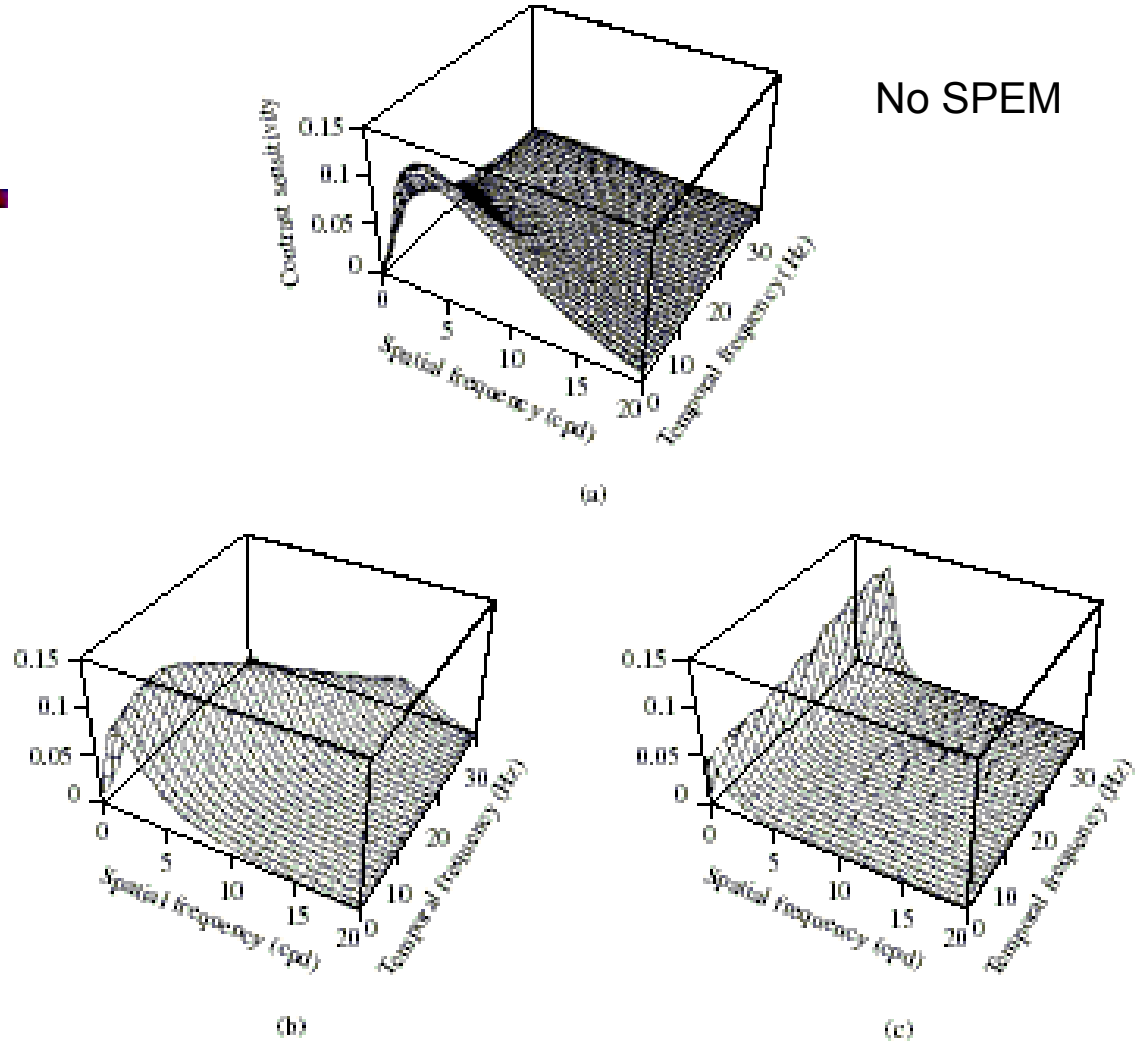


Figure 2.8 Spatiotemporal response of the HVS under smooth pursuit eye movements: (a) without smooth pursuit eye movement; (b) with eye velocity of 2 deg/s; (c) with eye velocity of 10 deg/s. Reprinted from Girod, B. "Motion compensation: visual aspects, accuracy, and fundamental limits." In Sezan, M. I., and R. L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing*. Boston: Kluwer Academic Publishers, 1993, 126–52, by permission of Kluwer Academic Publishers.

Designing Video Camera and Display Based on Frequency Responses of HVS

- Frame rate:
  - >=60 Hz
  - Even higher to account for smooth pursuit eye movement (120 Hz)
- Frame size:
  - Should portrait at least 30 cpd in angular frequency
  - Depends on the display monitor size, viewing distance
  - Example 1 (watching TV), 1 m wide screen, viewed from 3m away
    - Horizontal view angle = 180 * 1/ (3.14*3) = 19 (degree)
    - To display up to 30 cpd, require fs=60 * 19= 1200 pixels/width
  - Example 2 (using computer), 1 ft wide screen, viewed from 1 ft
    - Horizontal view angle = 180 * 1/ (3.14*1) = 57 (degree)
    - To display up to 30 cpd, require fs=60 * 57 ~= 3600 pixels/width
      - Apple Retina Display!

$$\theta = 2\arctan(h/2d)(\text{radian}) \approx 2\text{h}/2\text{d}(\text{radian}) = \frac{180}{\pi}\frac{h}{d}(\text{degree})$$

$$\text{f}_\theta = \frac{\text{f}_s}{\theta} = \frac{\pi}{180}\frac{d}{h}\text{f}_s(\text{cycle/degree})$$

Video Sampling – A Brief Discussion

- Review of Nyquist sampling theorem in 1-D
- Extension to multi-dimensions
- Prefilter in video cameras
- Interpolation filter in video displays

Nyquist Sampling Theorem in 1-D

- Given a band-limited signal with maximum frequency f_{max}, it can be sampled with a sampling rate $f_s >= 2 f_{max}$. The original continuous signal can be reconstructed (interpolated) from the samples exactly, by using an ideal low pass filter with cut-off frequency at $f_s/2$.
- Practical interpolation filters: replication (sample-and-hold, 0th order), linear interpolation (1st order), cubic-spline (2nd order)
- Given the maximally feasible sampling rate f_s, the original signal should be bandlimited to $f_s/2$, to avoid aliasing. The desired prefilter is an ideal low-pass filter with cut-off frequency at $f_s/2$.
- Prefilter design: Trade-off between aliasing and loss of high frequency

Extension to Multi-dimensions

- If the sampling grid is aligned in each dimension (rectangular for image, cubic for video) and one performs sampling in each dimension separately, the extension is straightforward:
  - Requirement: $f_{s,i} >= 2 f_{max,i}$
  - Interpolation/pre-filter: ideal low-pass in each dimension
- If the sampling grid is an arbitrary lattice, the support region of the signal spectrum must be limited within the Voronoi region of the reciprocal of the sampling lattice
  - See Chapters 3 and 4 of [Wang, 2004] for sampling and sampling-rate conversion for K-D signals and for video in particular.
  - Interlaced scan uses a non-rectangular lattice in the vertical-temporal plane.

Aliasing in 2D

Example provided by Amy Reibman

Pictures off the web

Example provided by Amy Reibman

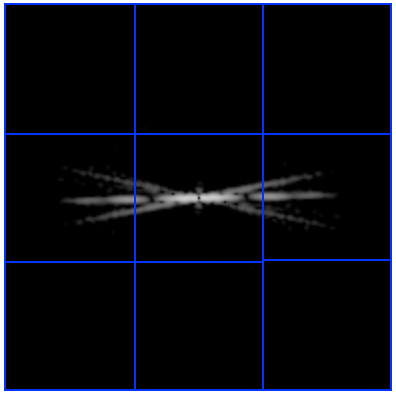
Fourier examples of aliasing: jaggies

Example provided by Amy Reibman

Fourier examples of aliasing: jaggies

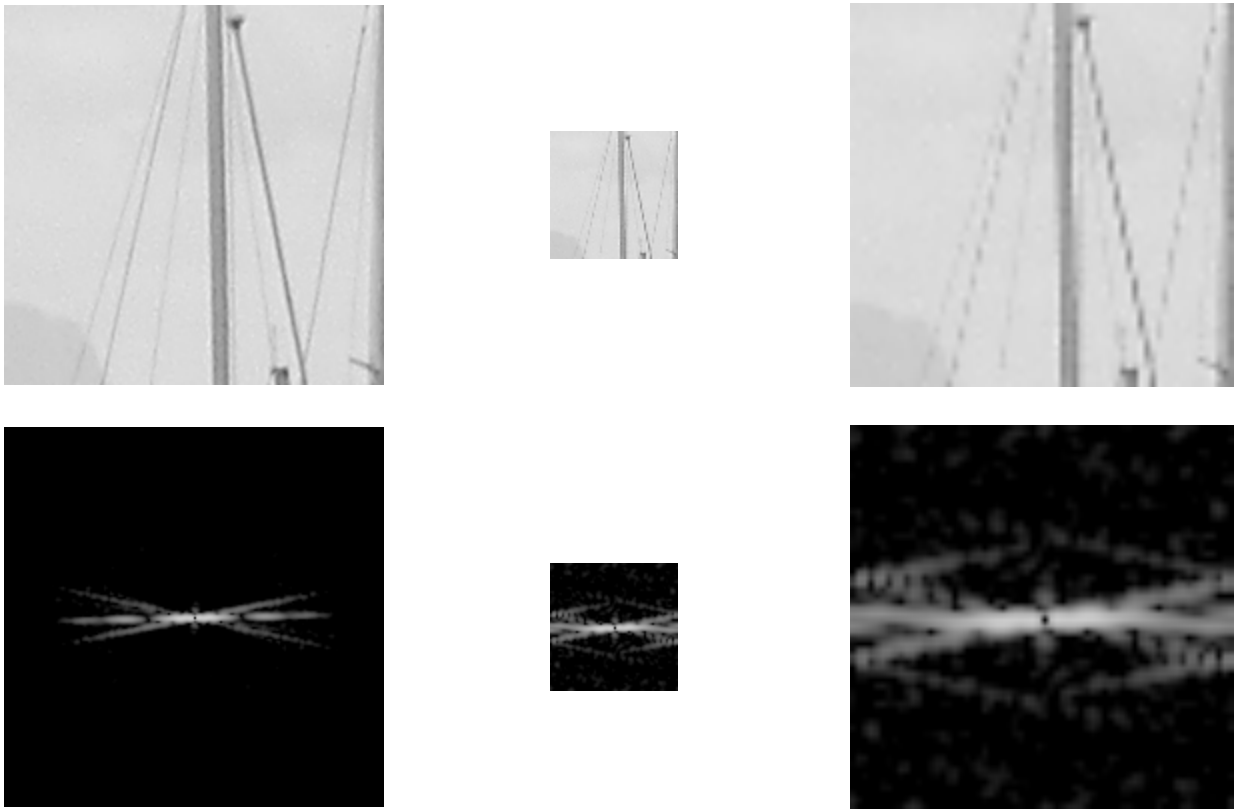
Example provided by Amy Reibman

Fourier examples of aliasing: jaggies



Example provided by Amy Reibman

Fourier examples of aliasing: jaggies



Example provided by Amy Reibman

Aliasing in Video

- Aliasing makes faster moving objects appear slower and moving in different directions!
- Example: Fast moving wheels appear turning backward
- Other example:
- http://www.red.com/learn/red-101/cinema-temporal-aliasing

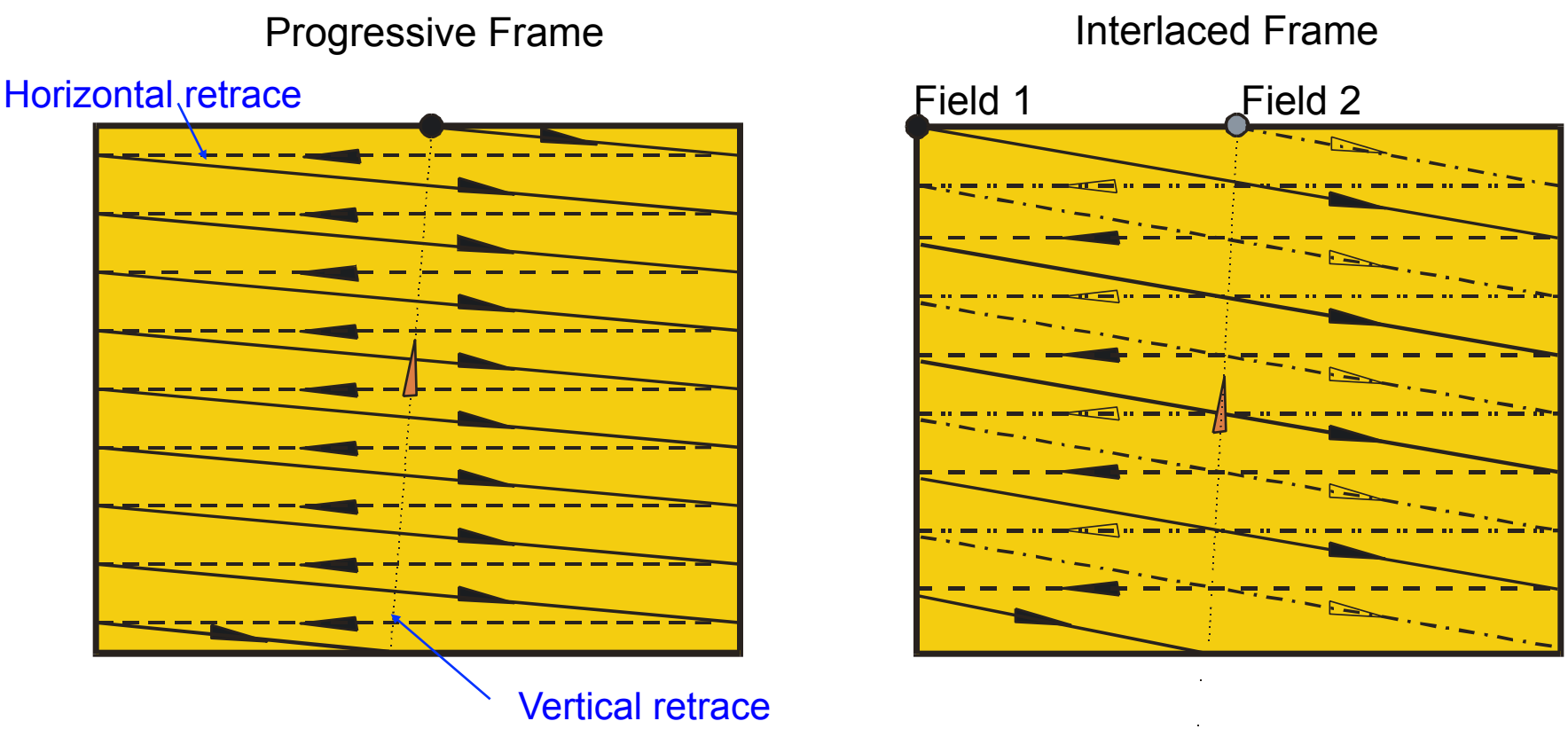
Video Cameras

- Sampling mechanism
  - All perform sampling in time (characterized by frame rate=frames/s)
  - Old film cameras capture continuous frames on film
  - Old analog video cameras sample in vertical but not horizontal direction, using either progressive or interlaced scan.
  - Digital cameras sample in both horizontal and vertical direction, yielding pixels with discrete 3-D coordinates
- How to determine necessary frame rate, line rate, and pixels/line?
  - Depending on display size, viewing distance, maximum frequency in the underlying signal, the maximum frequency that the HVS can detect (visual freq. thresholds), as well as technical feasibility and cost
  - If technically feasible, should sample at twice of visual freq. threshold!

Necessary Sampling Rates

- Frame rate (frames/sec):
  - Human eye can see up to 60 Hz
  - Even higher to account for smooth pursuit eye movement
  - Fast moving objects can lead to temporal freq. >60 cycles/s
  - Ideally video should be captured at >=120 Hz
- Line rate (lines/frame):
  - Should portrait at least 30 cpd in angular frequency, requiring sampling at >=60 samples/viewing degree! (Retina display is defined as >=53, mostly >70)
  - Depends on the display monitor size, viewing distance
- Pixels/line:
  - Depends on the desired aspect ratio
  - Pixels/line = line rate * width/height
  - Wide aspect ratio (16:9, HDTV) is preferred over standard (4:3, SDTV, analog TV)

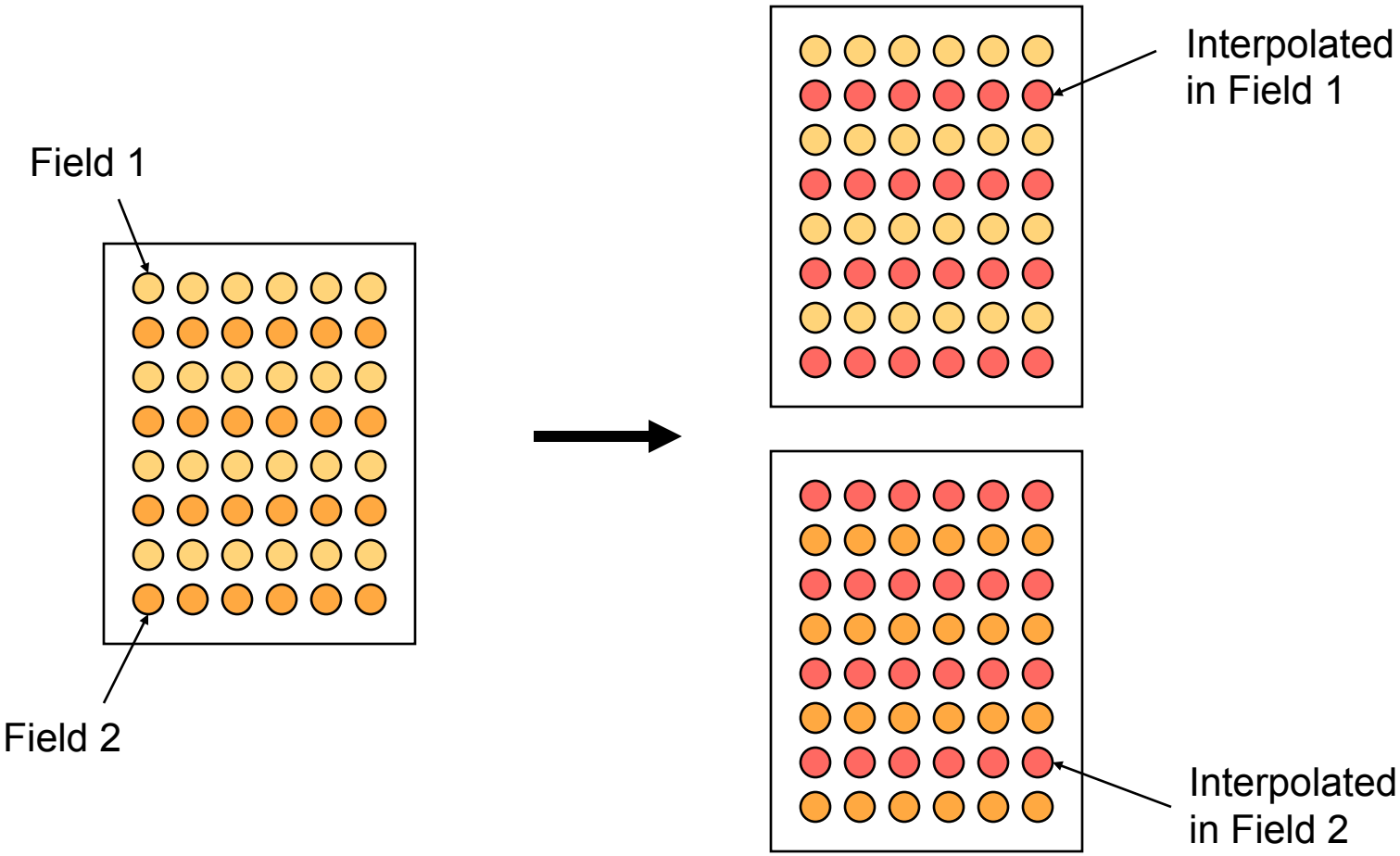
Progressive vs. Interlaced Scan



Why Interlacing?

- Interlaced scan is developed to provide a trade-off between temporal and vertical resolution, for the affordable line rate when analog TV is developed.
- Progressive: 30 frames/sec, 480 lines/frame, low temporal resolution, high vertical resolution
- Interlaced: 60 fields/sec, 240 lines/field, high temporal resolution, low vertical resolution
  - When there is no motion between two adjacent field times, the two fields combine to get 480 lines/frame
  - Where there is high motion, the HVS is less sensitive to spatial detail
  - A good engineering trade-off
  - Artifacts: when vertical lines moves horizontally, you will see jagged lines
- Initial digital video format = digitized analog video
  - Interlaced analog video -> interlaced digital video
  - Deinterlacing problem
- Current digital video format
  - Current technology enables 60 frames/sec or higher and still maintaining high spatial resolution, no need for interlaced format

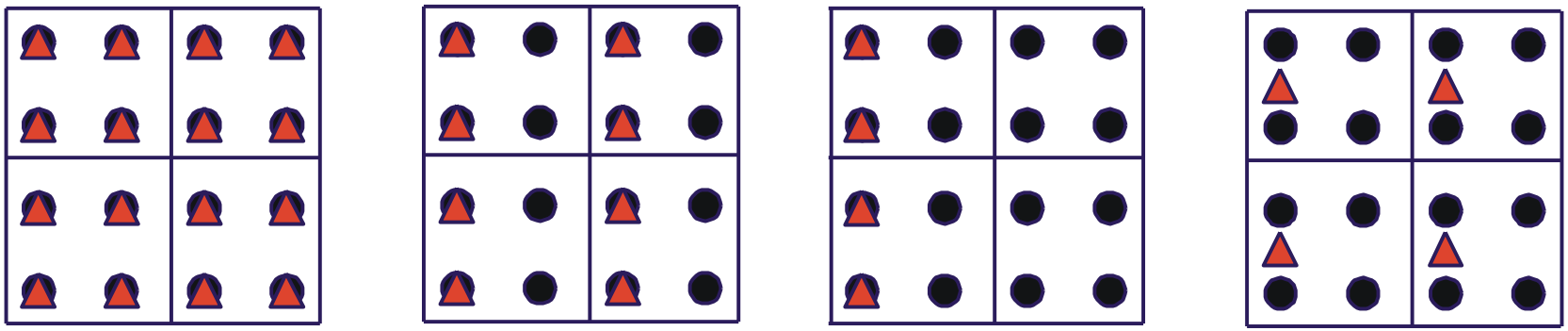
Deinterlacing



Example Video Formats

- Old analog TV system:
  - NTSC: 60 fields/sec (60i), 525 lines/frame (only 480 active lines)
  - PAL/SECAM: 50 fields/sec (50i), 625 lines/frame (only 576 active lines)
- BT601 Digital video (digitized version of NTSC/PAL)
  - NTSC->60i, 720x480, data rate=720*480*30*24=249Mbps
  - PAL/SECAM-> 50i, 720x576, data rate=720*576*50*24=249Mbps
- SD video format
  - 4CIF: 60 frames/sec, 720x480/frame, data rate=720*480*60*24=498Mbps
- HDTV
  - 1980p: 60 frames/sec, 1920x1080/frame, data rate=3 Gbps
- Ultra HD:
  - 4K: 60 frames/sec, 4096x2180/frame, data rate=12.9 Gbps
- Above assumes the three color components (e.g. YCbCr) are represented with the same resolution color (4:4:4 format)
- When Cb,Cr are represented with ½ resolution in both horizontal and vertical (4:2:0 format), each pixel takes 12 bits, reducing data rate by half.

Chrominance Subsampling Formats



4:4:4
For every 2x2 Y Pixels
4 Cb & 4 Cr Pixel
(No subsampling)

4:2:2
For every 2x2 Y Pixels
2 Cb & 2 Cr Pixel
(Subsampling by 2:1 horizontally only)

4:1:1
For every 4x1 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 4:1 horizontally only)

4:2:0
For every 2x2 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 2:1 both horizontally and vertically)

● Y Pixel ▲ Cb and Cr Pixel

Some Standard Digital Video Formats

Video Format	Y Size	Color Sampling	Frame Rate (Hz)	Raw Data Rate (Mbps)
HDTV Over air. cable, satellite, MPEG2 video, 20-45 Mbps				
SMPTT296M	1280x720	4:2:0	24P/30P/60P	265/332/664
SMPTE295M	1920x1080	4:2:0	24P/30P/60I	597/746/746
Video production, MPEG2, 15-50 Mbps				
CCIR601	720x480/576	4:4:4	60I/50I	249
CCIR601	720x480/576	4:2:2	60I/50I	166
High quality video distribution (DVD, SDTV), MPEG2, 4-10 Mbps				
CCIR601	720x480/576	4:2:0	60I/50I	124
Intermediate quality video distribution (VCD, WWW), MPEG1, 1.5 Mbps				
SIF	352x240/288	4:2:0	30P/25P	30
Video conferencing over ISDN/Internet, H.261/H.263, 128-384 Kbps				
CIF	352x288	4:2:0	30P	37
Video telephony over wired/wireless modem, H.263, 20-64 Kbps				
QCIF	176x144	4:2:0	30P	9.1

Prefilter in Typical Cameras

- Temporal prefilter: controlled by exposure time. the value read out at any frame is the average of the sensed signal over the exposure time (a simple averaging or box filter)

- Spatial prefilter: the value read out at any pixel is a weighted integration of the signal in a small window surrounding it, called the aperture, can be approximated by a box average or a 2-D Gaussian function

Video Display

- The display device presents a digital video on the screen by lighting the spatially-arranged phosphor elements behind the screen with intensity proportional to actual image intensity at different frame times to create the sensation of continuously varying signal in both time and space.
- Size of phosphor elements affects the spatial interpolation filter, decaying time of phosphors affects the temporal interpolation filter.
- The eye further performs lowpass filtering: fuses discrete frames and pixels as continuously varying, if the temporal and spatial sampling rates are sufficiently high.

Detection of Moving Object

- For video surveillance, we often want to detect moving foreground objects (e.g. people) in an otherwise stationary background
- How do we accomplish that?
- Take the difference between corresponding pixels in two successive frames. Pixels with large differences correspond to moving objects.
- Moving object detection → Change detection

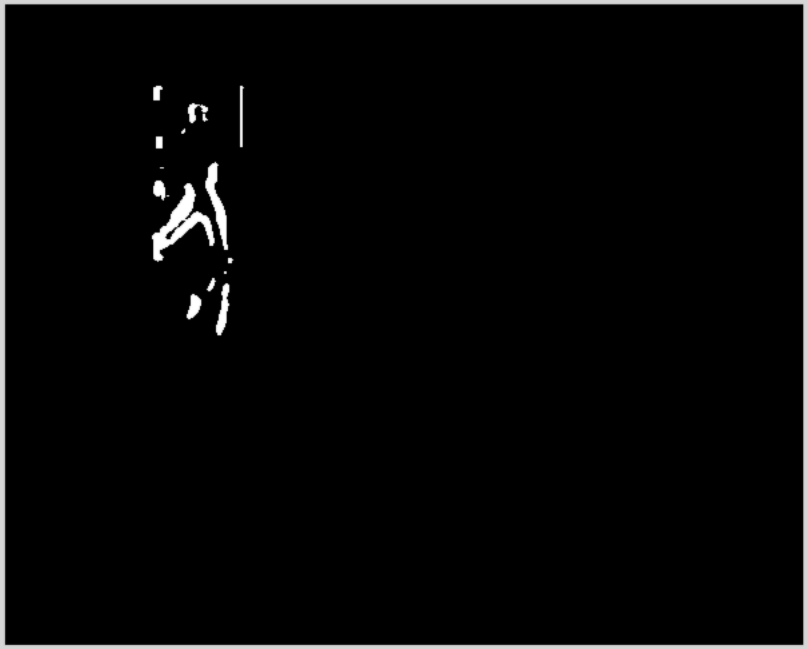
Moving object detection by examining frame difference

- Frame at t: $f(x,y,t)$
- Frame Difference at t: $e(x,y,t)=|f(x,y,t)-f(x,y,t-1)|$
- Thresholding the difference to highlight pixels with large change
- Show example

```
>> img1=imread('frame31.jpg');
>> img2=imread('frame32.jpg');
>> img1=rgb2gray(img1);
>> img2=rgb2gray(img2);
>> img1=int16(img1);
>> img2=int16(img2);
>> diff=abs(img1-img2);

>> figure(1),imshow(img1,[])
>> figure(2),imshow(img2,[])
>> figure(3),imshow(diff,[])
```

```
>> figure(3),imshow(diff,[])
>> max(max(diff))
>> diffT=(diff>20);
>> figure(4),imshow(diffT,[])
>> diffTM=medfilt2(diffT,[5 5]);
>> figure(5),imshow(diffTM,[])
```



Problem with frame difference

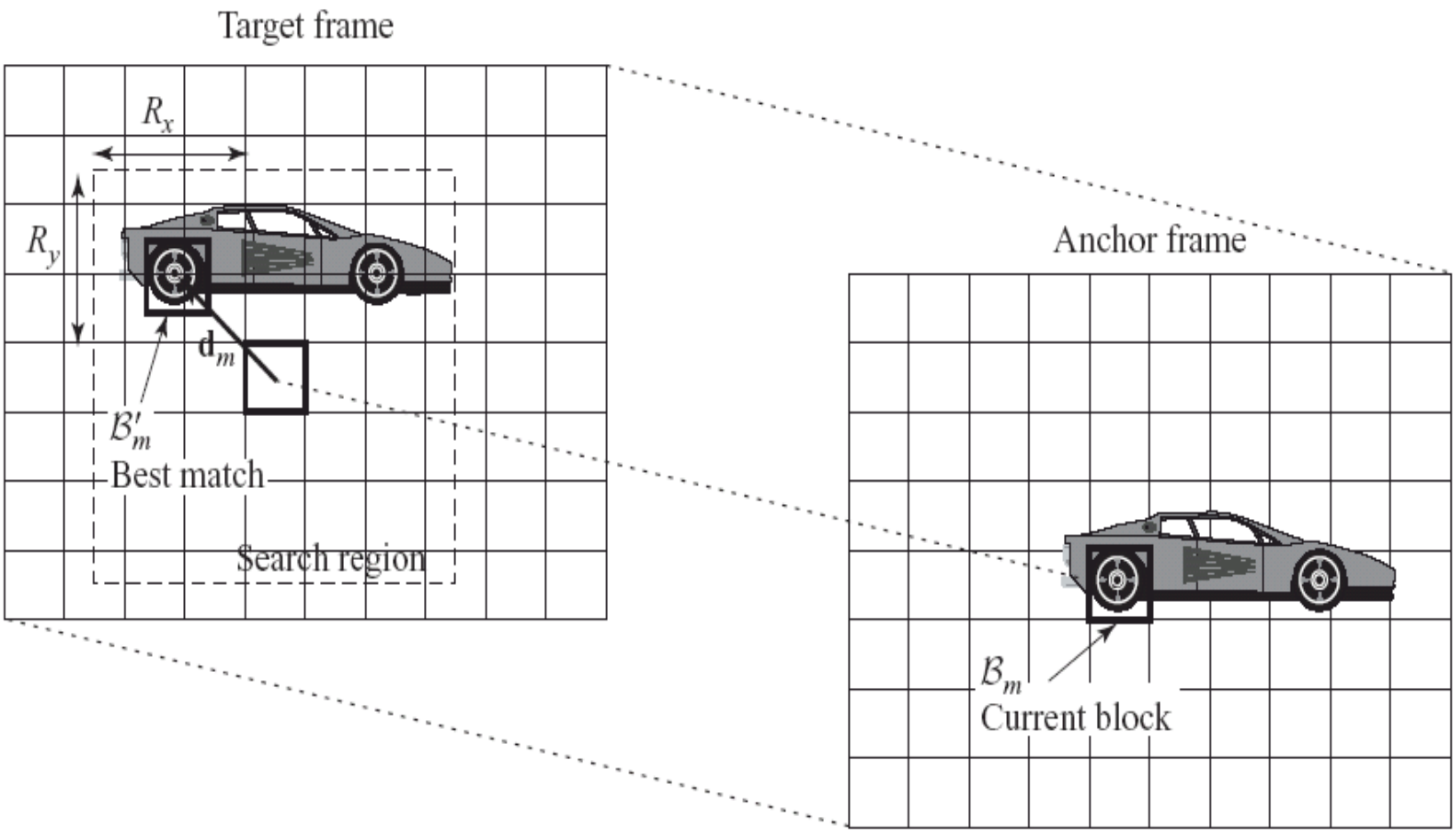
- The background may not be stationary
  - Tree leaf motion
  - Lighting change
  - Camera motion
- More advanced methods can compensate for background changes

Object Tracking

- Suppose you identified a person or an object in one frame, and you want to find how does it move in the subsequent frames.
- How do you do that?
- If you put a bounding box over the person, then the color pattern within the bounding box (template block) should not change much even if the box is moving over time
- We can find how does the box move by searching for a same sized box with similar color pattern in successive frames – Block Matching (also known as template matching)

Block Matching: Illustration

Exhaustive Block Matching Algorithm (EBMA)



How to measure whether a candidate block is a good match with a template block?

- Template Block T(x,y)
- Candidate block C(x,y)
- |T(x,y)-C(x,y)|^p should be small for (x,y) in the block
- SAD (Sum of absolute difference) (p=1, L1 norm)
  - SAD=sum_{(x,y} in the block} |T(x,y)-C(x,y)|
- SSD (Sum of squared difference) (p=2, L2 norm)
  - SSD=sum_{(x,y} in the block} (T(x,y)-C(x,y))^2
- SAD vs SSD
  - SAD requires less computation and is preferred in practice
  - SAD (L1 norm) is also more robust to outliers (small number of pixels with large error)

Block Matching by Exhaustive Search (EBMA): Matlab Script

```
function [xm,ym,matchblock]=EBMA(template,img,x0,y0,Rx, Ry)
%x0,y0 define the location of the top left corner of the template in the previous image
%xm,ym define the matched position of the template in the current image
%R is the search range
%template and img should be saved in integer or float

[H,W]=size(img);
[BH,BW]=size(template);
maxerror=BH*BW*255;
xm=x0, ym=y0;
for (k=max(1,x0-Rx):min(W-BW,x0+Rx))
   for (l=max(1,y0-Ry):min(H-BH,y0+Ry))
      candidate=img(l:l+BH-1,k:k+BW-1);
      error=sum(sum(abs(template-candidate))); %SAD
      if (error<maxerror)
         xm=k;ym=l;matchblock=candidate;
         maxerror=error;
      end
   end
end
```

Example of Object Tracking


```
>> figure(1),imshow(img1,[])
>> figure(2),imshow(img2,[])
>> x0=112,y0=59,x1=175,y1=202
>> Rx=24,Ry=10
>> template=img1(y0:y1,x0:x1);
>> [xm,ym,matchblock]=EBMA(template,img2,x0,y0,Rx,Ry);
>> xm, ym
```

Problems with Block Matching

- The shape of the object may change if the motion is not just a shift
- Different parts of the object may move differently
- More sophisticated algorithms are needed to solve these challenges (to be covered in the next lecture)
- However, when two frames are very close in time (e.g under high frame rate), the movement of most objects are small and simple block matching can work quite well

MATLAB Functions for Video Manipulation

- In MATLAB, you can read in a video stored in several formats using the tool: VideoReader() and several other associated functions. You can play a video using the movie() function. Please familiarize yourself with these tools using the help command.

- VideoReader, VideoWriter, movie, movie2avi.

Sample MATLAB Code for Reading a Video

```
xyloObj = VideoReader('input_video.mpg');                    % Create Object
nFrames = xyloObj.NumberOfFrames;                            % Obtain Frame Number
vidHeight = xyloObj.Height;                                  % Obtain Image Height
vidWidth = xyloObj.Width;                                    % Obtain Image Width
mov(1:nFrames)= struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),'colormap', []);
target_frame = read(xyloObj, 5);                             % Read in 5th Frame
for k = 1 : nFrames
   mov(k).cdata = read(xyloObj, k);                          % Load video into MOV
end
hf = figure;                                          % Create New Display Window
set(hf, 'position', [150 150 vidWidth vidHeight])            % Set window Location
movie(hf, mov, 1, xyloObj.FrameRate);                                  % Display video sequence
```

Provided by Fanyi Duanmu

Sample MATLAB Code for Saving a Video

```
myObj = VideoWriter('newfile.avi');          % Create Object
myObj.FrameRate = 30;                        % Set Frame Rate
open(myObj);
for i = 31:50
        temp = frame(:,:,:,i);
        writeVideo(myObj,temp);              % Write in 20 frames one by one
end
close(myObj);
```

Provided by Fanyi Duanmu

Summary

- Spatio-Temporal frequency:
  - Relation between temporal freq. and motion and spatial freq
  - Perceived freq. depends on viewing distance (Angular frequency)
- Frequency response of HVS
  - How it is evaluated, what are the implications (no need to display frequency components that are higher than what the eye can see!)
  - Interaction between spatial/temporal sensitivities
- Sampling of video
  - Determined by multiple factors: maximum freq. in the video, maximum observable freq by the HVS, engineering limitations
  - Interlaced vs. progressive scan
- Two basic video processing tasks
  - Moving object detection
  - Moving object tracking (Block matching algorithm)

Reading Assignment

- Wang et al, Video Processing, Chap 1-3

Written HW

1. Consider a vertical bar pattern on a TV screen with 200 cycles/screen-width. If the screen width is 1 meter, and the viewer sits at 4 meters away from the screen, what is the perceived angular frequency of this bar pattern? What if the viewer sits at 2 meter? In either case, would the viewer be able to perceive the bar pattern properly?
2. Consider an object that has a flat homogenously textured surface with a sinusoidal pattern with frequency of (fx,fy)=(2,4) cycles/meter, and is moving at a speed of (vx,vy)=(3,2) meter/second. What is the temporal frequency of the object surface at any point? At what moving speed, would the object appear stationary (temporal frequency becomes 0)?
3. For the scene in Prob. 2, suppose the eye tracks the moving object at a speed that is equal to the object speed. What is the perceived temporal frequency? What if the eye focus moves at a fixed speed of (1,1) meter/second?
4. What are the considerations you would use to determine the frame rate and line number and pixels/line when designing a video camera and display system?
5. Why does a computer monitor use a higher temporal refresh rate and much higher spatial resolution than does a typical TV monitor?
6. What are some of the benefit of using interlaced scan, compared to using progressive scan with the same total line number/sec? What are some of the problems?
7. Consider a camera with exposure time of T_e, and a camera aperture size of $T_x X T_y$. Its impulse response can be approximated as a

$$h(x,y,t)=\frac{1}{T_x T_y T_e}, \text{ for } |x|<\frac{T_x}{2}, |y|<\frac{T_y}{2}, t\in(0,T_e); \quad =0 \text{ otherwise}$$

Determine and plot the magnitude of its frequency response. Comment on the effect of the parameters T_x, T_y, T_e.

MATLAB Assignments

1. Write a program that can: i) read in a short video clip (keep up to only, say 30 frames); ii) Compare each two successive frames to extract pixels with significant change (absolute difference greater than a threshold T), set such pixels to white, while keeping other pixels as black; iii) Display the successive thresholded difference images as a movie; iv) Save the successive difference images as an ".avi" file.
2. Read the sample program for template matching given in the lecture note (templatematching( )) and understand how it works. Select two video frames from a video that contain the same object or person with shifted positions. Identify a bounding box for the object in one frame. Write your own program that can help find its corresponding position in another frame. Was the result accurate? If not, give some reasons that may have contributed to the error.
3. (optional) Modify Prob. 2 so that you can continuously track an object in successive frames. You need to manually determine the bounding box containing the interested object in the first frame.