2D Digital Filter Design and Implementation summary

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

Version 3.2.3

Artificial Intelligence & Information Analysis Lab

Outline

- FIR filter design
- Direct 2D FIR filter implementation
- Digital FIR filter implementation using FFT
- Block-based convolution methods
- IIR filter design.

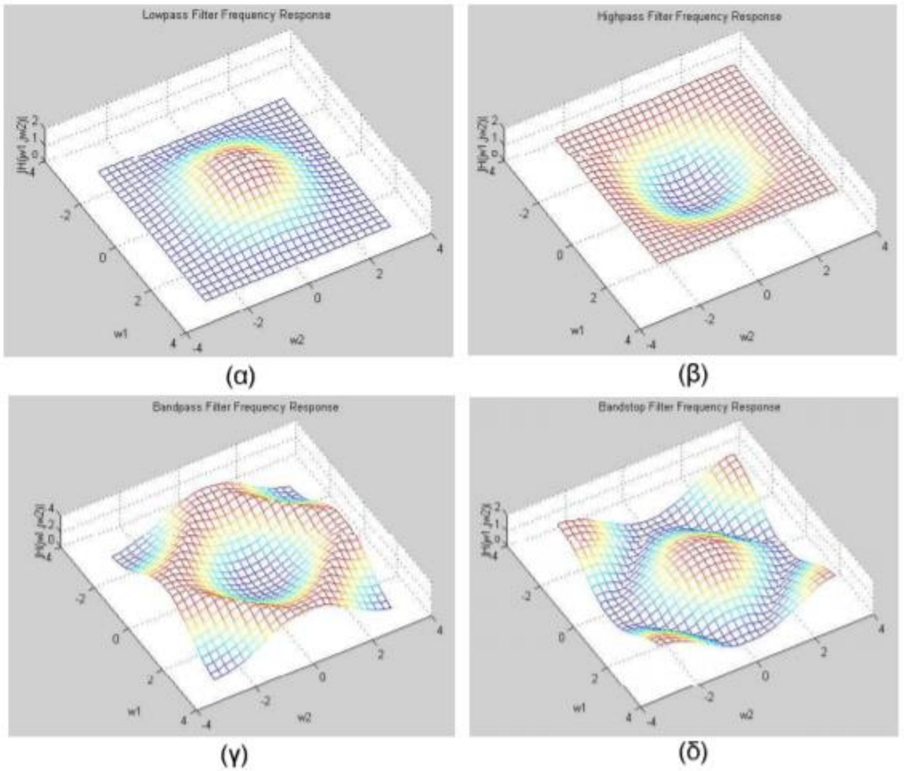
Type of 2D FIR Filter

- 2D digital filters are designed and implemented in such a way as to cut some spatial frequency bands, while allowing others to pass through.
- Types of 2D digital filters: Low pass filters, high pass filters, bandpass filters, bandage filters

Type of 2D FIR Filter

- ***Low-pass filters*** leave low-frequency content unaffected, while completely attenuating high-frequency content.
- ***High-pass filters*** behave just the opposite, leaving high frequencies unaffected and cutting off low frequencies.
- ***Bandpass filters*** filter all other frequencies, except those in their ***passband***.
- ***Bandstop filters*** perform exactly the opposite function: they cut the frequency content within a specific frequency ***stopband***.

Type of 2D FIR Filter



Frequency response of a 2D: a) Low-pass filter; b) High-pass filter; c) Bandpass filter; d) Bandstop filter.

2D FIR Filter Design

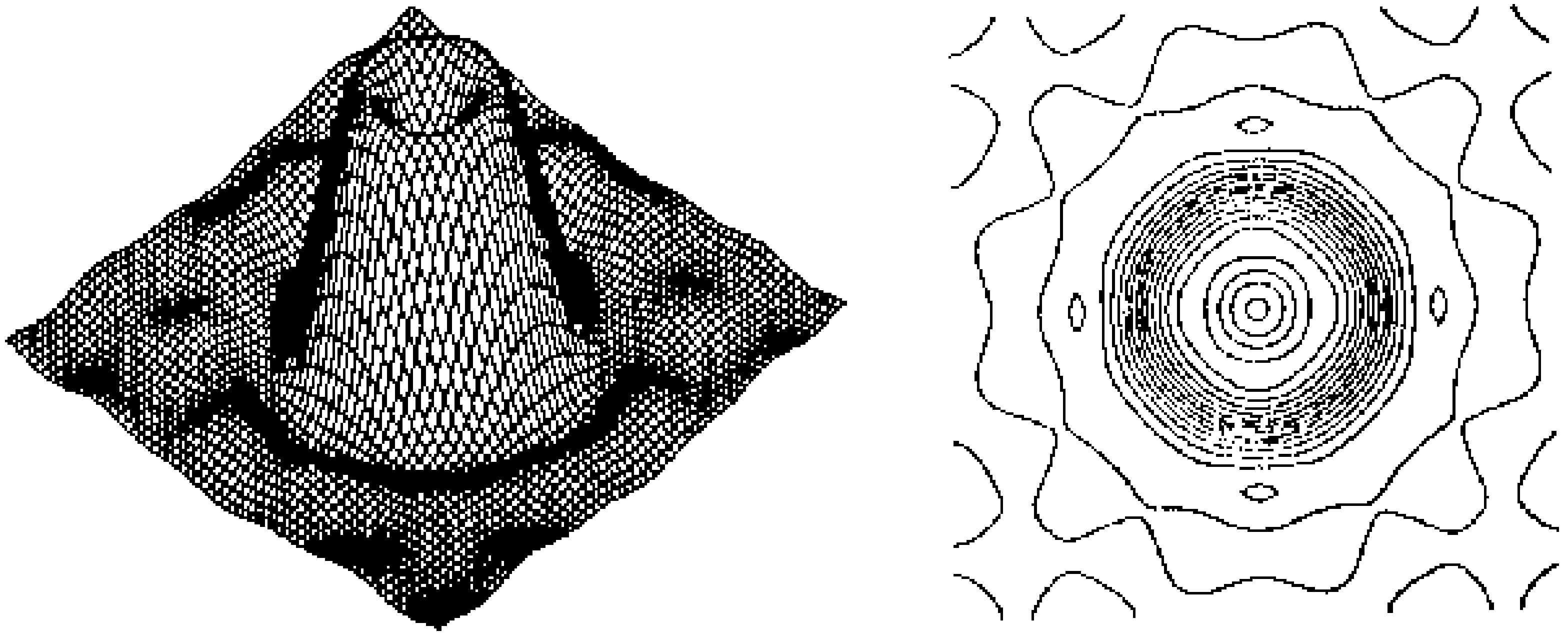
Windows-based 2D FIR filter design method is also employed in the 1D case.

- It is rather simple and thus widely used:

$$h(n_1,n_2) = i(n_1,n_2)w(n_1,n_2).$$

- $i(n_1,n_2)$: ideal 2D filter impulse response.
- $h(n_1,n_2)$: designed 2D filter impulse response.
- $w(n_1,n_2)$: 2D window function.

2D FIR Filter Design



Low-pass 11×11 2D FIR filter designed with the aid of windows method, utilizing a separable 2D window.

Direct 2D FIR filter implementation

2D FIR filters are linear, translation invariant 2D systems of finite support region.

- They have non-zero impulse response only within:

$$\mathcal{R}_{M_1M_2} = [0, M_1) \times [0, M_2) = \{(n_1, n_2): 0 \leq n_1 < M_1, 0 \leq n_2 < M_2\}$$

- The output of a 2D FIR filter is given by a ***linear convolution***:

$$y(n_1, n_2) = \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2).$$

for a *filter window* (region of support) $[0, M_1 - 1] \times [0, M_2 - 1]$.

Direct 2D FIR Filter Implementation

- For centered filter window $[-v_{1,}\ \ v_1] \times [-v_{2,}\ \ v_2]$, $M_i\ = 2v_i + 1$, $i = 1, 2$:

$$y(n_{1,} n_2) = \sum_{k_1=-v_1}^{v_1} \sum_{k_2=-v_2}^{v_2} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2).$$

Without input zero padding:

- an FIR filter having window $[0, M_1\ - 1] \times [0, M_2\ - 1]$ can operate only within the $[M_1 - 1, N_1) \times [M_2 - 1, N_2)$ part of the input image.
- an FIR filter having window $[-v_{1,}\ \ v_1] \times [-v_{2,}\ \ v_2]$ can operate only within the $[v_1, N_1 - v_1) \times [v_2, N_2 - v_2)$ part of the input image.

Direct 2D FIR Filter Implementation

2D Moving Average filter is a 2D FIR filter:

$$y(n_1, n_2) = \left(\frac{1}{M_1 M_2}\right) \sum_{k_1=-\nu_1}^{\nu_1} \sum_{k_2=-\nu_2}^{\nu_2} x(n_1 - k_1, n_2 - k_2),$$

where $M_i = 2v_i + 1,\ i = 1, 2$.

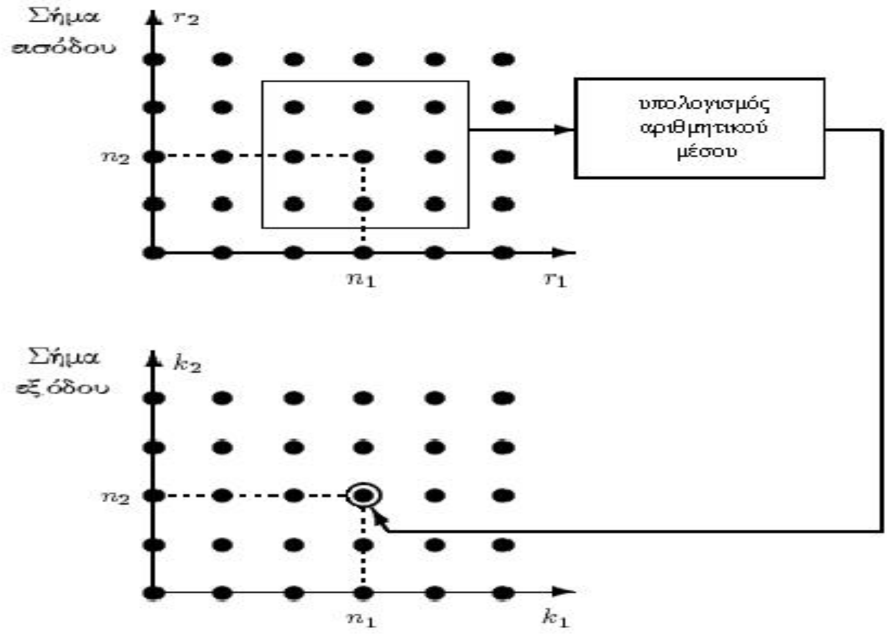
Properties:

- Very efficient in removing additive white Gaussian noise.
- It tends to blur edges and image details (e.g., lines, fine texture).
- It degrades image quality.

Direct 2D FIR Filter Implementation

Moving average image filtering.

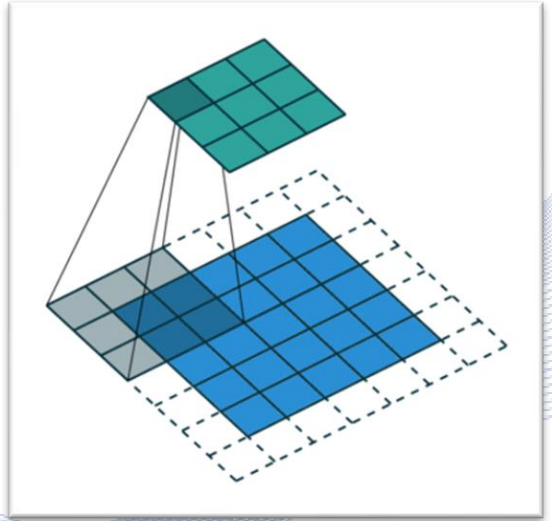
Direct 2D FIR Filter Implementation



Row-wise image scanning for 2D FIR filtering.

Direct 2D FIR Filter Implementation

- ***Zero padding*** the input image edges by a $\nu_1 = \nu_2 = \nu$ pixel wide border ribbon allows convolution operator to operate on the entire input image domain.
- Padding is arbitrary and can be done by any other pixel value, e.g., the ones of the outermost image rows and column pixels.
- If no padding is performed, the output image has reduced size $(N_1 - 2\nu) \times (N_2 - 2\nu)$.



2D FFT Filter Implementation

- Circular convolution of signal $x(n_1,n_2)$ and impulse response $h(n_1,n_2)$ is defined by:

$$y(n_1,n_2) = h(n_1,n_2) \circledast\circledast x(n_1,n_2)$$
$$= \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} x(k_1,k_2) h\left(((n_1-k_1))_{N_1}, ((n_2-k_2))_{N_2} \right)$$

- $((n))_N = n \bmod N$.

Artificial Intelligence & Information Analysis Lab

2D FFT Filter Implementation

- Circular convolution of sequences x, h satisfy the property:

$$y(n_1, n_2) = \mathrm{IDFT}\big[\mathrm{DFT}[x(n_1, n_2)]\mathrm{DFT}[h(n_1, n_2)]\big]$$

- DFT: 2D Discrete Fourier transform:

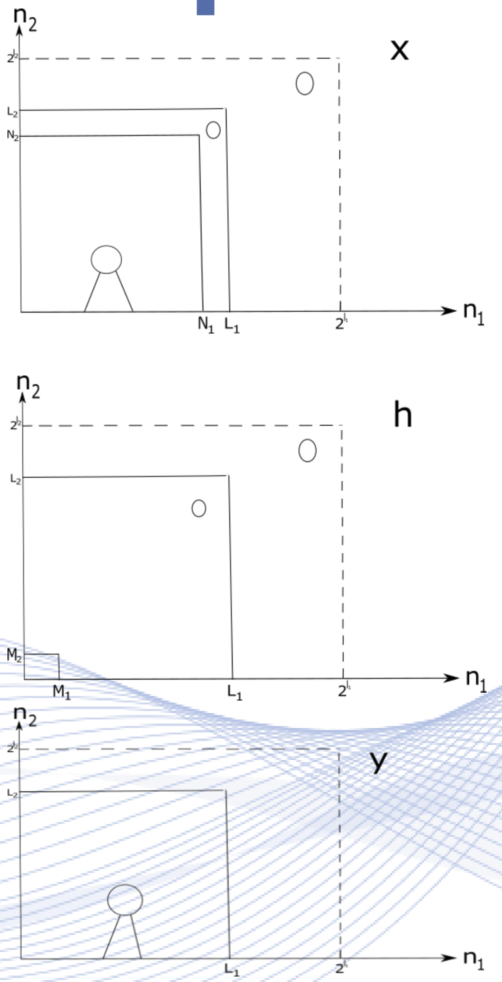
$$X(k_{1,} k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-2} x(n_1, n_2) W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2}.$$

- IDFT: inverse discrete Fourier transform.
- DFT and IDFT can be calculated effectively using 2D FFT.

2D FFT Filter Implementation

- If $x(n_1, n_2)$ support region is: $[0, N_1) \times [0, N_2)$ and $h(n_1, n_2)$ support region is: $[0, M_1) \times [0, M_2)$, $y(n_1, n_2)$ support region is $[0, N_1 + M_1 - 2) \times [0, N_2 + M_2 - 2)$.
- 2D linear convolution can be estimated in the same way, by embedding it to a 2D circular convolution.

2D FFT Filter Implementation



Zero padding for embedding a 2D linear convolution to a cyclic one.

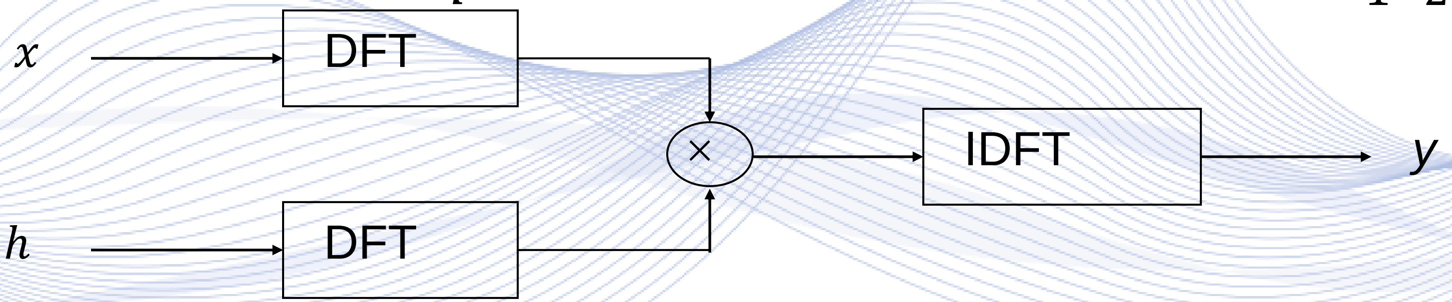
2D FFT Filter Implementation

- Calculate $y_p(n_1, n_2)$ by using the inverse IDFT of:

$$Y_p(k_1, k_2) = X_p(k_1, k_2)H_p(k_1, k_2).$$

- Calculate $y_p(n_1, n_2)$ by using the inverse DFT.
- The result of the linear convolution is:

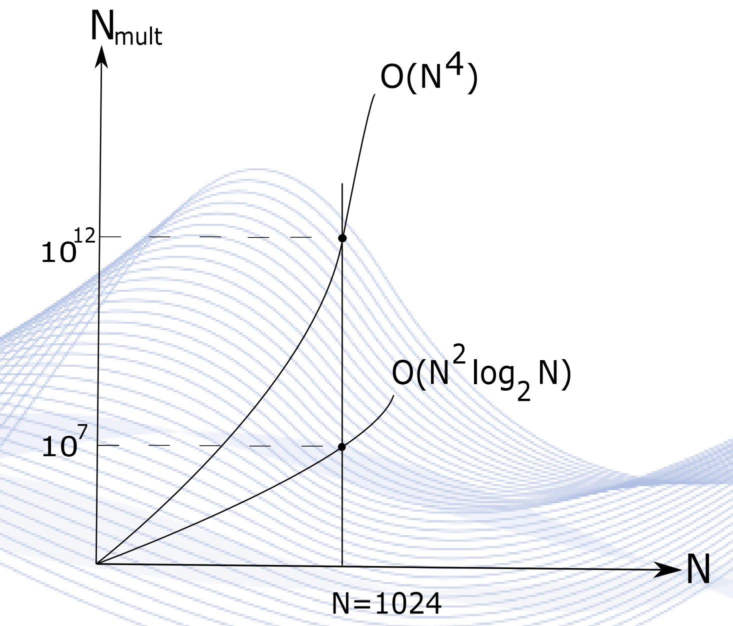
$$y(n_1, n_2) = y_p(n_1, n_2), \qquad (n_1, n_2) \in \mathcal{R}_{L_1 L_2}.$$



Convolution calculation using the DFTs.

2D FFT Filter Implementation

- For larger filters (close to the image size), computational complexity is:
  - $O(kN^4)$ for the direct method.
  - $O(kN^2 \log_2 N)$ using 2D FFT.



Computational complexity of 2D FIR filters.

2D IIR filter design

In the following, $I(\omega_1, \omega_2)$ will denote the ideal frequency response and $H(\omega_1, \omega_2)$ denotes the frequency response of the IIR filter under design:

$$H(\omega_1, \omega_2) = \frac{A(\omega_1, \omega_2)}{B(\omega_1, \omega_2)} = \frac{\sum_{n_1}\sum_{n_2} a(n_1, n_2) e^{(-i\omega_1 n_1 - i\omega_2 n_2)}}{\sum_{n_1}\sum_{n_2} b(n_1, n_2) e^{(-i\omega_1 n_1 - i\omega_2 n_2)}}.$$

2D IIR filter design

- Norms E_2, E_p, E_∞ can be used as a measure of filter design success:

$$E_2 = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\left|I(\omega_1,\omega_2) - \frac{A(\omega_1,\omega_2)}{B(\omega_1,\omega_2)}\right|^2 d\omega_1 d\omega_2$$

$$E_p = \left[\left(\frac{1}{4\pi^2}\right)\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\left|I(\omega_1,\omega_2) - \frac{A(\omega_1,\omega_2)}{B(\omega_1,\omega_2)}\right|^p d\omega_1 d\omega_2\right]^{\frac{1}{p}}$$

$$E_\infty = \max\left|I(\omega_1,\omega_2) - \frac{A(\omega_1,\omega_2)}{B(\omega_1,\omega_2)}\right|$$

2D IIR Filter Implementation

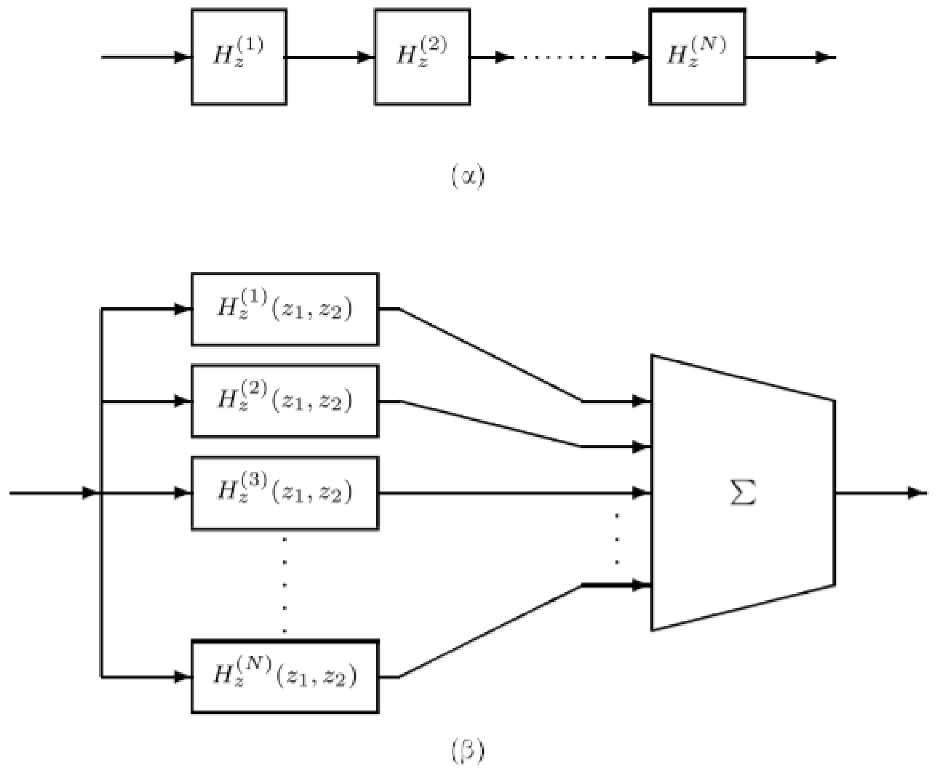
Chain or parallel 2D IIR filter implementation may be used, by bring the filter transfer function in the following product or sum forms, respectively:

$$\mathrm{H}(z_1, z_2) = \prod_{i=1}^{N} \frac{A_i(z_1, z_2)}{B_i(z_1, z_2)},$$

$$\mathrm{H}(z_1, z_1) = \sum_{j=1}^{N'} \frac{A_j(z_1, z_2)}{B_j(z_1, z_2)}.$$

Artificial Intelligence & Information Analysis Lab

2D IIR Filter Implementation



a) Chain 2D IIR filter implementation (b) Parallel 2D IIR filter.

Artificial Intelligence & Information Analysis Lab

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

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