Video Processing and Standards Conversion

Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 3.1

Artificial Intelligence &
Information Analysis Lab

Video Processing and Standards Conversion

- **Multidimensional Signals and Systems**
- Multidimensional Signal Transforms
- Video Denoising
- Video Interpolation

Introduction to multidimensional signal processing

- A multidimensional signal is a function of $M \geq 2$ independent variables.
- Video is a three dimensional (3D) spatiotemporal signal $f(x, y, t)$.
- Such signals can be separated into the following categories:
  - A ***continuous*** multidimensional signal $f(\mathbf{x})$ is a function M independent variables having domain $\mathbf{x} \in \mathbb{R}^M$ and range are $\mathbb{R}$, $f(x) \in \mathbb{R}$.
  - A ***discrete*** multidimensional signal $f(\mathbf{n})$ is a function defined on a set of discrete values (grid) $\mathbf{n} \in \mathbb{Z}^M$, whose range are $f(\mathbf{n}) \in \mathbb{R}$ or the integers $f(\mathbf{n}) \in \mathbb{Z}$ or a subset of $\mathbb{Z}$.

Discrete spatiotemporal systems

- A M-dimensional discrete system T transforms an M-dimensional discrete input signal, $f(\mathbf{n}) \in \mathbb{Z}^M$ to an M-dimensional discrete output signal $g(\mathbf{n})$:

$$g(\mathbf{n}) = T[f(\mathbf{n})].$$

- An M-dimensional system is ***linear***, if :

$$T[a_1 f_1 + a_2 f_2] = T[a_1 f_1] + T[a_2 f_2].$$

- It is called ***shift-invariant***, if :

$$g(\mathbf{n} - \mathbf{m}) = T[f(\mathbf{n} - \mathbf{m})].$$

Discrete spatiotemporal systems

- If an M-dimensional system is linear and shift-invariant, its input-output relation is defined by an M-dimensional convolution by an ***impulse response*** $h(\mathbf{n})$:

$$g(\mathbf{n}) = \sum_{k} f(\mathbf{k})h(\mathbf{n}-\mathbf{k}).$$

- If $h(\mathbf{n})$ is defined only on a finite domain of $\mathbb{Z}^M$, e.g., on $0 \leq n_1 \leq N_1, \dots, 0 \leq n_M < N_M$, the ***Finite Impulse Response*** (***FIR***) system is described by a M-D convolution:

$$g(n_1, \dots, n_M) = \sum_{k_1=0}^{N_1-1} \dots \sum_{k_M=0}^{N_M-1} h(k_1, \dots, k_M) f(n_1 - k_1, \dots, n_M - k_M).$$

Discrete spatiotemporal systems

Multidimensional systems having impulse responses with infinite area of support are called ***Infinite Impulse Response*** (***IIR***) ones.

- The are described by a ***difference equation***:

$$\sum_{k_1} \cdots \sum_{k_M} b(k_1, \ldots, k_M) g(n_1 - k_1, \ldots, n_M - k_M) =$$

$$\sum_{r_1} \cdots \sum_{r_M} a(r_1, \ldots, r_M) f(n_1 - r_1, \ldots, n_M - r_M).$$

- If carefully designed, have the same performance with FIR filters, but fewer coefficient and much less computational complexity.
- If not carefully designed, they may have ***stability*** problems.

Discrete spatiotemporal systems

3D moving average filter having 3D $L_1 \times L_2 \times L_3$ filter window of odd number size: $L_i = 2v_i + 1,\ i = 1,2,3$:

$$g(n_1, n_2, n_3) = \frac{1}{L_1 L_2 L_3} \sum_{k_1=-v_1}^{v_1} \sum_{k_2=-v_2}^{v_2} \sum_{k_3=-v_3}^{v_3} f(n_1 - k_1, n_2 - k_2, n_3 - k_3).$$

- It has impulse response:

$$h(n_1, n_2, n_3) = \frac{1}{L_1 L_2 L_3}, \quad (n_1, n_2, n_3) \in [-v_1, v_1] \times [-v_2, v_2] \times [-v_3, v_3].$$

- It is a 3D linear low-pass FIR filter.

Discrete spatiotemporal systems

3D $L_1 \times L_2 \times L_3$ ***median filter***:

$$g(n_1, n_2, n_3) = \underset{k_1,k_2,k_3}{\mathrm{med}} \{f(n_1 - k_1, n_2 - k_2, n_3 - k_3)\} ,$$

- $(k_1, k_2, k_3) \in [-\nu_1, \nu_1] \times [-\nu_2, \nu_2] \times [-\nu_3, \nu_3]$.
- med is the pixel median value in the local filter window.
- Odd size filter windows can be easily centered around filter center.

Video Processing and Standards Conversion

- Multidimensional Signals and Systems
- **Multidimensional Signal Transforms**
- Video Denoising
- Video Interpolation

Multidimensional/three dimensional $\mathcal{Z}$ transform

- Definition of ***M*-dimensional $\mathcal{Z}$ transform**:

$$F(z_1, \dots, z_M) = \sum_{n_1=-\infty}^{\infty} \cdots \sum_{n_M=-\infty}^{\infty} f(n_1, \dots, n_M) z_1^{-n_1} \dots z_M^{-n_M},$$

- $f(n_1, \dots, n_M)$: a M-dimensional discrete signal
- $z_1, \dots, z_M$: complex variables.
- Inverse M-dimensional $\mathcal{Z}$ transform:

$$f(n_1, \dots, n_M) = \left(\frac{1}{2\pi i}\right)^M \oint_{C_1} \cdots \oint_{C_M} F(z_1, \dots, z_M)\, z_1^{n_1-1} \dots z_M^{n_M-1} dz_1 \cdots dz_M,$$

Multidimensional/three dimensional $\mathcal{Z}$ transform

- The complex contour integrals are defined over the contours $C_i = 1, \dots, M$ lying in the region of convergence of the $\mathcal{Z}$ transform.

- The convolution in the spatial domain $\mathbb{Z}^M$ corresponds to multiplication in the $\mathcal{Z}$ transform domain:

$$g(\mathbf{n}) = \sum_{k} f(\mathbf{k})\, h(\mathbf{n} - \mathbf{k}) \leftrightarrow G(z_1, \dots, z_M) = F(z_1, \dots, z_M) H(z_1, \dots, z_M).$$

Transfer function of multidimensional digital filters

- ***Transfer function*** of an M-dimensional system:

$$H(z_1, \dots, z_M) = \frac{G(z_1, \dots, z_M)}{F(z_1, \dots, z_M)},$$

- $F(z_1, \dots, z_M), G(z_1, \dots, z_M)$: $\mathcal{Z}$ transforms of system input and output signals.
- Transfer function of an FIR digital filter:

$$H(z_1, \dots, z_M) = \sum_{k_1=0}^{N_1-1} \cdots \sum_{k_M=0}^{N_M-1} h(k_1, \dots, k_M)\, {z_1}^{-k_1} \dots {z_M}^{-k_M}.$$

Transfer function of multidimensional digital filters

- Transfer function of a 3D moving average filter :

$$H(z_1, z_2, z_3) = \frac{1}{L_1 L_2 L_3} \sum_{k_1=-v_1}^{v_1} \sum_{k_2=-v_2}^{v_2} \sum_{k_3=-v_3}^{v_3} {z_1}^{-k_1} {z_2}^{-k_2} {z_3}^{-k_3}.$$

- Transfer function of an IIR digital filter is a rational function:

$$H(z_1, \dots, z_M) = \frac{\sum_{r_1} \dots \sum_{r_M} a(r_1, \dots, r_M) {z_1}^{-r_1} \dots {z_M}^{-r_M}}{\sum_{k_1} \dots \sum_{k_M} b(k_1, \dots, k_M) {z_1}^{-k_1} \dots {z_M}^{-k_M}}.$$

- Multidimensional IIR systems may be unstable.
- Stability check of multidimensional IIR filters can be very difficult.

Discrete Spatiotemporal Fourier Transform

- ***3-D discrete spatiotemporal Fourier transform***:

$$F(\omega_1, \omega_2, \omega_3) = \sum_{n_1} \sum_{n_2} \sum_{n_3} f(n_1, n_2, n_3)\, e^{i(\omega_1 n_1 + \omega_2 n_2 + \omega_3 n_3)}.$$

- n_3 is used as discrete time variable (instead of n_t).
- It results from M-dimensional Z transform, when defined on the M unit circles: $|z_1| = 1, \ldots, |z_M| = 1$.
- $\omega_i = \Omega_i T_i, \quad i = 1, \ldots, M$ are ***angular frequencies***, defined on the M unit circles $-\pi \leq \omega_i \leq \pi, i = 1, \ldots, M$.
- $T_i, \quad i = 1, \ldots, M$: sampling intervals.

Discrete Spatiotemporal Fourier Transform

- The 3D discrete spatiotemporal Fourier transform exists, if the signal $f(n_1, n_2, n_3)$ is absolutely integrable:

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_3=-\infty}^{\infty} |f(n_1, n_2, n_3)| = S < \infty.$$

- Inverse 3D discrete spatiotemporal Fourier transform :

$$f(n_1, n_2, n_3) = \frac{1}{8\pi^3} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} F(\omega_1, \omega_2, \omega_3)\, e^{(i\omega_1 n_1 + i\omega_2 n_2 + i\omega_3 n_3)} d\omega_1 d\omega_2 d\omega_3.$$

- The integration contours are the three unit circles $|z_i| = 1, i = 1,2,3$.

Discrete Spatiotemporal Fourier Transform

- Signal convolution in the spatiotemporal domain is equivalent to multiplication in the Fourier transform domain:

$$g(n_1, n_2, n_t) = f(n_1, n_2, n_t) * h(n_1, n_2, n_t) \leftrightarrow$$
$$G(\omega_1, \omega_2, \omega_t) = F(\omega_1, \omega_2, \omega_t)\, H(\omega_1, \omega_2, \omega_t).$$

- ***Frequency response*** of a spatiotemporal filter:

$$H(\omega_1, \omega_2, \omega_t) = \frac{G(\omega_1, \omega_2, \omega_t)}{F(\omega_1, \omega_2, \omega_t)}.$$

- It defines its frequency response characteristics of a 3D filter.

Multidimensional Discrete Fourier transform

Multidimensional Discrete Fourier Transform (***DFT***):

$$F(\mathbf{k}) = \sum_{\mathbf{n}\in R_N} f(\mathbf{n})\exp(-i\mathbf{k}^T(2\pi\mathbf{N}^{-1})\mathbf{n}) \,.$$

- $R_N = \{\mathbf{n}:\ 0 \le n_i \le \ N_i - 1, i = 1, \ldots, M\}$,
- $\mathbf{N} = \mathrm{diag}(N_1, \ldots, N_M)$.
- Inverse multidimensional DFT has the form:

$$f(\mathbf{n}) = \frac{1}{|\det \mathbf{N}|} \sum_{\mathbf{k}\in R_N} F(\mathbf{k})\exp(i\mathbf{k}^T(2\pi\mathbf{N}^{-1})\mathbf{n}) \,.$$

Multidimensional Discrete Fourier transform

- Multidimensional DFT supports the circular convolution of multidimensional signals:

$$g(\mathbf{n}) \triangleq f(\mathbf{n}) \circledast h(\mathbf{n}) = \sum_{\mathbf{m}} h(\mathbf{m}) f\left(\left((n_1 - m_1)\right)_{N_1}, \ldots, \left((n_M - m_M)\right)_{N_M}\right),$$

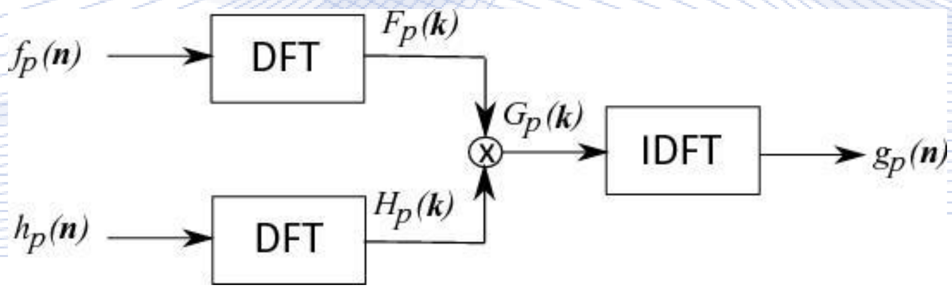
- $((n))_N = n \bmod N$.
- Spatial circular convolution over $\mathbb{Z}^M$ is equivalent to multiplication in the DFT domain :

$$g(\mathbf{n}) = f(\mathbf{n}) \circledast h(\mathbf{n}) \leftrightarrow G(\mathbf{k}) = F(\mathbf{k}) H(\mathbf{k}).$$

Artificial Intelligence & Information Analysis Lab

Multidimensional Discrete Fourier transform

- Linear multidimensional convolution $g(\mathbf{n}) = f(\mathbf{n}) * h(\mathbf{n})$ can be embedded in a circular one $g_p(n) = f_p(n) \circledast h_p(n)$.
- Signals $f(\mathbf{n}), h(\mathbf{n})$ must be zero padded of signals $f(\mathbf{n}), h(\mathbf{n})$ to get the signals $f_p(\mathbf{n}), h_p(\mathbf{n})$.
- Fast algorithms using multidimensional FFT, to calculate the multidimensional direct and inverse DFT.



Artificial Intelligence & Information Analysis Lab

Video Processing and Standards Conversion

- Multidimensional Signals and Systems
- Multidimensional Signal Transforms
- **Video Denoising**
- Video Interpolation

Three-dimensional spatiotemporal filtering

Simple additive noise model given by :

$$f(n_1, n_2, t) = s(n_1, n_2, t) + w(n_1, n_2, t),$$

- $f(n_1, n_2, t)$, $s(n_1, n_2, t)$, $w(n_1, n_2, t)$ denote the noisy video, the ideal video and the recorded noise, respectively, on frame t.
- 3D denoising filter types:
  - ***Temporal*** filters;
  - ***spatial*** (intraframe) filters or
  - ***spatiotemporal*** (interframe) filters.

Temporal Video Filters

Temporal filters:

- one-dimensional filters, e.g., to calculate the time-weighted average of successive video frames:

$$\hat{s}(n_1, n_2, t) = \sum_{l=-v}^{v} a(l) f(n_1, n_2, t-l),$$

- $a(l)$: filter coefficients for $2v+1$ consecutive video frames.
- 1D temporal moving average filter:

$$a(l) = \frac{1}{2v+1}, \mathrm{l} = -v, \dots, v.$$

Artificial Intelligence & Information Analysis Lab

Temporal Video Filters

Temporal filters:

- Filter coefficients can be determined by minimizing the following equation :

$$\min_{\mathbf{a}} E[\,(s(n_1,n_2,t) - \hat{s}(n_1,n_2,t))^2],$$

- $\mathbf{a}$: filter coefficient vector.
- $E[.]$: expectation operator (or error norm).
- Special case for an L_2 error norm: ***Wiener filter***.

Spatiotemporal video filters

3D moving average filters:

- They remove well additive noise;
- They have best performance in additive Gaussian noise removal;
- They tend to smooth/blur spatiotemporal edges.

3D median filters:

- They remove very well impulse noise;
- They preserve spatiotemporal object boundaries;
- They tend to destroy the spatiotemporal video details.

Adaptive spatiotemporal filters: they change their spatiotemporal region of support to adapt to local spatiotemporal video luminance characteristics.

Adaptive mean square filter

Mean adaptive square filter:

$$\hat{s}(n_1, n_2, t) = \left(1 - \frac{\sigma_w^2(n_1,n_2,t)}{\sigma_f^2(n_1,n_2,t)}\right) f(n_1, n_2, t) + \frac{\sigma_w^2(n_1,n_2,t)}{\sigma_f^2(n_1,n_2,t)} \mu_f(n_1, n_2, t),$$

- $\mu_f(n_1, n_2, t)$, $\sigma_f^2(n_1, n_2, t)$: local noisy signal mean and variance.
- $\sigma_w^2(n_1, n_2, t)$: recorded noise variance.

- It reduces filtering impact on spatiotemporal video edges, where there is large local luminance dispersion.

Artificial Intelligence & Information Analysis Lab

Adaptive mean square filter

- Local signal mean and variance estimators in the filter window $\mathcal{A}_{n_1,n_2,t}$:

$$\hat{\mu}_f(n_1,n_2,t) = \frac{1}{L}\sum_{(k_1,k_2,l)\epsilon\mathcal{A}_{n_1,n_2,t}} f(k_1,k_2,l)\,,$$

$$\hat{\sigma}_f^2(n_1,n_2,k) = \frac{1}{L}\sum_{(k_1,k_2,l)\epsilon\mathcal{A}_{n_1,n_2,t}} [f(k_1,k_2,l) - \hat{\mu}_f(n_1,n_2,t)]^2,$$

- $L = |\mathcal{A}_{n_1,n_2,t}|$ (number of pixels in the filter window).
- If $|\mathcal{A}_{n_1,n_2,t}$ is the parallelepiped $[-v_1,v_1]\times[-v_2,v_2]\times[-v_3,v_3]$, then $L= (2v_1+1)(2v_2+1)(2v_3+1)$.
- $\hat{\sigma}_w^2(n_1,n_2,k)$ is estimated in a small homogeneous region of video $f(n_1,n_2,t)$.

Adapted weighted average filter

- It computes a weighted average of the pixel values with in spatiotemporal filter window $\mathcal{A}_{n_1,n_2,t}$.

$$\hat{s}(n_1, n_2, t) = \sum_{(k_1,k_2,l)\epsilon\mathcal{A}_{n_1,n_2,t}} w(k_1, k_2, l) f(k_1, k_2, l),$$

$$w(k_1, k_2, l) = \frac{a(n_1, n_2, t)}{1 + \max\{\varepsilon, [f(n_1, n_2, t) - f(k_1, k_2, l)]^2\}},$$

- It is particularly suitable for filtering video shots that contain rapidly changing content.

Motion Compensated filtering

Motion compensated filter structure depends on:

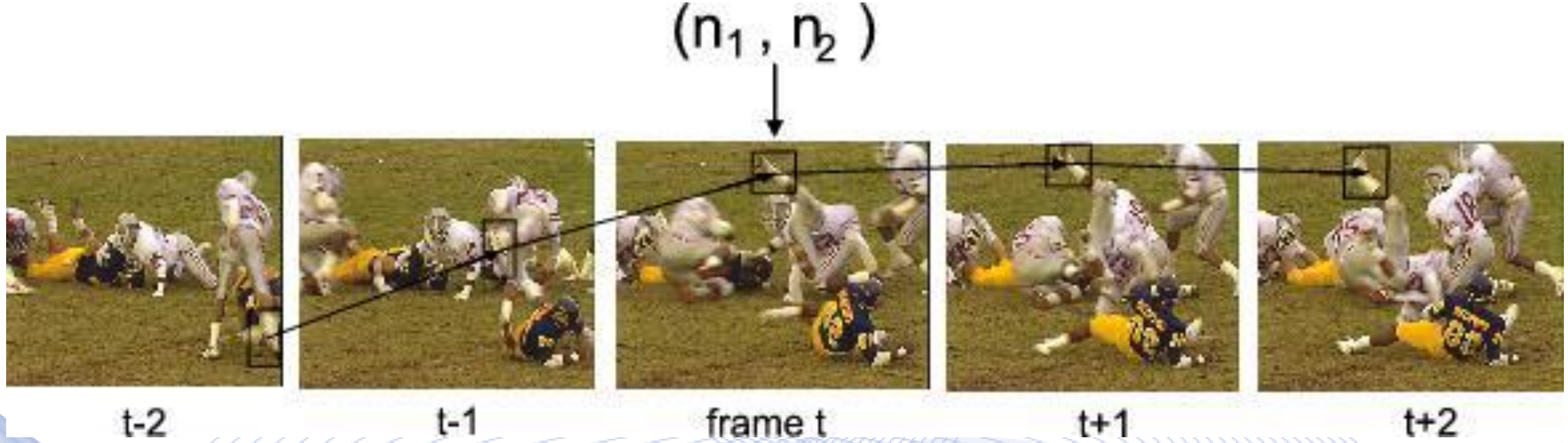
- the motion estimation method,
- the filter type (i.e., FIR versus IIR, adaptive versus non-adaptive).
- the filter window type (e.g., temporal versus spatiotemporal),

Filtering the t-th video frame of an image sequence using N video frames $t-v, \dots, t, \dots, t+v$, where $N = 2v+1$:

- evaluate the motion trajectory $\mathbf{d}(n_1, n_2, t, l)$, $l = t-v, \dots, t, \dots, t+v$ for each pixel (n_1, n_2) at frame t.
- The filter window $\mathcal{A}_{n_1,n_2,t}$ is the union of spatial neighborhoods (e.g., of size 3×3 pixels) that are centered on the motion trajectory positions of pixel (n_1, n_2, t).

Artificial Intelligence & Information Analysis Lab

Motion Compensated filtering



Motion trajectory in five successive video frames.

Video Processing and Standards Conversion

- Multidimensional Signals and Systems
- Multidimensional Signal Transforms
- Video Denoising
- **Video Interpolation**

Temporal video interpolation

- If we want to change the video sampling period from Δt_1 to Δt_2 we can interpolate the video using linear filtering :

$$f_i(x, n\Delta t_2) = \sum_m f(x, m\Delta t_1) h(n\Delta t_2 - m\Delta t_1).$$

- $h(t)$: ***interpolation kernel***.

Temporal video interpolation

- If the continuous video spectrum pixels satisfies the Nyquist criterion along Ω_t, then the interpolated pixel can be calculated by using the ideal sinc kernel:

$$h(t) = \frac{\sin(\pi t/\Delta t_1)}{\pi t/\Delta t_1}.$$

- This kernel entails large computational complexity.
- Alternatively, other interpolation kernels can be used, especially of polynomial form, e.g., zero-order, first-order (linear), spline interpolation.

Temporal video interpolation

- In most cases, simple low-order interpolation kernels are used.
- Zero-order interpolation kernel:

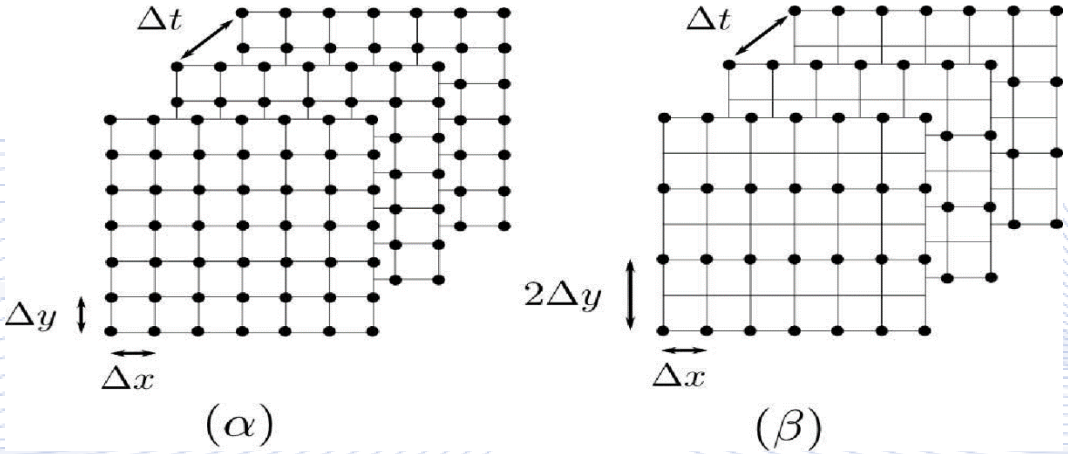
$$h(t) = \begin{cases} 1, & \text{if } 0 \leq t \leq \Delta t_1 \\ 0, & \text{elsewhere.} \end{cases}$$

- Linear interpolation kernel:

$$h(t) = \begin{cases} 1 - |t|/\Delta t_1, & \text{if } 0 \leq t \leq \Delta t_1 \\ 0, & \text{elsewhere.} \end{cases}$$

Spatiotemporal video interpolation

- 3D spatiotemporal sampling grids Λ_1, Λ_2.
- Deinterlacing: transformation of interlaced video to progressive one.



Sampling grids for: a) Progressive; b) 2:1 interlaced video.

Spatiotemporal video interpolation

- Union and the intersection of two grids as follows :

$$\Lambda_1 \cup \Lambda_2 = \{\mathbf{x}_t | \mathbf{x}_t \in \Lambda_1 \ \mathbf{x}_t \in \Lambda_2\},$$

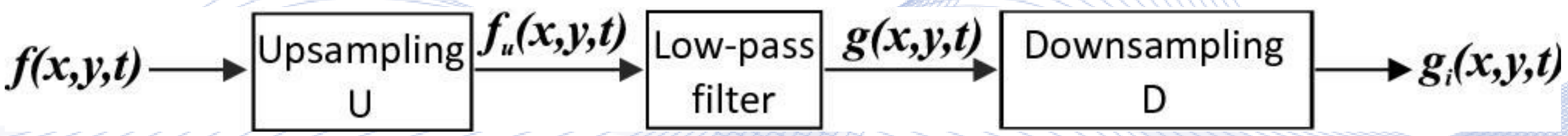
$$\Lambda_1 \cap \Lambda_2 = \{\mathbf{x}_t | \mathbf{x}_t \in \Lambda_1 \ \mathbf{x}_t \in \Lambda_2\}.$$

- $\mathbf{x}_t = [x, y, t]$.

Spatiotemporal video interpolation

Video sampling conversion from 3D spatiotemporal sampling grid from an initial grid Λ_1 to a final grid Λ_2:

- Initially, the input video is sampled on the grid $\Lambda_1 \cup \Lambda_2$.
- After appropriate filtering, it is down sampled at the output grid Λ_2.



Sampling grid conversion

Spatiotemporal video interpolation

- Oversampling from Λ_1 to $\Lambda_1 \cup \Lambda_2$:

$$f_u(\mathbf{x}_t) = \begin{cases} f_p(\mathbf{x}_t), & \mathbf{x}_t \in \Lambda_1 \\ 0, & \mathbf{x}_t \notin \Lambda_1, \end{cases} \qquad \mathbf{x}_t \in \Lambda_1 \cup \Lambda_2.$$

- Interpolation filter is applied to grid $\Lambda_1 \cup \Lambda_2$:

$$g(\mathbf{x}_t) = \sum_{\mathbf{z}_\tau \in \Lambda_1} f(\mathbf{z}_\tau) h(\mathbf{x}_t - \mathbf{z}_\tau), \qquad \mathbf{x}_t \in \Lambda_1 \cup \Lambda_2$$

- The frequency response of the digital filter $h(\mathbf{x}_t)$ must be defined in the unit cell of the grid $(\Lambda_1 \cup \Lambda_2)^*$.

Spatiotemporal video interpolation

- Downsampling from $\Lambda_1 \cup \Lambda_2$ to Λ_2 :

$$g_i(\mathbf{x}_t) = g(\mathbf{x}_t), \quad \mathbf{x}_t \in \Lambda_2.$$

- The undersampled output video is given by:

$$g_i(\mathbf{x}_t) = \sum_{\mathbf{z}_\tau \in \Lambda_1} f_p(\mathbf{z}_\tau) h(\mathbf{x}_t - \mathbf{z}_\tau), \qquad \mathbf{x}_t \in \Lambda_2.$$

Video deinterlacing

Deinterlacing: video conversion from an interlaced sampling grid to a progressive grid.

- Sampling matrices for the interlaced video $\mathbf{V}_i$ and progressive video $\mathbf{V}_p$:

$$\mathbf{V}_i = \begin{bmatrix} \Delta x & 0 & 0 \\ 0 & 2\Delta y & \Delta y \\ 0 & 0 & \Delta t/2 \end{bmatrix}, \qquad \mathbf{V}_p = \begin{bmatrix} \Delta x & 0 & 0 \\ 0 & \Delta y & 0 \\ 0 & 0 & \Delta t \end{bmatrix}.$$

- Passband of the interpolation filter: the unit cell of the inverse of the progressive video sampling grid:

$$\left(-\frac{1}{2\Delta x}, \frac{1}{2\Delta x}\right) \times \left(-\frac{1}{2\Delta y}, \frac{1}{2\Delta y}\right) \times \left(-\frac{1}{2\Delta t}, \frac{1}{2\Delta t}\right).$$

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr