Image Pyramids

Computer Vision

Carnegie Mellon University (Kris Kitani)

What are image pyramids used for?

Image compression

Multi-scale
texture mapping

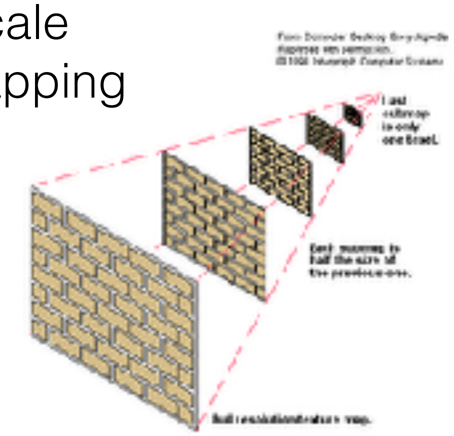
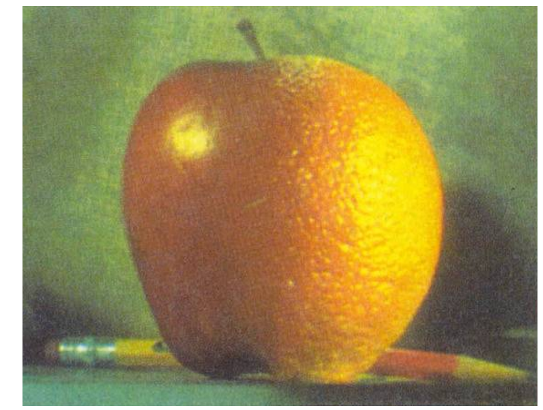
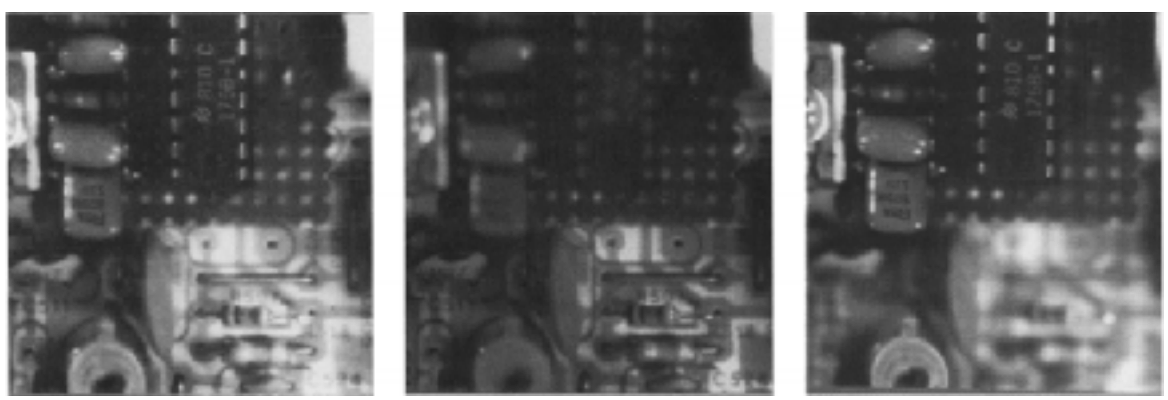


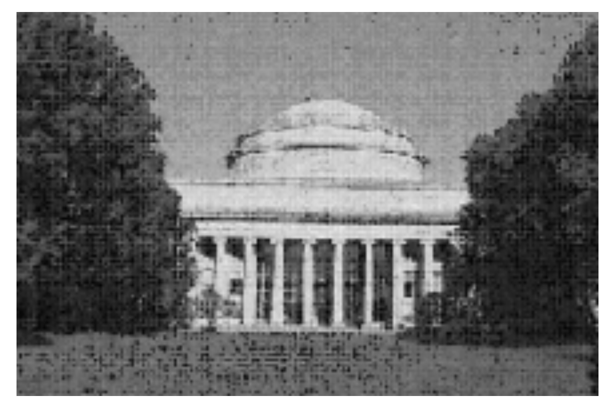
Image blending



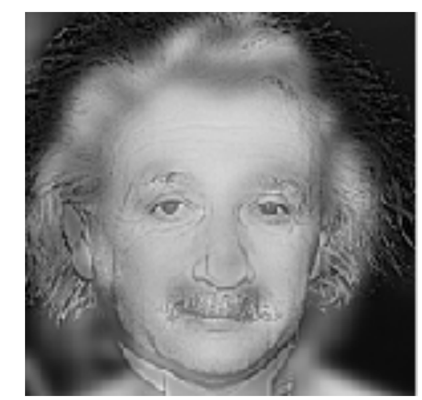
Multi-focus composites



Noise removal

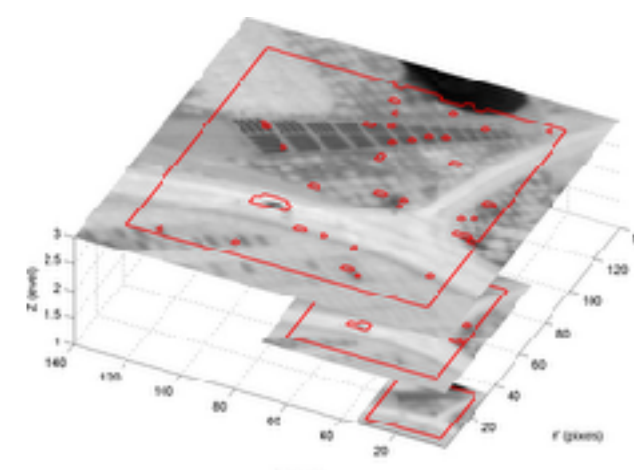


Hybrid images



Multi-scale detection

Multi-scale registration



The Laplacian Pyramid as a Compact Image Code

PETER J. BURT, MEMBER, IEEE, AND EDWARD H. ADELSON

Abstract—We describe a technique for image encoding in which local operators of many scales but identical shape serve as the basis functions. The representation differs from established techniques in that the code elements are localized in spatial frequency as well as in space.

Pixel-to-pixel correlations are first removed by subtracting a low-pass filtered copy of the image from the image itself. The result is a net data compression since the difference, or error, image has low variance and entropy, and the low-pass filtered image may represented at reduced sample density. Further data compression is achieved by quantizing the difference image. These steps are then repeated to compress the low-pass image. Iteration of the process at appropriately expanded scales generates a pyramid data structure.

The encoding process is equivalent to sampling the image with Laplacian operators of many scales. Thus, the code tends to enhance salient image features. A further advantage of the present code is that it is well suited for many image analysis tasks as well as for image compression. Fast algorithms are described for coding and decoding.

does not permit simple sequential coding. Noncausal approaches to image coding typically involve image transforms, or the solution to large sets of simultaneous equations. Rather than encoding pixels sequentially, such techniques encode them all at once, or by blocks.

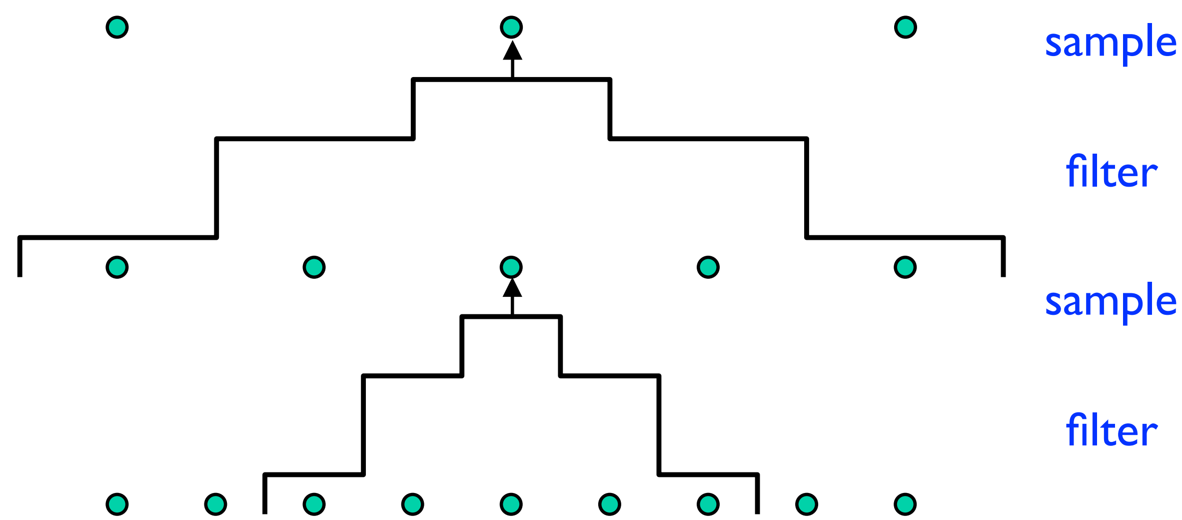
Both predictive and transform techniques have advantages. The former is relatively simple to implement and is readily adapted to local image characteristics. The latter generally provides greater data compression, but at the expense of considerably greater computation.

Here we shall describe a new technique for removing image correlation which combines features of predictive and transform methods. The technique is noncausal, yet computations are relatively simple and local.

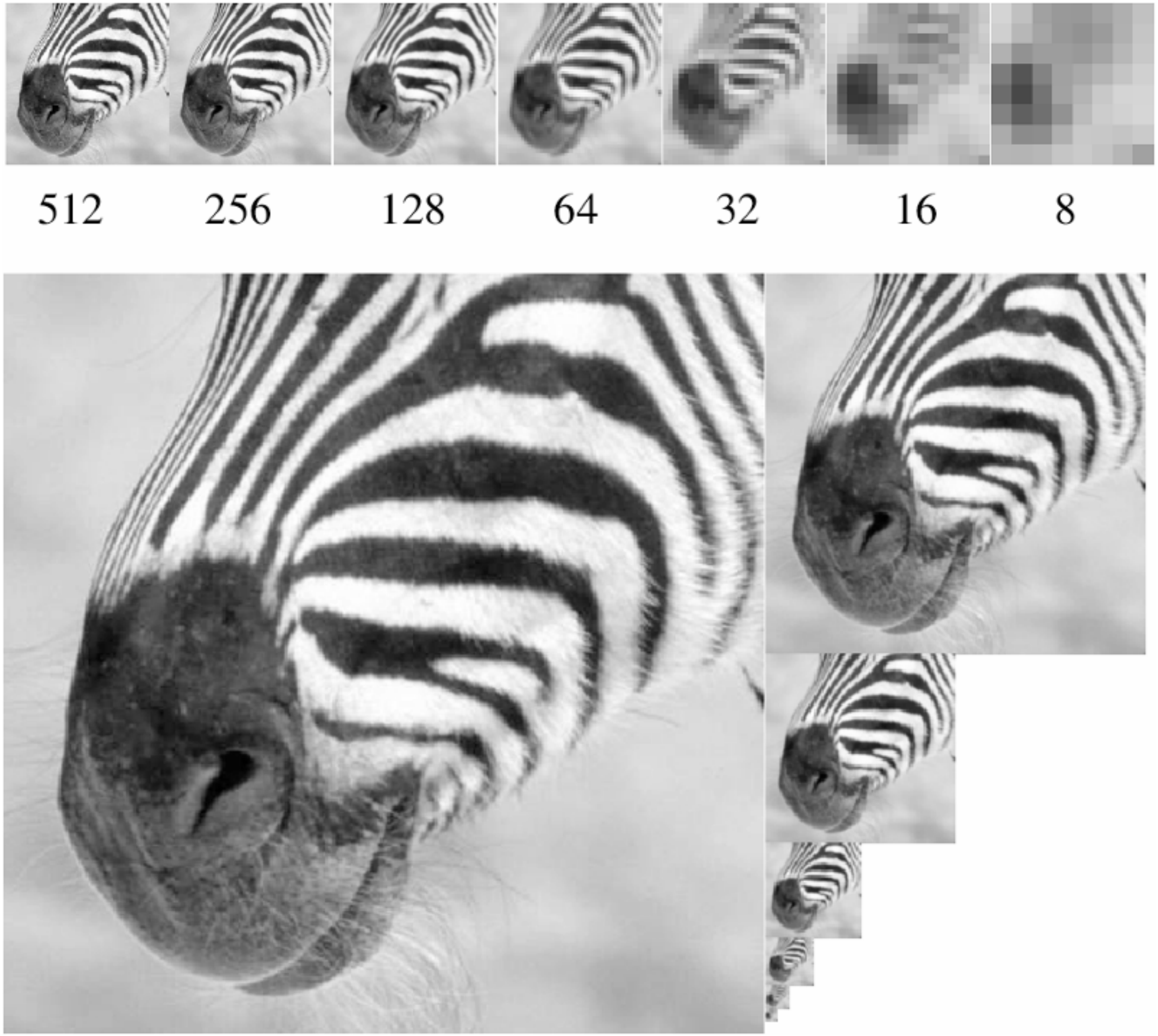
The predicted value for each pixel is computed as a local weighted average, using a unimodal Gaussian-like (or related

Constructing a Gaussian Pyramid

```
repeat
    filter
    subsample
until min resolution reached
```



Whole pyramid is only 4/3 the size of the original image!

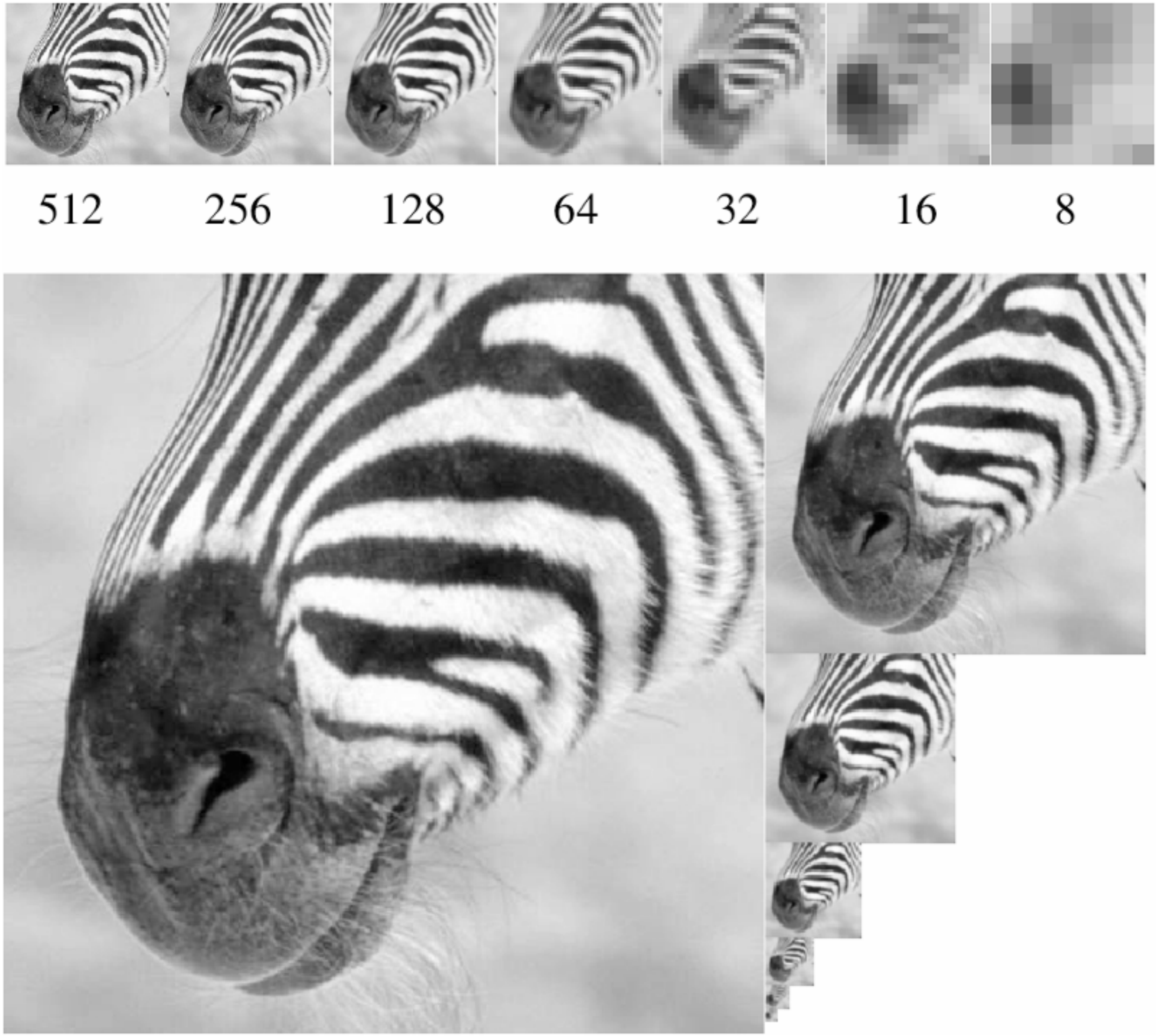


Gaussian pyramid

What happens to the details of the image?

What is preserved at the higher scales?

How would you reconstruct the original image using the upper pyramid?



Gaussian pyramid

What happens to the details of the image?

What is preserved at the higher scales?

Not possible

Level 0

Level 1

What is lost between levels?

What does blurring take away?

Level 0

-

Level 1

=

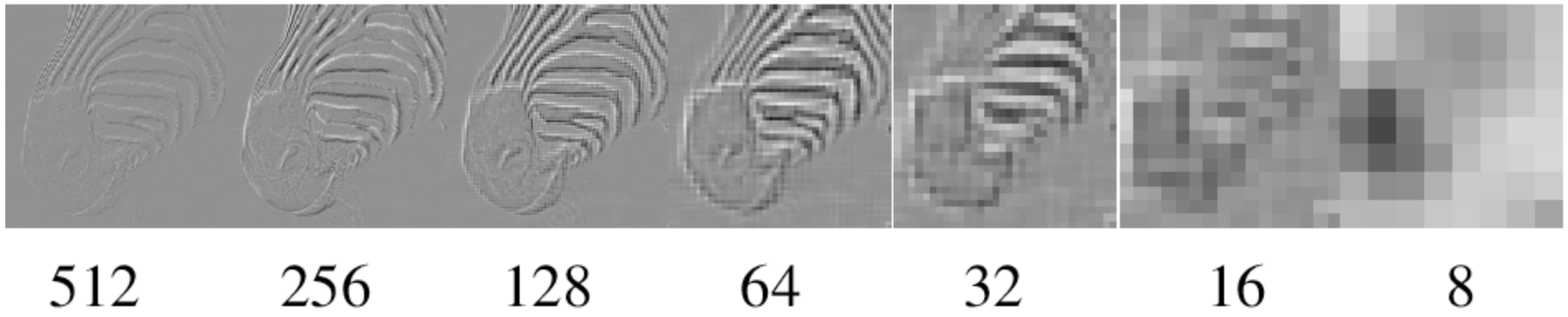
Residual

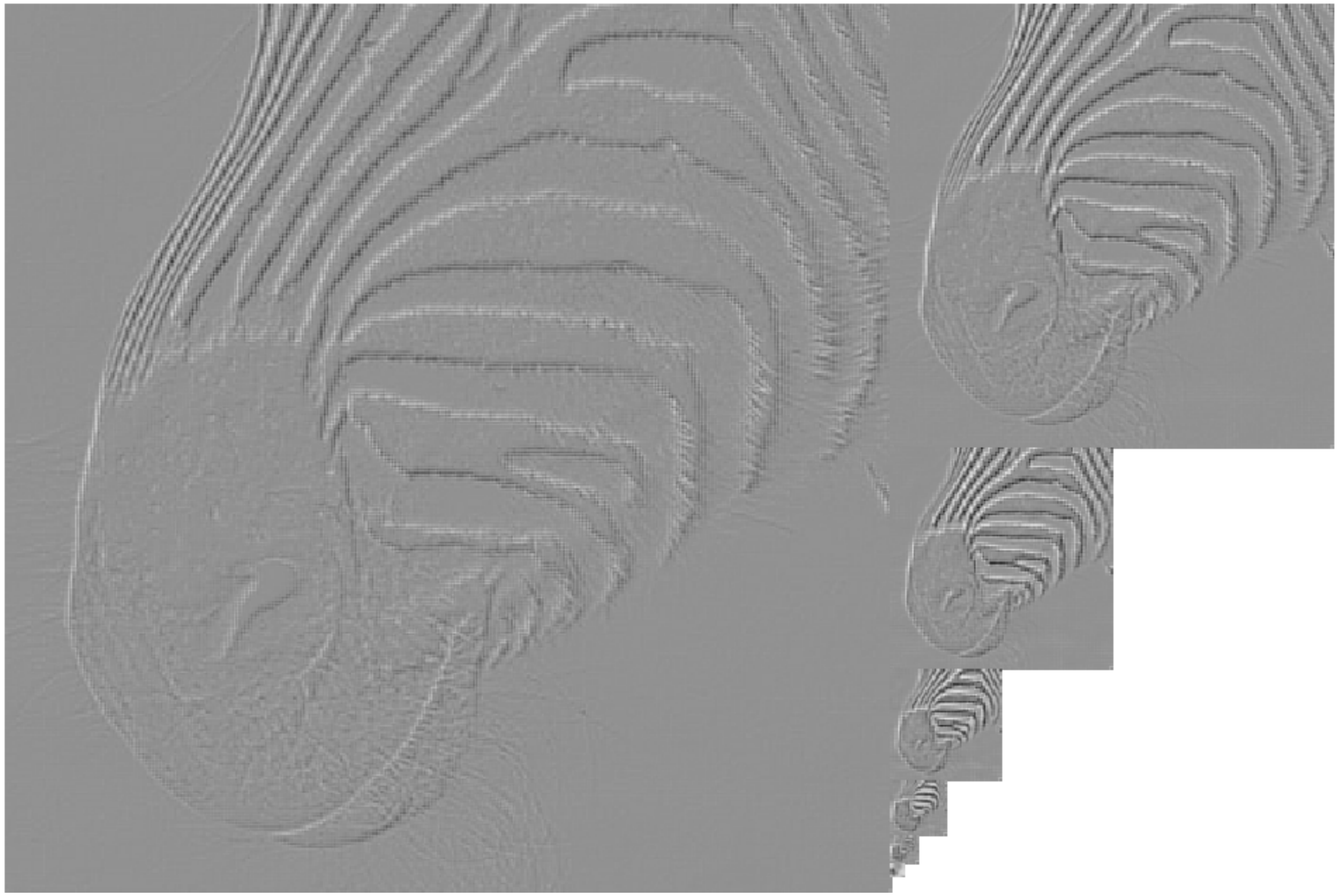
(thrown away by blurring)

(band-pass filter)

We can retain the residuals with a ...

Laplacian pyramid





Retains the residuals (details) between pyramid levels

Can you reconstruct the original image using the upper pyramid?

What exactly do you need to reconstruct the original image?

Partial answer:

Level 0

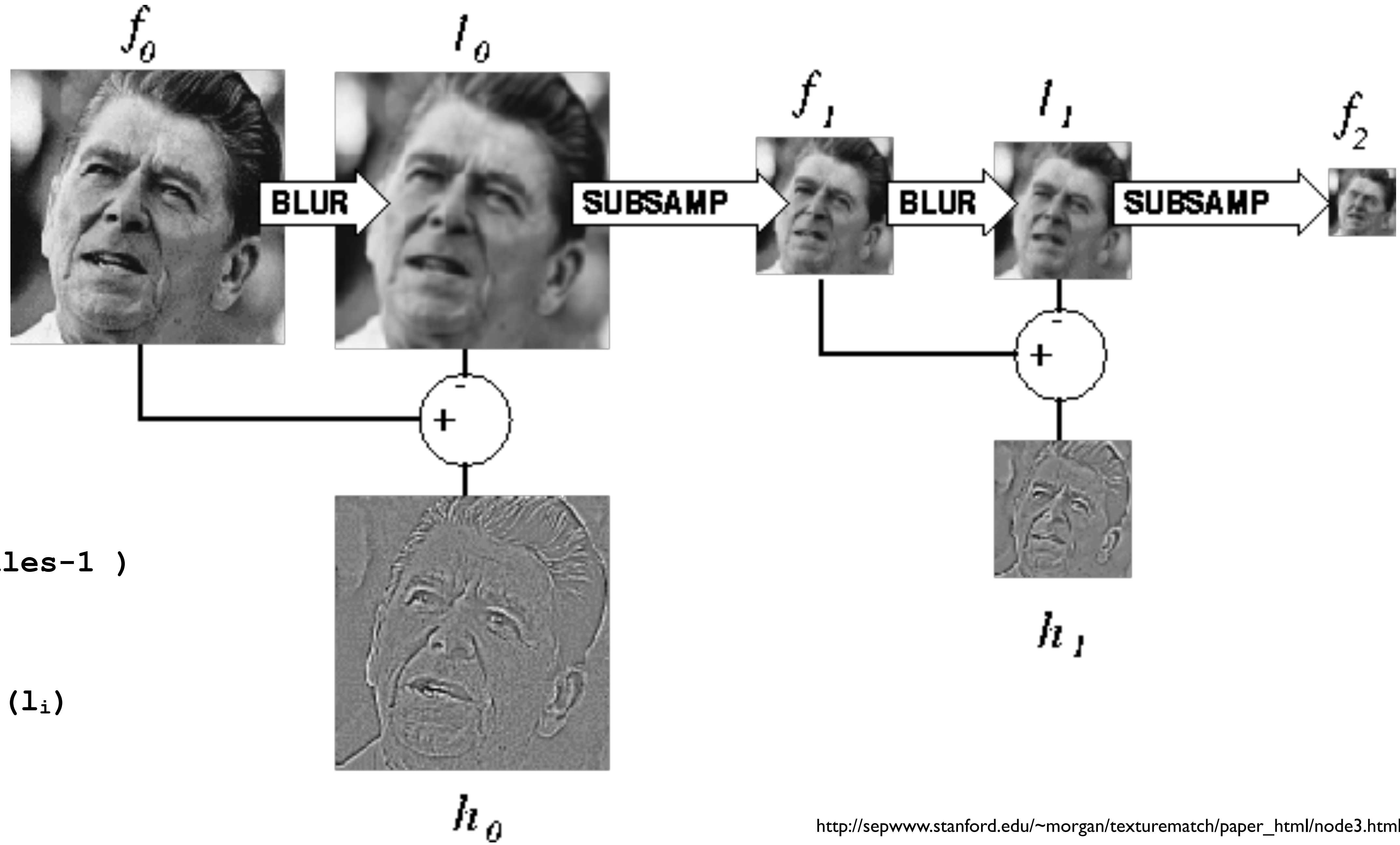
Level 1
(resized)

Level 0

Low frequency
component

High frequency
component

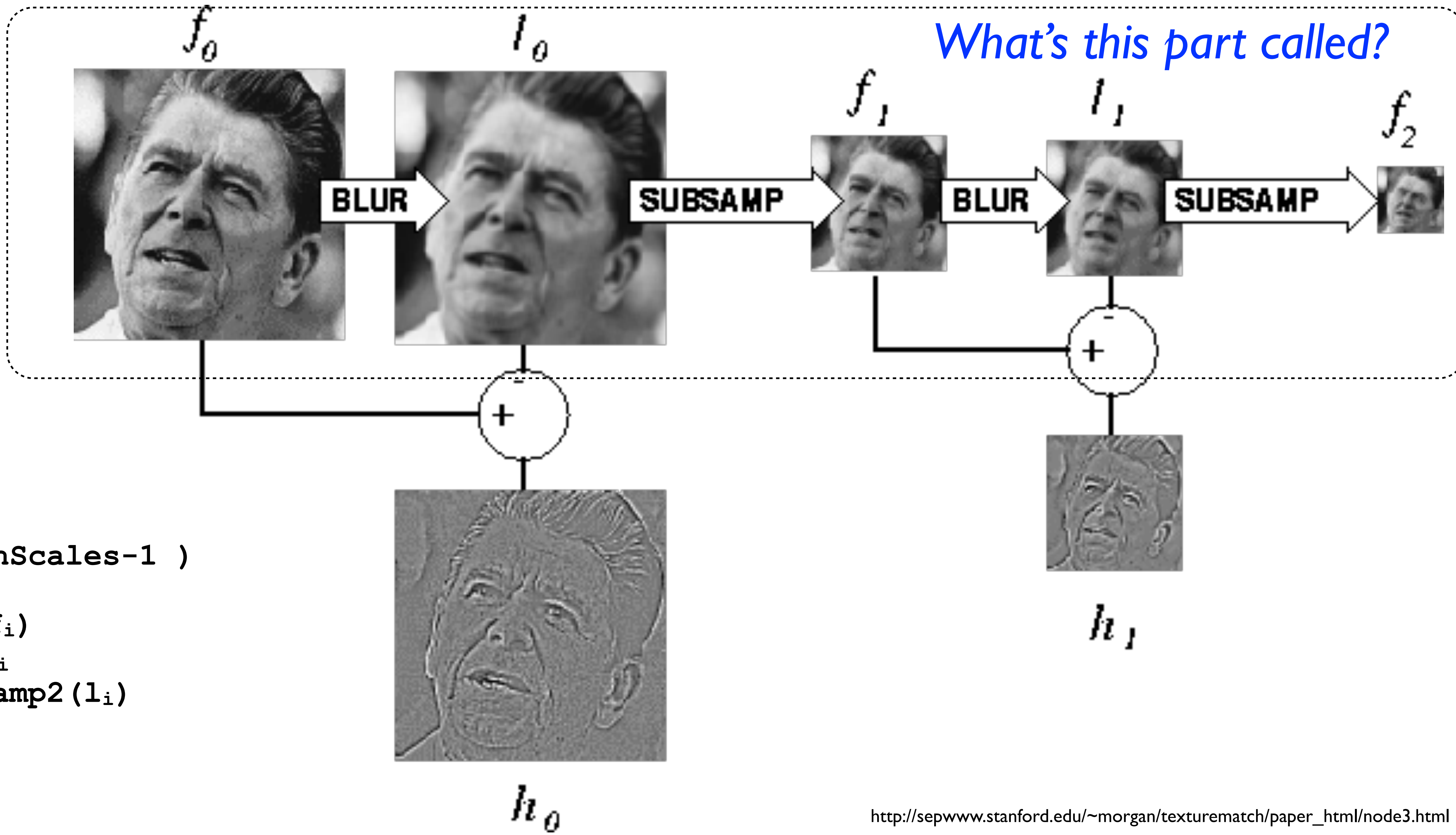
Constructing the Laplacian Pyramid




```
do( i = 0 : nScales-1 )
{
  li = blur(fi)
  hi = li - fi
  fi+1 = subSamp2(li)
}
```

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

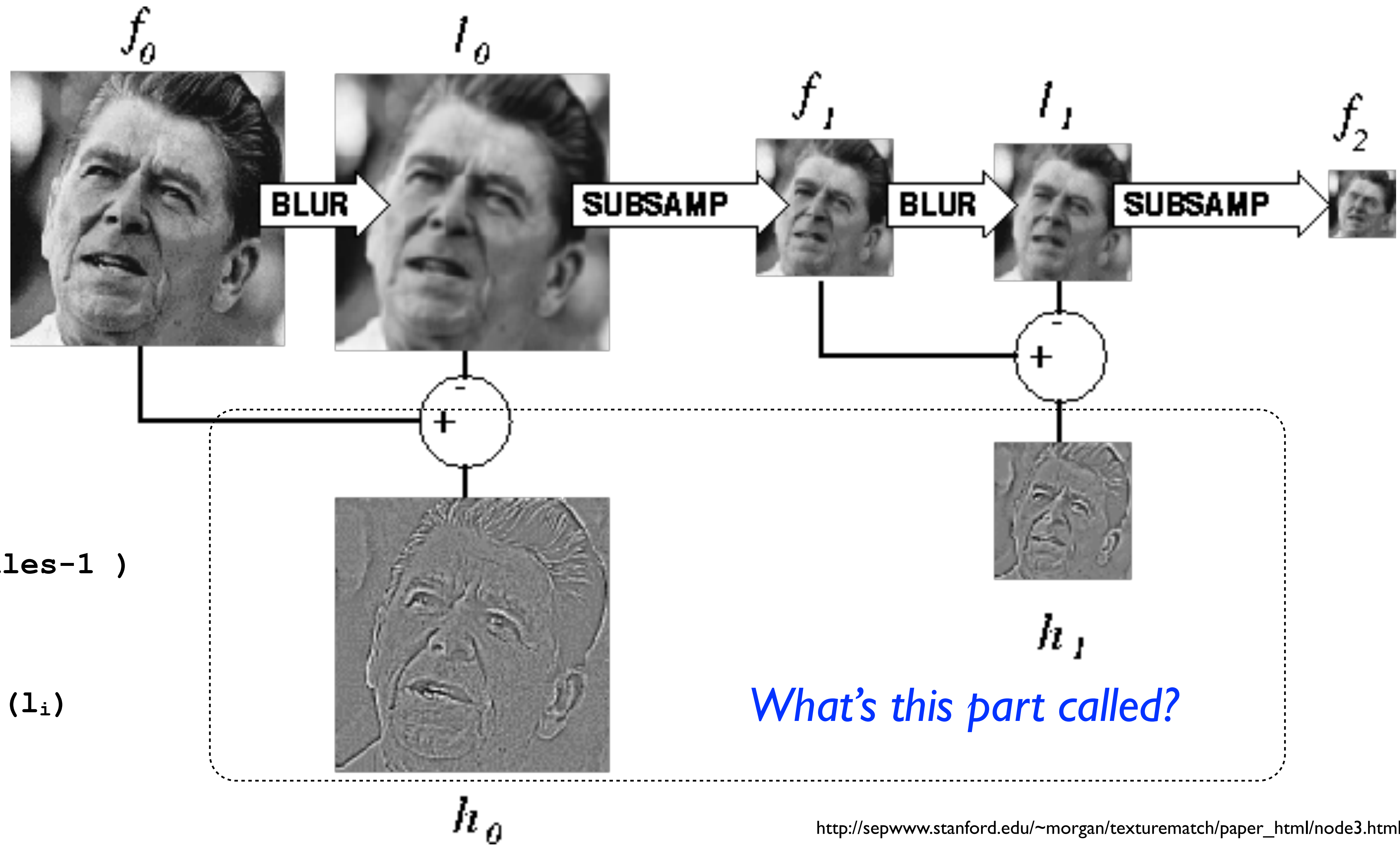
Constructing the Laplacian Pyramid




```
do( i = 0 : nScales-1 )
{
  l_i = blur(f_i)
  h_i = l_i - f_i
  f_i+1 = subSamp2(l_i)
}
```

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

Constructing the Laplacian Pyramid

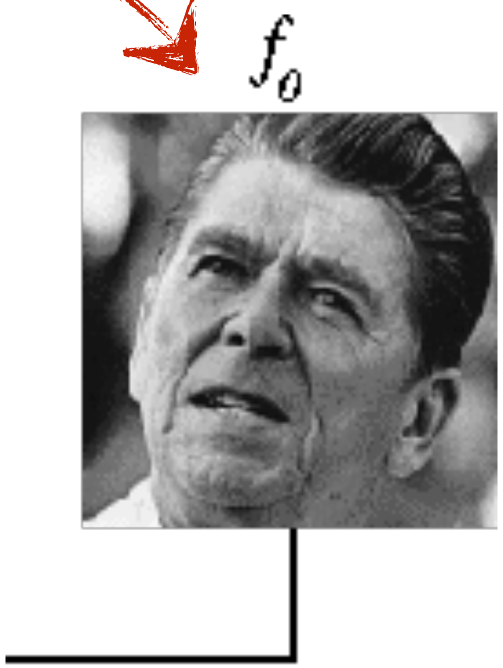



```
do( i = 0 : nScales-1 )
{
  li = blur(fi)
  hi = li - fi
  fi+1 = subSamp2(li)
}
```

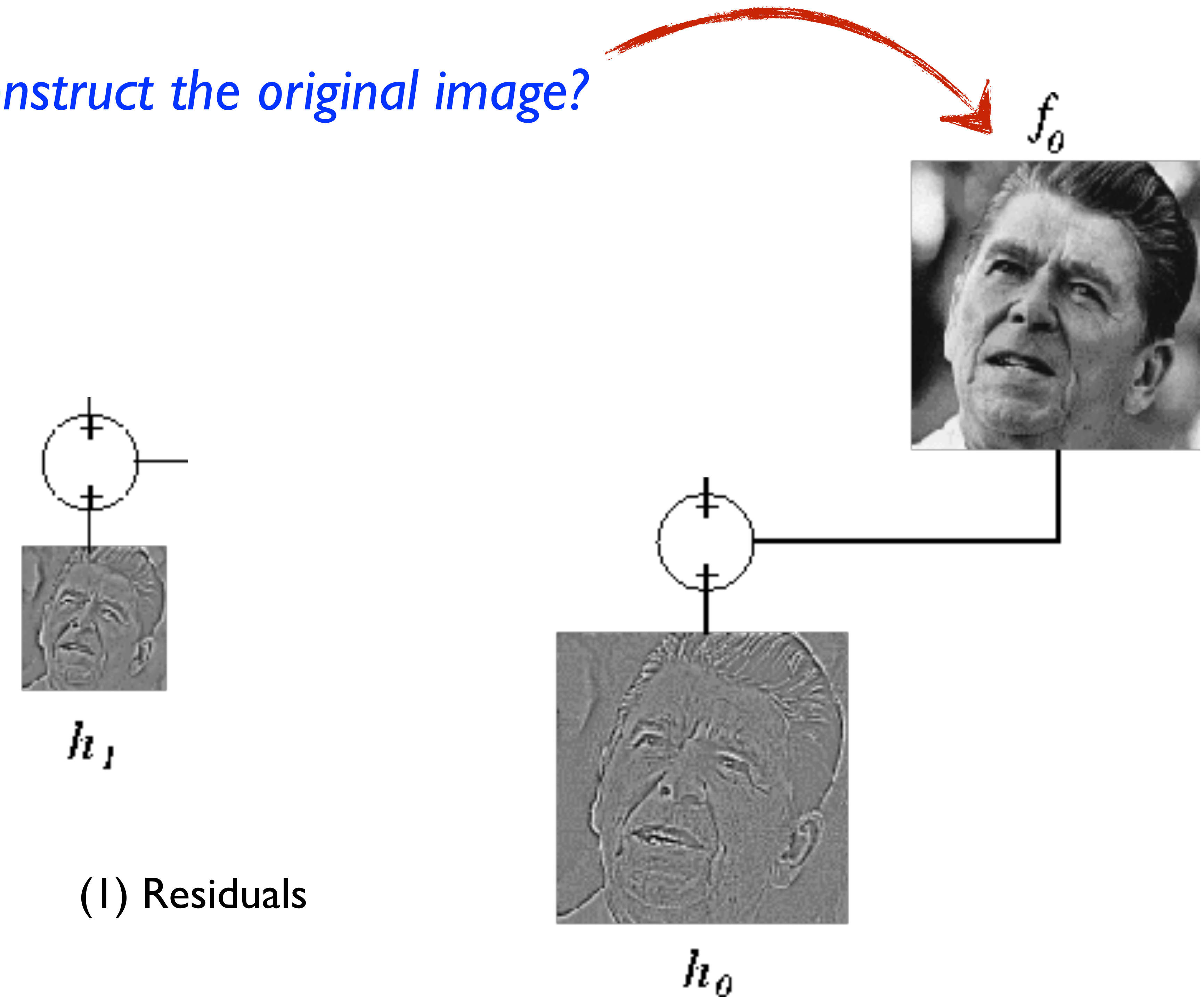
What's this part called?

http://sepwww.stanford.edu/~morgan/texturematch/paper_html/node3.html

What do you need to construct the original image?

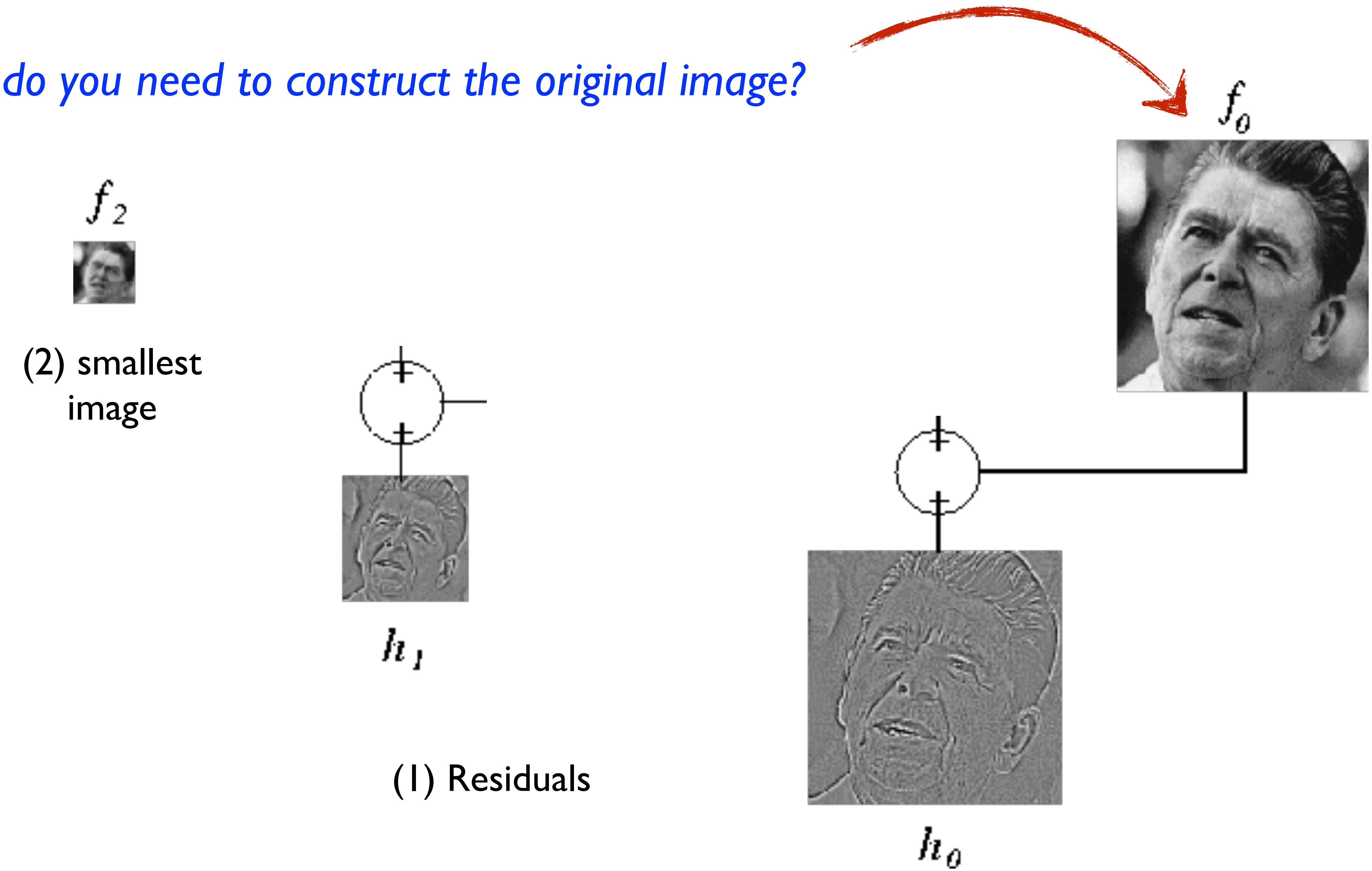


What do you need to construct the original image?

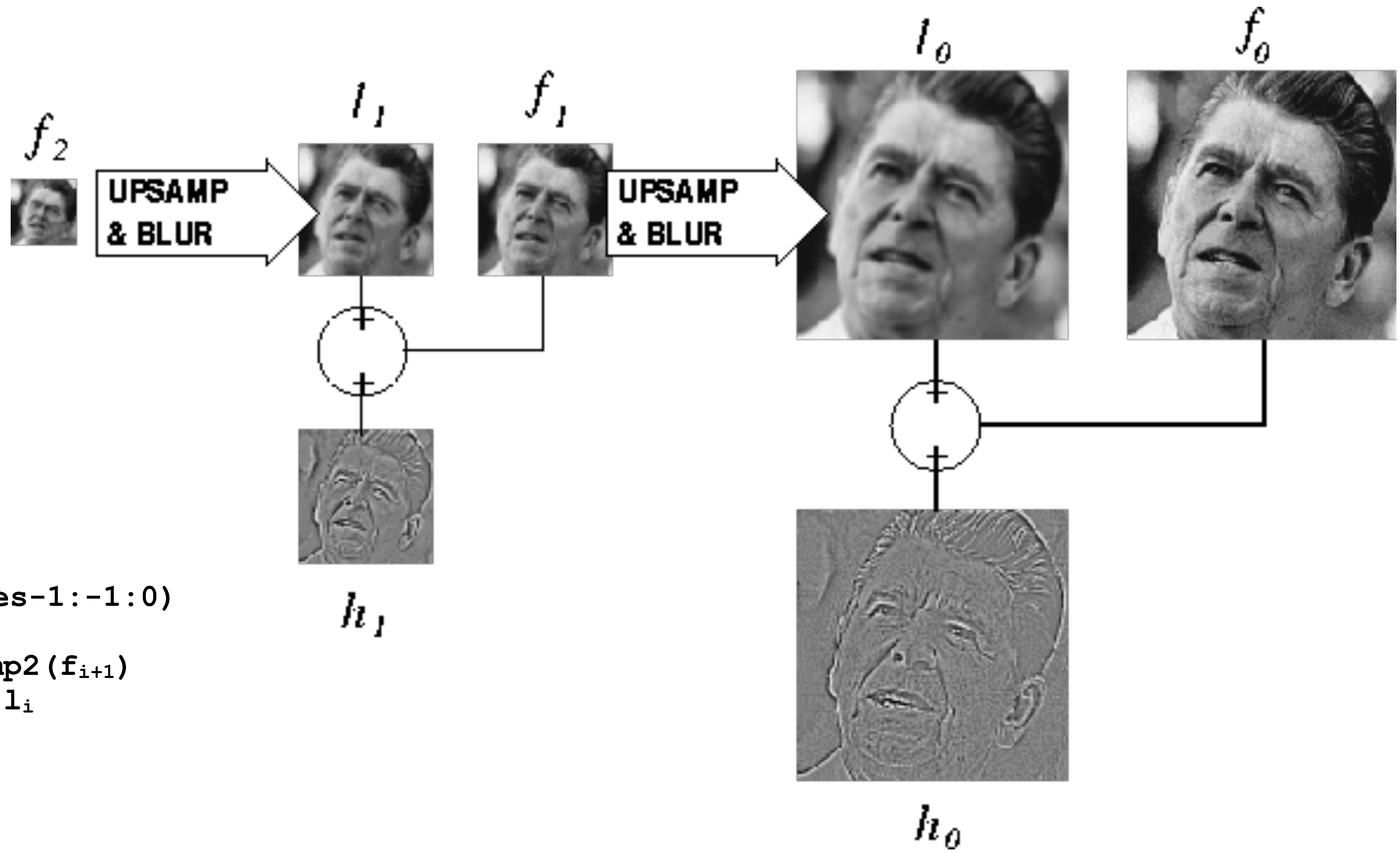


(1) Residuals

What do you need to construct the original image?

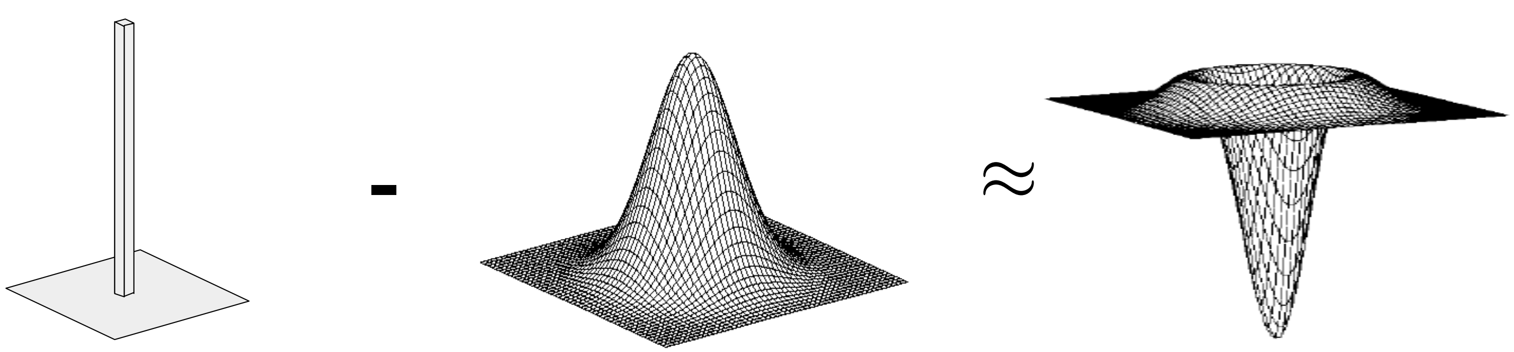


Reconstructing the original image



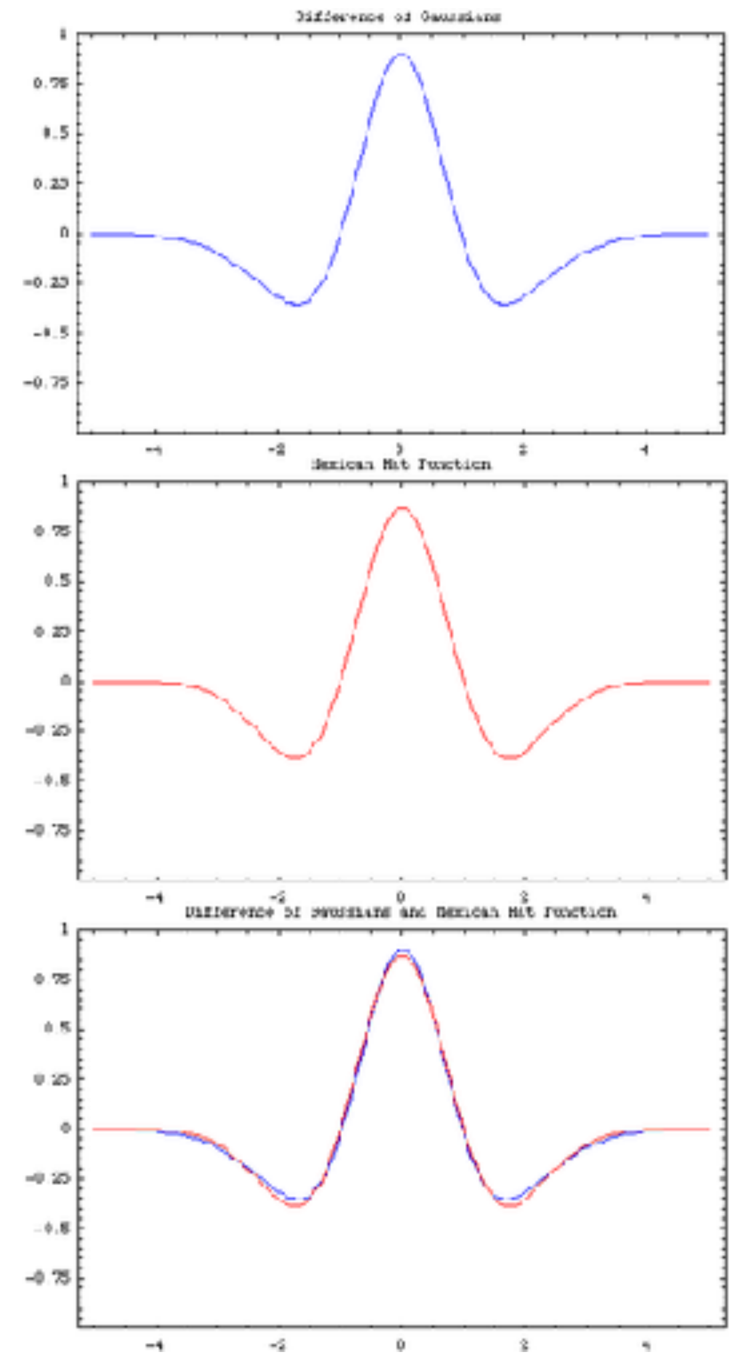

```
do( i = nScales-1:-1:0)
{
    li = upSamp2(fi+1)
    fi = hi + li
}

output: f0
```

Why is it called the Laplacian Pyramid?



unit - Gaussian ≈ Laplacian

Difference of Gaussians approximates the Laplacian



http://en.wikipedia.org/wiki/Difference_of_Gaussians