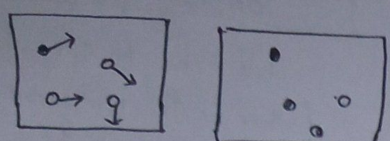
Optical flow: Optical flow used for motion estimatⁿ in visual odometry.

Problem: Given two consecutive image frames, estimate the motion of each pixel.



$I(x,y,t)$ $I(x,y,t')$

Extimate the motion (flow) b/w these two conseccutive images

Key Assumptions

- Brightness constancy
- Small motⁿ

Assumptions

① Brightness constancy
⇓
Color constancy
(Brightness constancy for intensity images)
&
Implicatⁿ: Allow for pixel to pixel comparison.
(not image features)

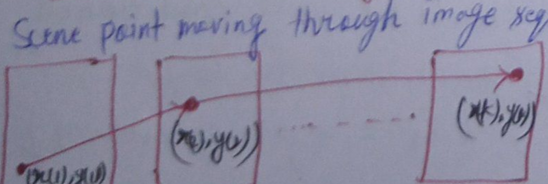
② Small Motion
⇓
pixels only move a little bit
&
Implicatⁿ: Linearizatⁿ of the brightness constancy constraint.

Look for nearby pixels with the same color

- nearby pixels ⇓ (small motⁿ)
- same color ⇓ (color constancy)

Assumptⁿ 1: Brightness constancy

Scene point moving through image sequence

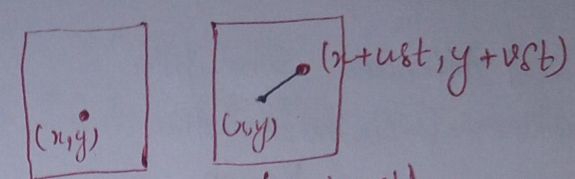


$I(x,y,1)$ $I(x,y,2)$. $I(x,y,k)$

- Brightness of the point will remain the same

$$\boxed{I(x(t), y(t), t) = \underset{\text{constant}}{C}}$$

Assumption 2: Small Motion



$I(x,y,t)$ $I(x,y,t+\delta t)$

Optical flow (velocities): (u, v)

Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

For a really small space-time step:

$$I(x+u\delta t, y+v\delta t+\delta t) = I(x,y,t)$$

the brightness b/w two consecutive image frames is the same.

$$\boxed{\underset{\text{total derivative}}{\frac{dI}{dt}} = \underbrace{\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}}_{\text{partial derivative}} = 0}$$

- If the time step is really small, we can linearize the intensity functⁿ.

→ Multivariable Taylor Series Expansion (First order approximatⁿ, two variables).

$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) - f_y(a,b)(y-b)$$

$$\Rightarrow I(x,y,t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t$$

divide by δt & take limit $\delta t \to 0$

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

Brightness Constancy Eqⁿ:

$$\boxed{\underbrace{\frac{\partial I}{\partial x}}_{I_x}\underbrace{\frac{dx}{dt}}_{u} + \underbrace{\frac{\partial I}{\partial y}}_{I_y}\underbrace{\frac{dy}{dt}}_{v} + \underbrace{\frac{\partial I}{\partial t}}_{I_t} = 0}$$

flow velocities, temporal gradient

$$\boxed{I_x u + I_y v + I_t = 0}$$ Short hand notatⁿ.

Image gradients (at a point p)

$$\boxed{\nabla I^T u + I_t = 0}$$ Vector form

(1×2) (2×1)

$$\boxed{I_x = \frac{\partial I}{\partial x} \; ; \; I_y = \frac{\partial I}{\partial y}}$$

spatial derivative

⇓ compute ⇓

Forward difference Sobal filter

Derivative-of-Gaussian filter.

$$\boxed{I_t = \frac{\partial I}{\partial t}}$$

temporal derivative

⇓ Compute ⇓

frame Differencing

$$\boxed{u = \frac{dx}{dt} \quad v = \frac{dy}{dt}}$$

optical flow

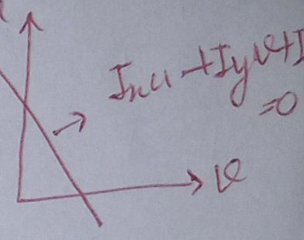
⇓

We need to solve for this

(this is the unknown in the optical flow problem)

⇓

Sol^n lies on a line



Known

$$\boxed{I_x u + I_y v + I_t = 0}$$

Unknown

Horn-Schunck Optical flow (1981)

- → brightness constancy
- → small mot^n
- → "Smooth" flow
- → flow can vary from pixel to pixel
- → Global methods (dense)

Lucas-Kanade Optical Flow (1981)

- → Methods of Diff.
- → "Constant" flow
- → flow is contant for all pixels.
- → local method (sparse).

Eqn to solving:

$$A^T A \hat{x} = A^T b$$

$$\begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix}$$

Where the summation is over each pixel p in patch P.

$$x = (A^T A)^{-1} A^T b$$

---

* Constant Flow : $I_x u + I_y v + I_t = 0$

Assume that the surrounding patch (5x5) has "constant flow"

Assumptions:
- → Flow is locally smooth
- → Neighboring pixels have same displacements.

Using a 5x5 image patch, gives us 25 eqn

$$I_x(P_1)u + I_y(P_1)v = -I_t(P_1)$$
$$I_x(P_2)u + I_y(P_2)v = -I_t(P_2)$$
$$\vdots$$
$$I_x(P_{25})u + I_y(P_{25})v = -I_t(P_{25})$$

When $A^T A \hat{x} = A^T b$ solvable?

- → $A^T A$ should be invertible
- → $A^T A$ should not be too small
  - λ_1 & λ_2 should not be too small
- → $A^T A$ should be well conditioned
  - λ_1 / λ_2 should not be too large.
  - i.e. (λ_1 = larger eigenvalues).

Implications

- → Corners are when λ_1 & λ_2 are big; this is also when Lucas-Kanade optical flow works best.
- → Corners are regions with two diff. direct^n of gradient (at least)
- → Corners are good places to compute flow!

Wavelet Transform in One Dimension

→ Discrete WT ✓

→ Continuous WT ✓

* Wavelet Series Expansions:

$$f(x) = \sum_{k} c_{j_0}(k)\, \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k} d_j(k)\, \psi_{j,k}(x)$$

approximat^n/ Scaling coefficients

Wavelet coefficient

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x)\, \varphi_{j_0,k}(x)\, dx \qquad \int f(x)\, \varphi_{j_0,k}(x)\, dx$$

$$d_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x)\, \psi_{j,k}(x)\, dx \qquad \int f(x)\, \psi_{j,k}(x)\, dx$$

Discrete Wavelet Transform

$$W_\varphi(j_0,k) = \frac{1}{\sqrt{M}} \sum_{n} f(n)\, \varphi_{j_0,k}(n) \rightarrow \text{appro}$$

wave coff

$$W_\psi(j,k) = \frac{1}{\sqrt{M}} \sum_{n} f(n)\, \psi_{j,k}(n) \;;\; j \geq j_0$$

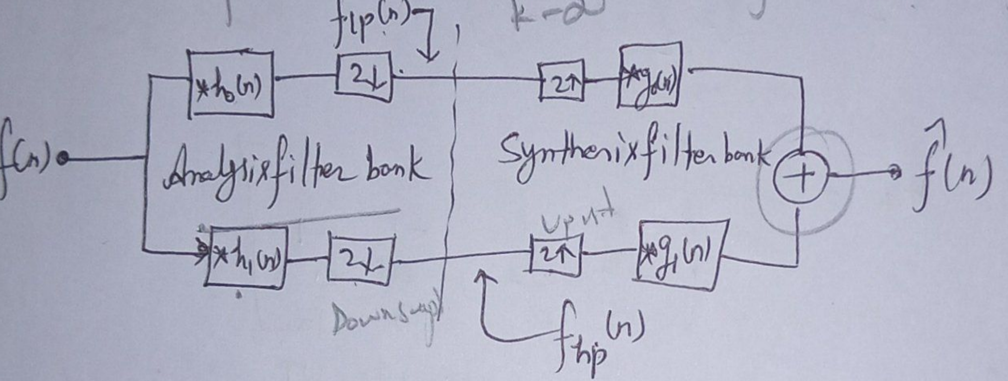
Normalized factor

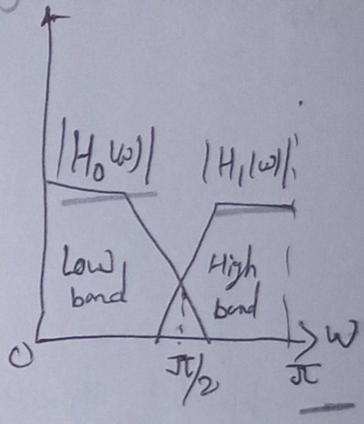
$$f(n) = \frac{1}{\sqrt{M}} \sum_{k} W_\varphi(j_0,k)\, \varphi_{j_0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{k} W_\psi(j,k)\, \psi_{j,k}(n)$$

2-band subband coding & decoding system

$$f(n-2) = \begin{cases} f(0) & \text{for } n=2 \\ f(1) & \text{for } n = 2+1 = 3 \end{cases}$$

$$\hat{f}(n) = \sum_{k=-\infty}^{\infty} h(k)\, f(n-k) = f(n) * h(n)$$



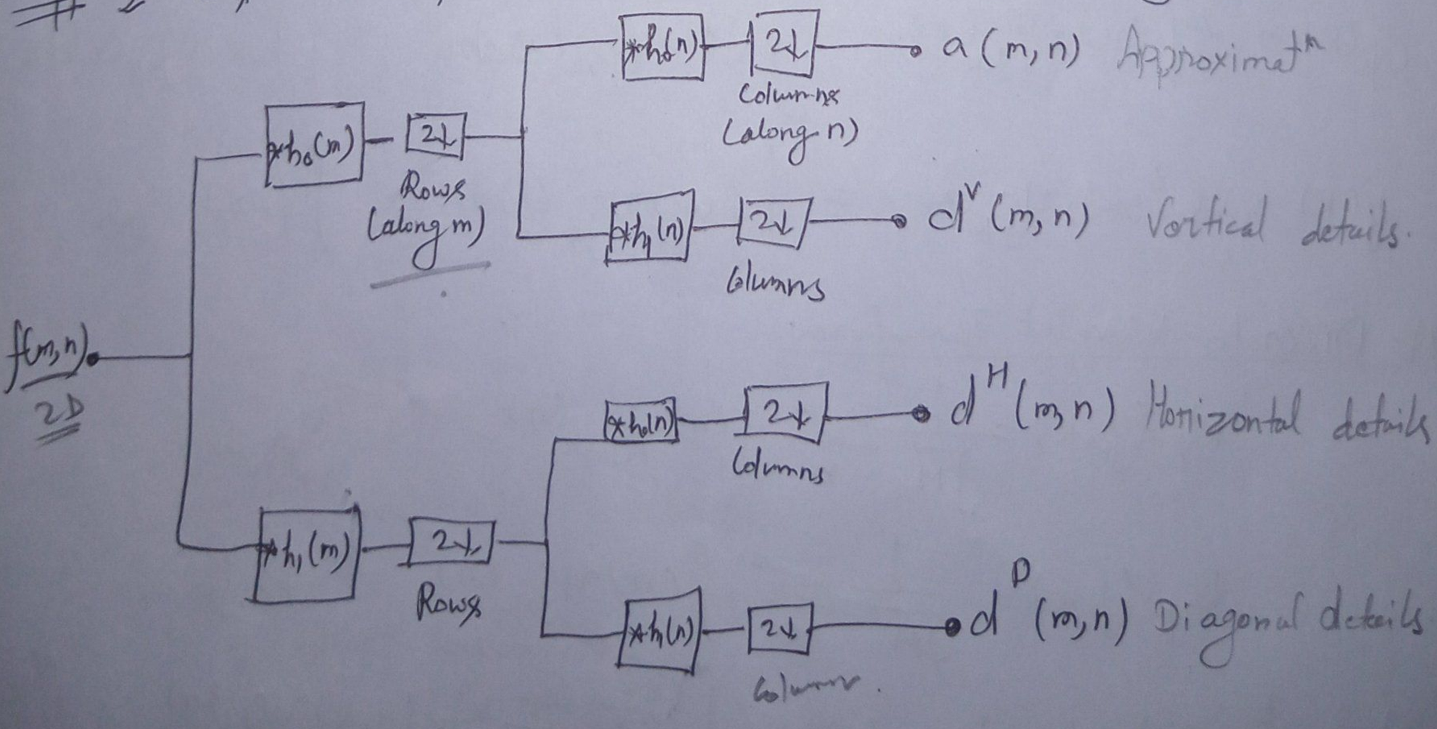


$$h_2(n) = -h_1(n)$$

$$g_0(n) = (-1)^n h_1(n)$$

$$g_1(n) = (-1)^{n+1} h_0(n)$$

2-D, 4 band filter bank for subband image coding



nuous Wavelet Transform CWT

$$W_\psi(s,\tau) = \int_{-\infty}^{\infty} f(x)\,\psi_{s,\tau}(x)\,dx$$

Where,

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\,\psi\left(\frac{x-\tau}{s}\right)$$

$\tau \rightarrow$ Translatⁿ parameter.

$s \rightarrow$ Scal

$$f(x) = \frac{1}{C_\psi}\int_{0}^{\infty}\int_{-\infty}^{\infty} W_\psi(s,\tau)\,\frac{\psi_{s,\tau}(x)}{s^2}\,d\tau\,ds$$

Where,

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(u)|^2}{|u|}\,du$$

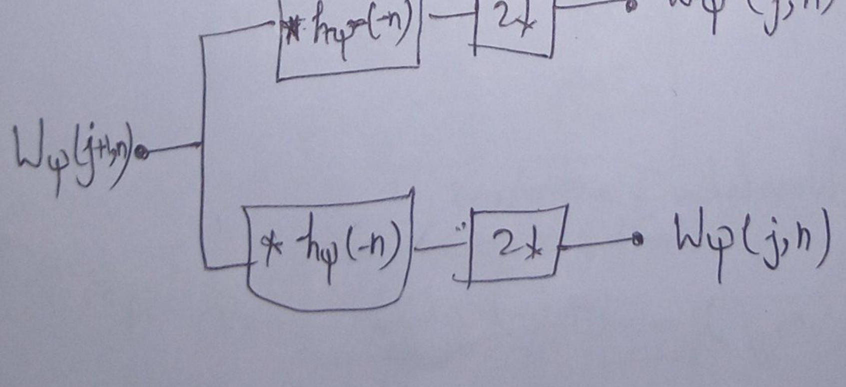
$\Psi(u) \rightarrow$ fourier transform of $\psi(x)$

Fast Wavelet Transform

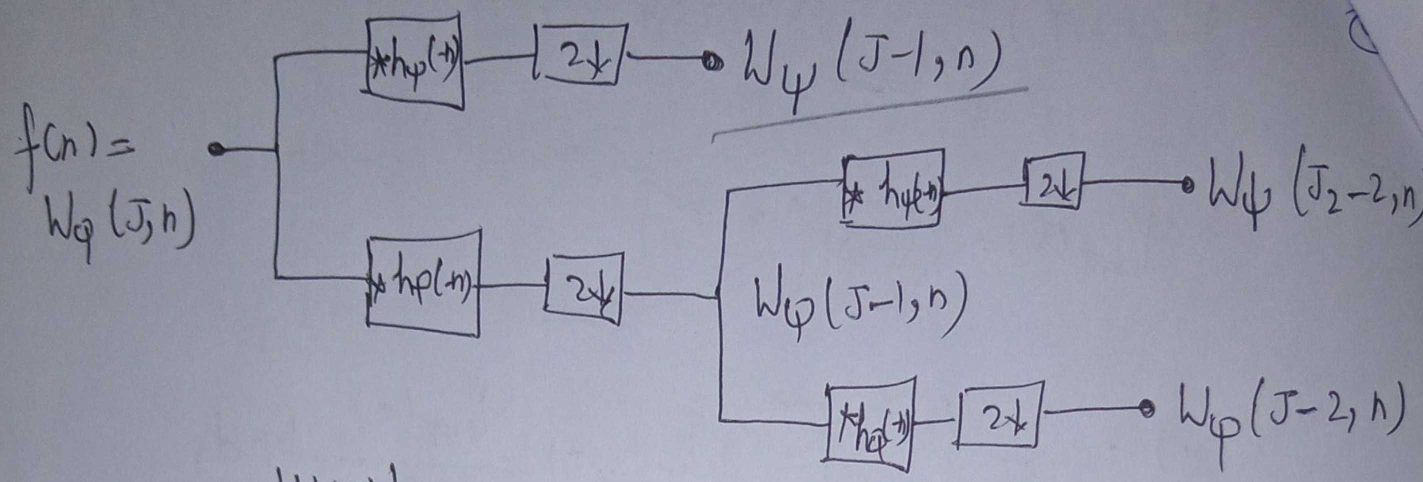
$$\boxed{\varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\,\varphi(2x-n)}$$

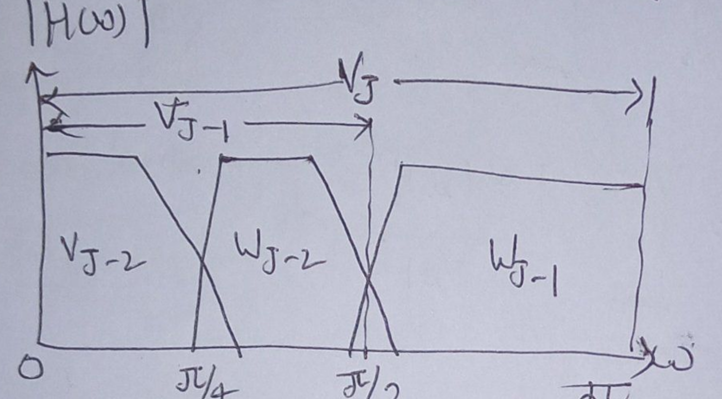
$$\boxed{W_\psi(j,k) = h_\psi(-n) * W_\varphi(j+1,n)}\Big|_{n=2k,\,k\geq 0}$$

$$\boxed{W_\varphi(j,k) = h_\varphi(-n) * W_\varphi(j+1,n)}\Big|_{n=2k,\,k\geq 0}$$

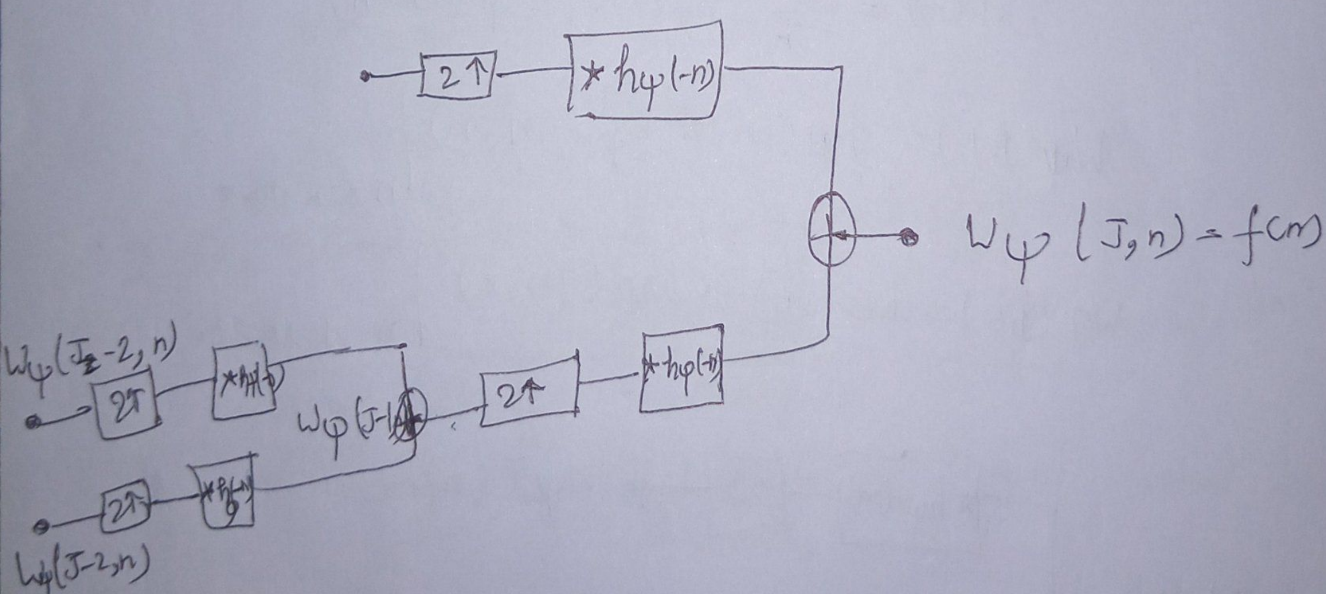


2-stage or 2-scal FWT analysis bank & its frequency sp...





FWT^{-1}



2D - Wavelet Transforms

$$\varphi(x,y) = \varphi(x)\varphi(y) \quad \text{— ①} \quad \text{approximation}$$

$$\psi^H(x,y) = \psi(x)\varphi(y) \quad \text{— ②}$$

$$\psi^V(x,y) = \varphi(x)\psi(y) \quad \text{— ③}$$

$$\psi^D(x,y) = \psi(x)\,\psi(y)$$ (5)

Ist define scaled & translated basic funct^n

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\,\varphi(2^j x - m,\ 2^j y - n)$$ — approx

wavelet $$\psi^i_{j,m,n}(x,y) = 2^{j/2}\,\psi^i(2^j x - m,\ 2^j y - n),\ \forall\, i = \{H,V,D\}$$

Wavelet transform

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\,\varphi_{j_0,m,n}(x,y)$$

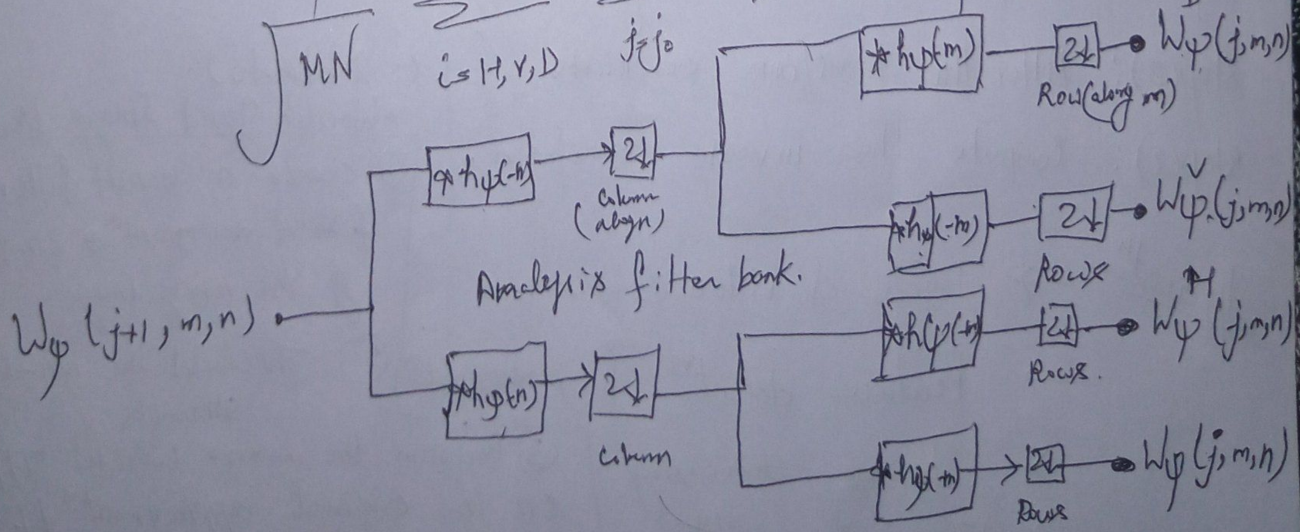
$\Downarrow$ Appr. coefficient

$$W_\psi(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\,\psi^i_{j,m,n}(x,y)$$

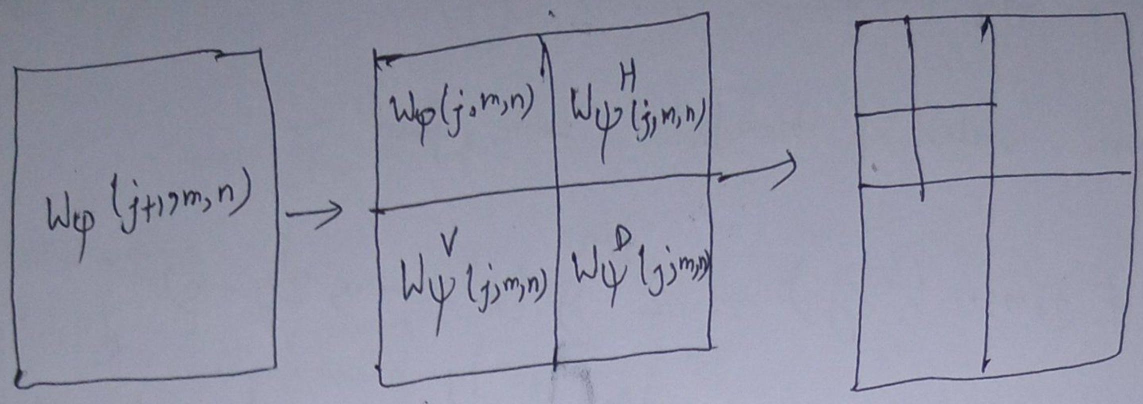
$\Downarrow$ H,V,D cofficient $\qquad i = \{H, V, D\}$.

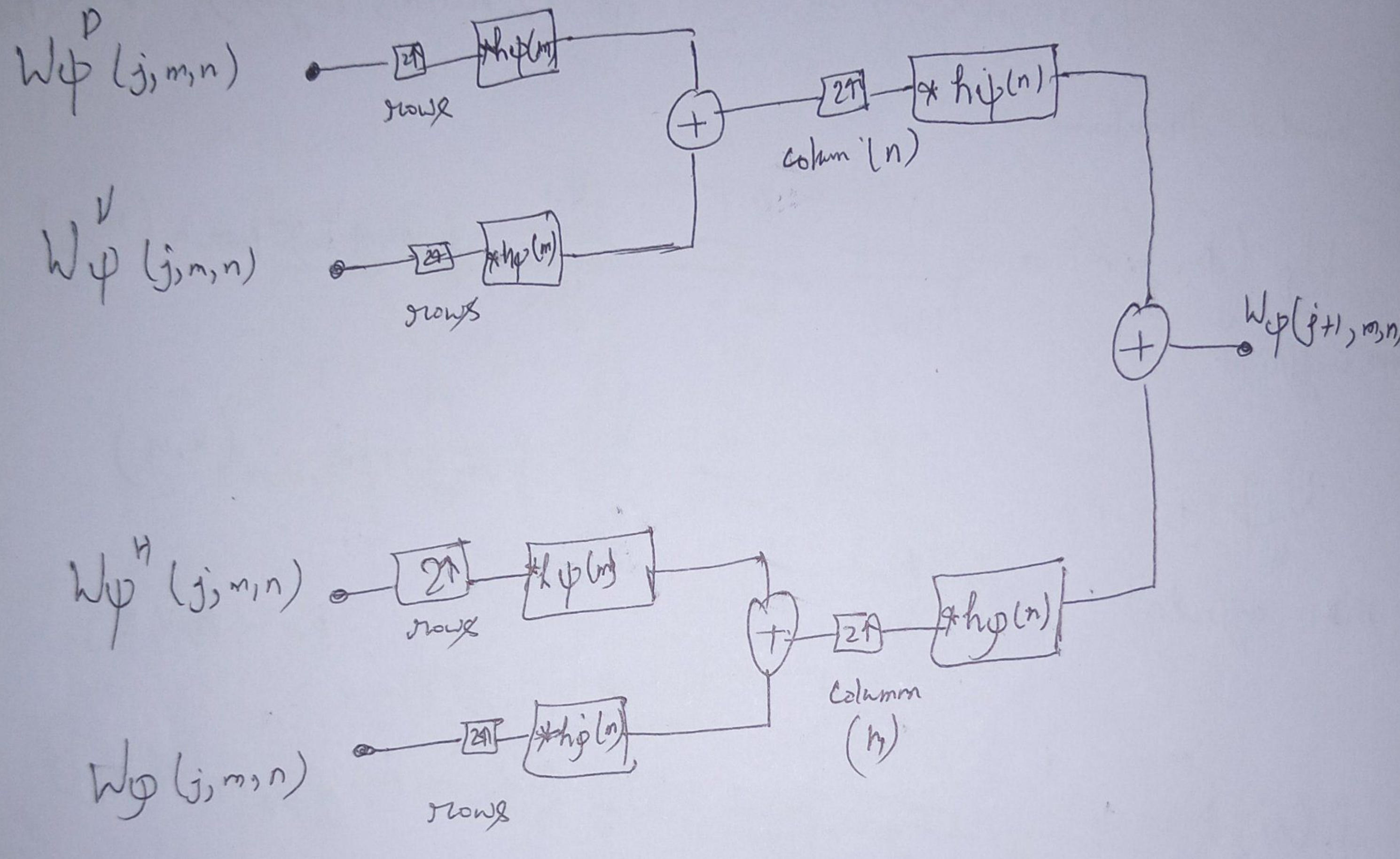
$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n)\,\varphi_{j_0,m,n}(x,y)$$

$$+ \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W^i_\psi(j,m,n)\,\psi^i_{j,m,n}(x,y)$$



Analysis filter bank.





Wavelet Transform in image processing

Step (1): Compute the 2D-DWT of an image.

step(2): Alter the transform coefficients (i.e. subbands).

step(3): Compute the inverse transform.

Applicatⁿ → Image & video compression

→ Feature detectⁿ & recognitⁿ

→ Image denoising

→ Face Recognitⁿ.

* Wavelet-Based Image Denoising

① choose a wavelet filter (Haar, Symlet) & No. of decompositⁿ → 2D-DWT of the noisy image.

② Threshold the non-LL subbands.

③ Perform the inverse wavelet T/F on the original approximatⁿ LL-subband & Modified non-LL subbands.