CS448f: Image Processing For Photography and Vision

Wavelets Continued

Last Time:

- Last time we saw the Daubechies filter satisfied the following:
  - fully orthogonal
  - four taps
  - as smooth as possible
  - wavelet filter a simple modification of the scaling filter
- Why did we care about our wavelet basis functions being orthogonal?

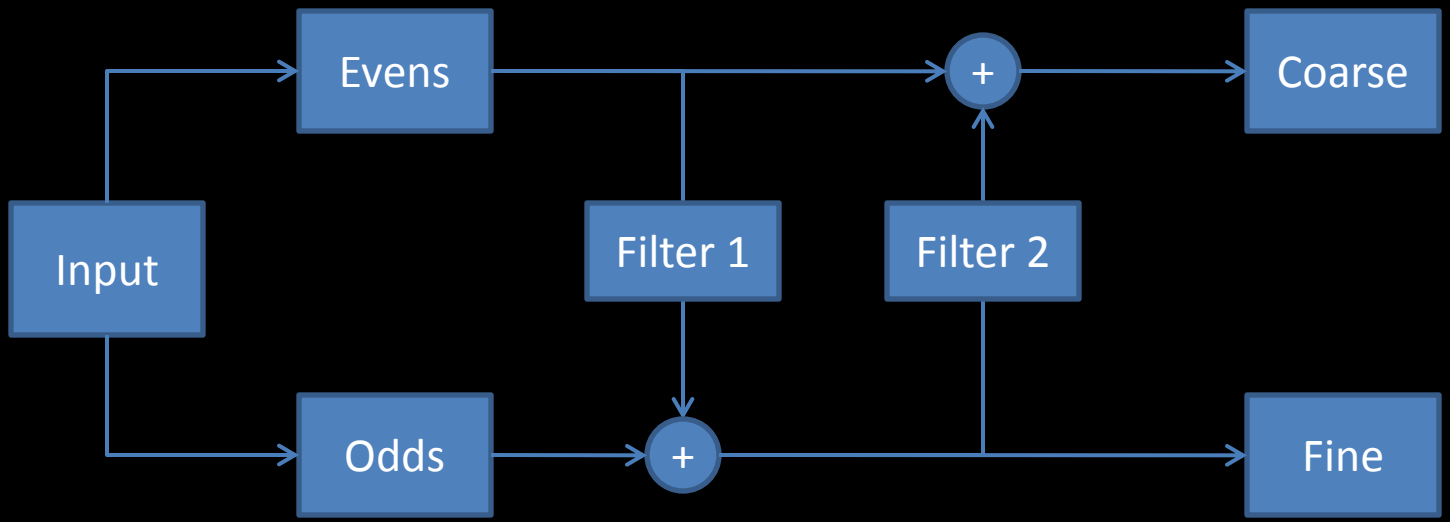
Orthogonality

- Why did we care about our wavelet basis functions being orthogonal?
  - Easy to invert
  - Orthogonal transforms preserve distance
- We can probably relax this requirement, provided we get something that's still easy to invert

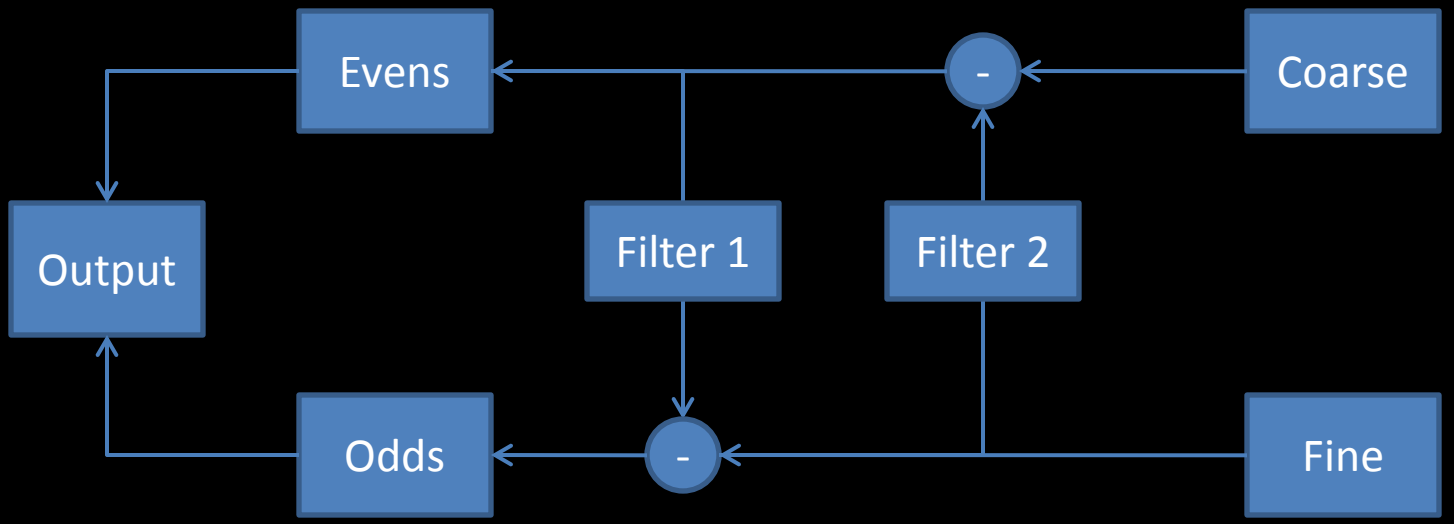
Lifting

- Let's construct our filters using the following sequence:
  - Divide the inputs into evens and odds
  - Add some function of the odds to the evens
  - Add some function of the evens to the odds
  - Repeat as long as you like
  - Eventually the evens form a coarse layer and the odds form a fine layer
- This is easy to invert

Forward Transform



Inverse Transform



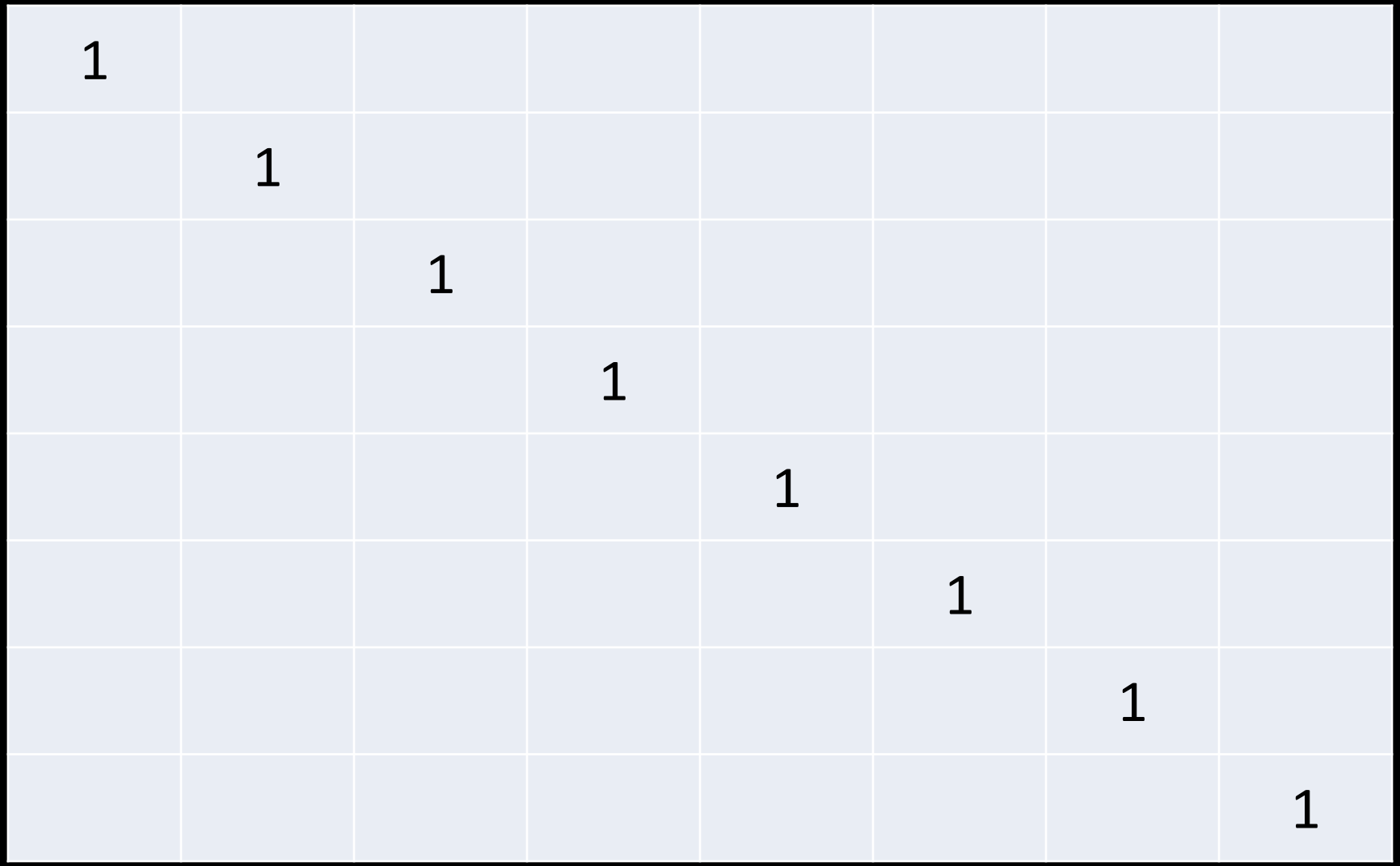
What makes a good fine layer?

- Average value is 0
- So filter 1 should probably be something that enforces that.

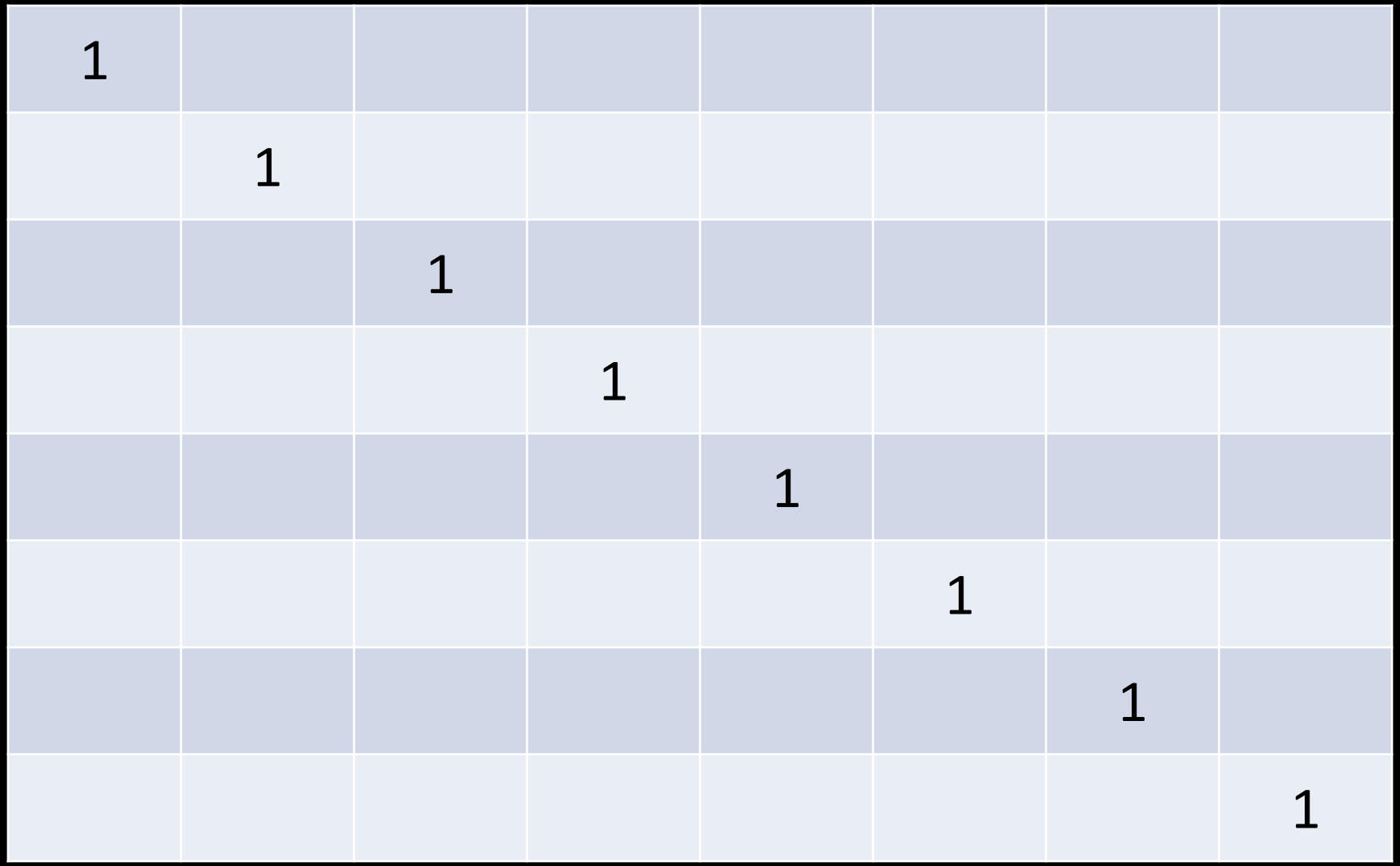
What makes a good coarse layer?

- Why is subsampling bad?
  - Some pixels in the input count more than others
- Each pixel in the input should count equally
  - E.g. Averaging down
- Something should sum up to 1

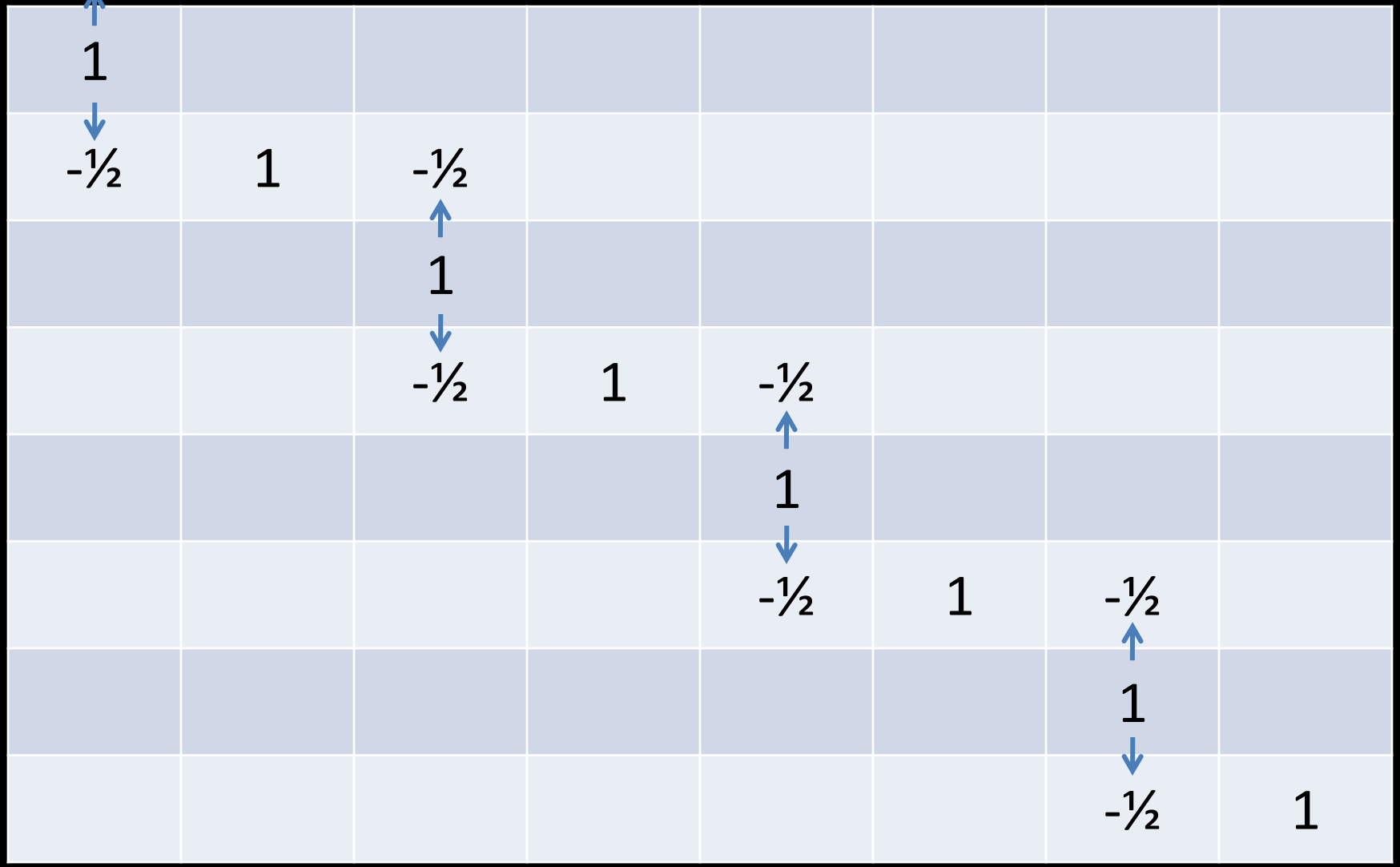
Let’s track the linear transform



Divide the rows into even and odd



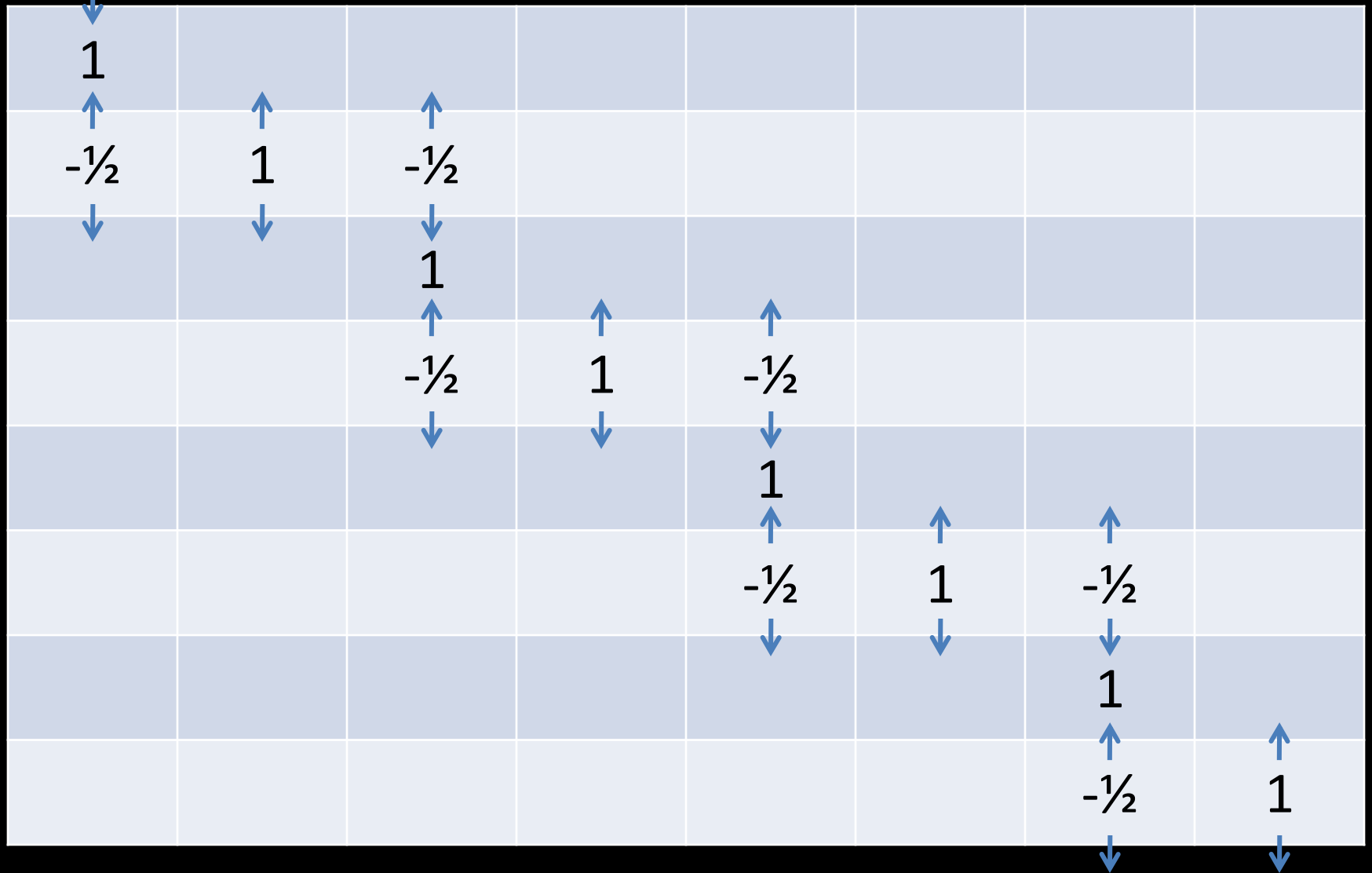
Add a filter of the even rows to the odd rows: [-½ 0 -½]



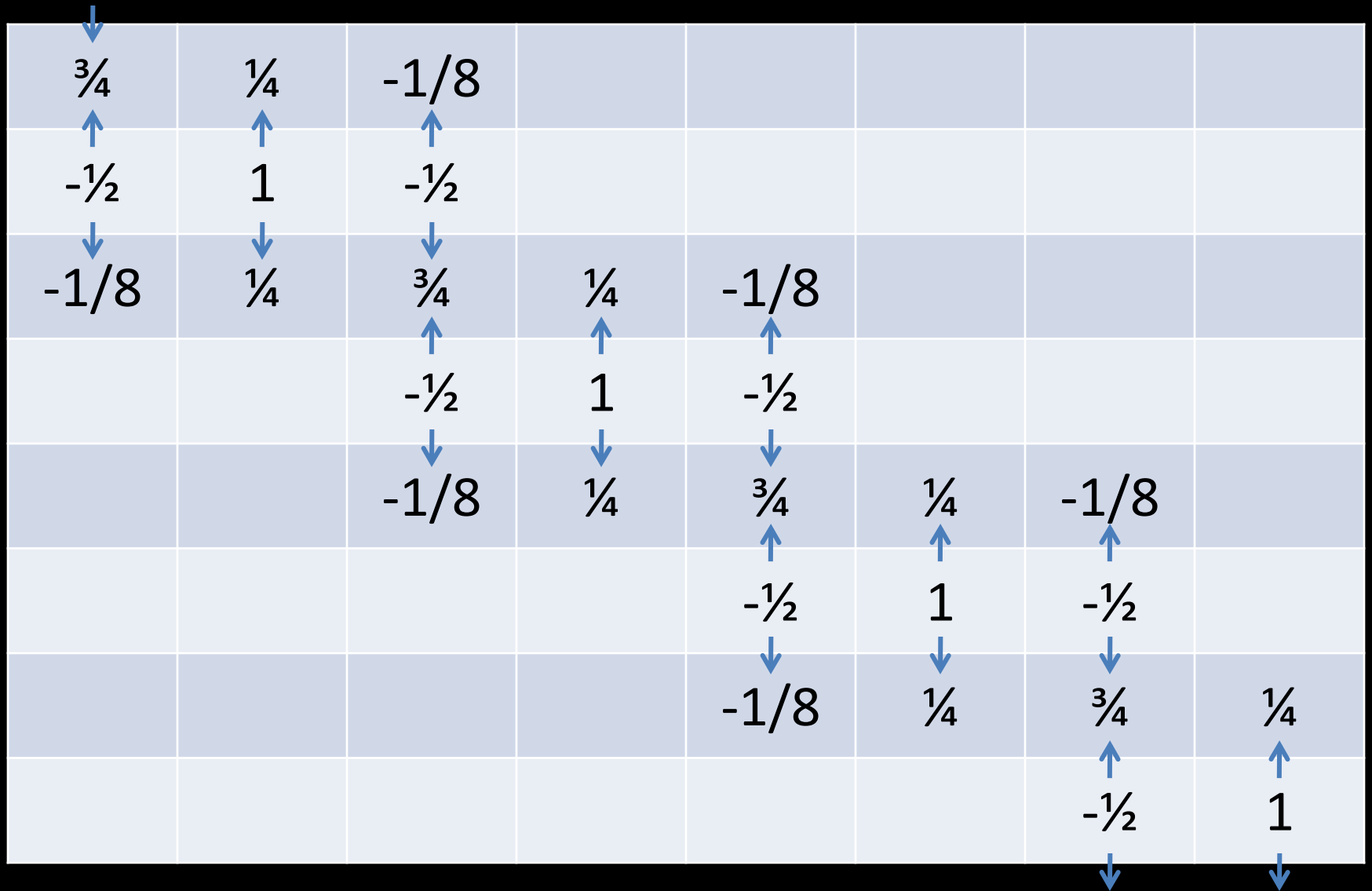
The odd rows are now a fine layer

1							
-½	1	-½					
		1					
		-½	1	-½			
				1			
				-½	1	-½	
						1	
						-½	1

Add a filter of the odd rows to the even rows: [¼ 0 ¼]



Add a filter of the odd rows to the even rows: [¼ 0 ¼]



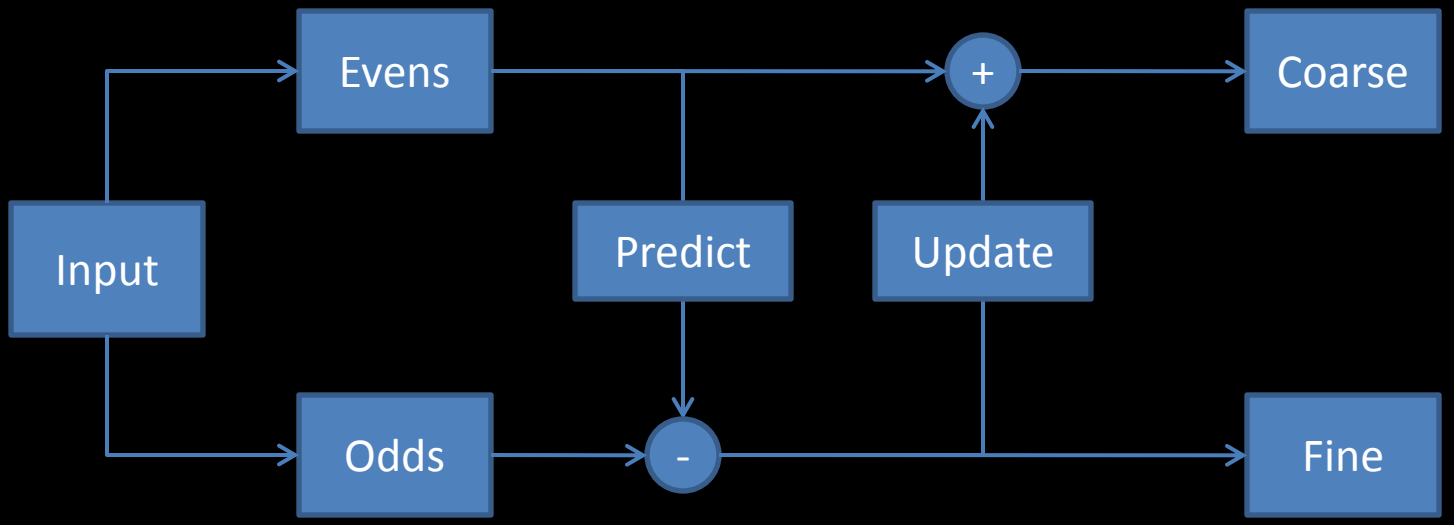
Why did I pick ¼?

¾	¼	-1/8					
-½	1	-½					
-1/8	¼	¾	¼	-1/8			
		-½	1	-½			
		-1/8	¼	¾	¼	-1/8	
				-½	1	-½	
				-1/8	¼	¾	¼
						-½	1

In the coarse layer, each input pixel now counts equally (sum along columns is constant)

¾	¼	-1/8					
-½	1	-½					
-1/8	¼	¾	¼	-1/8			
		-½	1	-½			
		-1/8	¼	¾	¼	-1/8	
				-½	1	-½	
				-1/8	¼	¾	¼
						-½	1

Lifting



- Using an interpolation for the predict filter gives an appropriate fine layer

- The update filter can be computed from the predict filter

Wavelets

- A coarse/fine decomposition that is fast to compute and takes no more memory than the original
- Better or worse than a Laplacian pyramid?