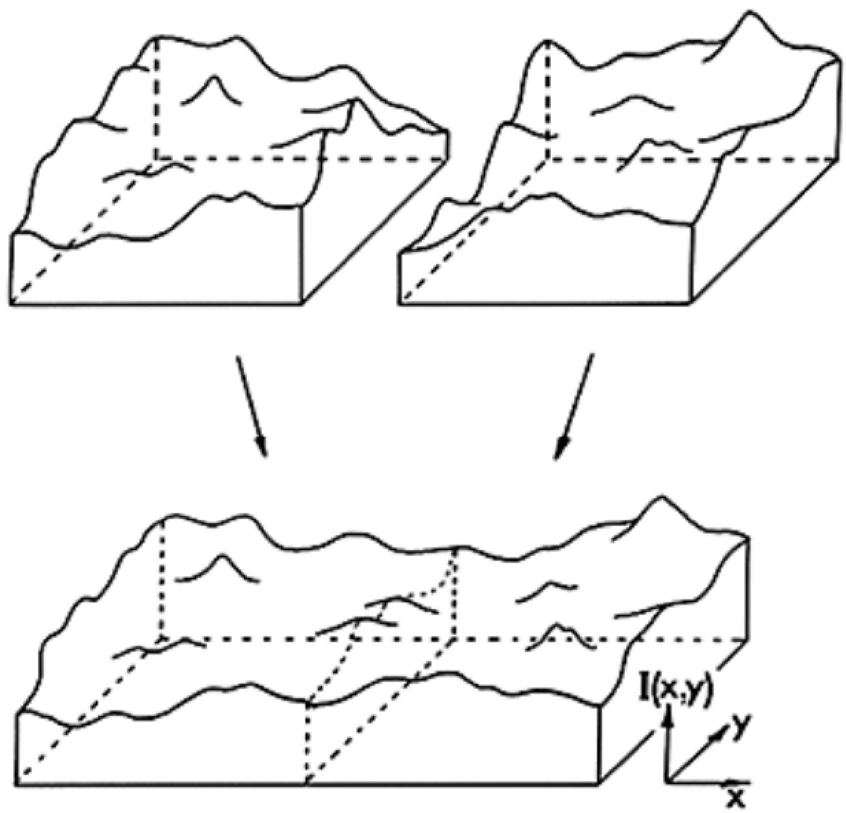
MultiBand Blending:

Splines:

Two images to be joined may be considered as two surfaces,
where the image intensity I(x, y) corresponds to the elevation above the x,y plane.



A technical problem common to all applications of photomosaics is joining two images so that the edge between them is not visible. Even slight differences in image gray level across an extended boundary can make that boundary quite noticeable.

Unfortunately, such gray level differences are frequently unavoidable; they may be due to such factors as differences in camera position or in image processing prior to assembly.

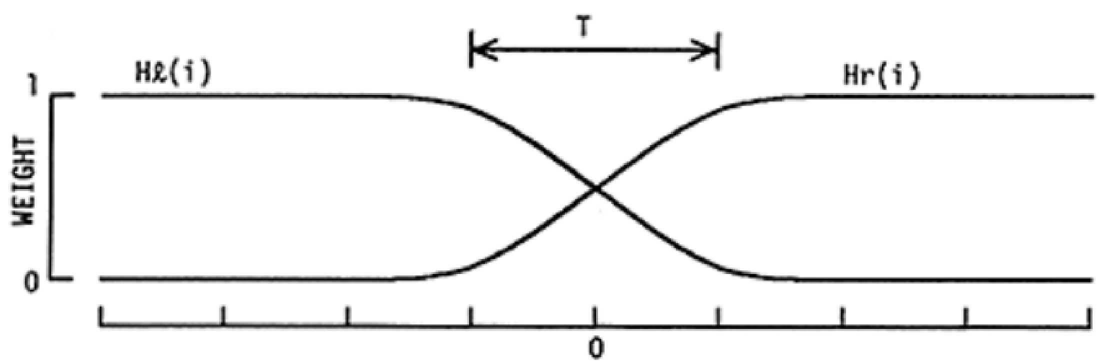
Thus, a technique is required which will modify image gray levels in the vicinity of a boundary to obtain a smooth transition between images.

We will use the term ***image spline*** to refer to digital techniques for making these adjustments. A good image spline will make the seam perfectly smooth, yet will preserve as much of the original image information as possible.

We are concerned with a **weighted average splining technique**.

It is assumed that the images to be joined overlap so that it is possible to compute the gray level value of points within a transition zone as a weighted average of the corresponding points in each image.

Suppose that one image, Fl(i), is on the left and the other, Fr(i), is on the right, and that the images are to be splined at a point i ˆ (expressed in one dimension to simplify notation). Let Hl(i) be a weighting function which decreases monotonically from left to right and let Hr(i) = 1 – Hl(i)

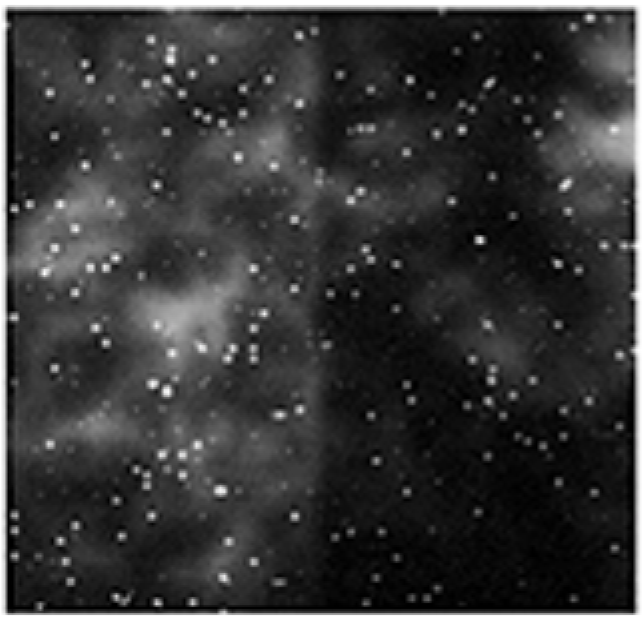


Then, the splined image F is given by

$$F(i) = Hl(i - \hat{i})\, Fl(i) + Hr(i - \hat{i})\, Fr(i).$$

Clearly, the size of the transition zone, relative to the size of image features, plays a critical role in image splining.

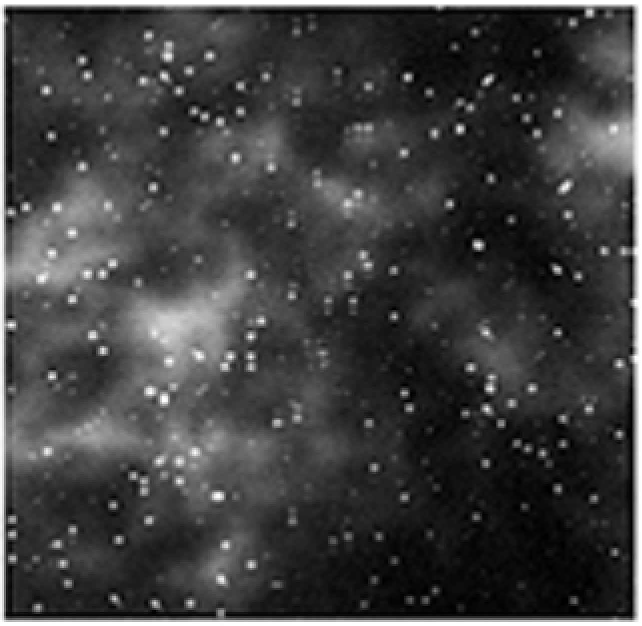
To eliminate a visible edge the transition width should be at least comparable in size to the largest prominent features in the image.



d

On the other hand, to avoid a double exposure effect, the zone should not be much larger than the smallest prominent image features.

e



There is **no choice of T which satisfies both requirements**

These constraints can be stated more precisely in terms of the image spatial frequency content. In particular, a suitable T can only be selected if the images to be splined occupy a relatively narrow spatial frequency band.

As a rough requirement, we may stipulate that T should be comparable in size to the wave-length of the lowest predominant frequency in the image.

If T is smaller than this the spline will introduce a noticeable edge. On the other hand, to avoid a double exposure effect, T should not be much larger than two wavelengths of the highest prominent frequency component in the images.

This ensures that there will not be room for multiple features within the transition zone.

While it is likely that these limits can be exceeded somewhat without noticeable degradation, the general conclusion—that the band width of images to be splined should be roughly one octave—is an important one.

How can images which occupy more than an octave be splined?

The approach proposed here is that such images should first be decomposed into a set of band- pass component images.

A separate spline with an appropriately selected T can then be performed in each band.

Finally, the splined band-pass components are recombined into the desired mosaic image. We call this approach the multi-resolution spline or multiband blending.
In decomposing the image into frequency bands, it is important that the range of frequencies in the original be covered uniformly, although the bands themselves may overlap.

As a practical matter, a set of low-pass filters are applied to generate a sequence of images in which the band limit is reduced from image to image in one-octave steps. (Gaussian)

Band-pass images can then be obtained simply by subtracting each low-pass image from the previous image in the sequence. (Laplacian)

This not only ensures complete coverage of spatial frequencies but means that the final mosaic can be obtained simply by summing the band-pass component images.

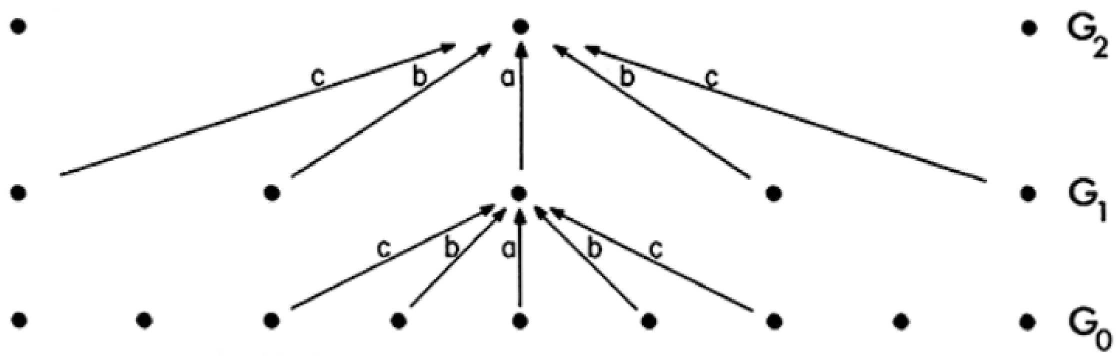
Pyramids

Gaussian Pyramids:

A sequence of low-pass filtered images G_0 , G_1 , . . ., G_N can be obtained by repeatedly convolving a small weighting function with an image.

With this technique, image sample density is also decreased with each iteration so that the bandwidth is reduced in uniform one-octave steps.

Sample reduction also means that the cost of computation is held to a minimum.



1. The value of each node in the next row, G_1 , is computed as a weighted average of a 5 x 5 subarray of G_0 nodes, as shown.
2. Nodes of array G_2 are then computed from G_1 using the same pattern of weights.
3. The process is iterated to obtain G_2 from G_1 , G_3 from G_2 and so on.
4. The sample distance is doubled with each iteration so that successive arrays are half as large in each dimension as their predecessors.
5. If we imagine these arrays stacked one above the other, the result is the tapering data structure known as a pyramid.
6. If the original image measures 2^N + 1 by 2^N + 1, then the pyramid will have N + 1 levels.

To achieve this, the REDUCE process is used:

$$G_l = \text{REDUCE } [G_{l-1}],$$

by which we mean:

$$G_l(i,j) = \sum \sum_{m,n=1}^{5} w(m,n) G_{l-1}(2i+m, 2j+n).$$

The pattern of weights w(m, n) used to generate each pyramid level from its predecessor is called the generating kernel.

These weights are chosen subject to four constraints:

1. For computational convenience, the generating kernel is separable, w(m, n) = ŵ (m) ŵ (n).
2. The one dimensional function ŵ is symmetric, ŵ (0) = a, ŵ (- 1) = ŵ (1) = b, and ŵ (- 2) = ŵ (2) = c.
3. ŵ is normalized, **a + 2b + 2c = 1**.
4. Each level l node must contribute the same total weight to level l + 1 nodes: thus, **a + 2c = 2b**.
5. Now, combining constraints, we find that a may be considered a free variable, while **b = 1/4 and c = 1/4 — a/2.**
6. If we consider **a = 0.4**, the weighting function looks like a Gaussian Approximation, and thus is called the Gaussian Pyramid.

- Pyramid construction is equivalent to convolving the image with a set of Gaussian-like functions to produce a corresponding set of filtered images.
- Because of the importance of the multiple filter interpretation, we call this sequence of images G 0 ,G 1 , . . ., G N as the Gaussian pyramid.

Laplacian Pyramids:

The Gaussian pyramid is a set of low-pass filtered images. In order to obtain the band-pass images required for the multiresolution spline we subtract each level of the pyramid from the next lowest level.

Because these arrays differ in sample density, it is necessary to interpolate new samples between those of a given array before it is subtracted from the next lowest array.

Interpolation can be achieved by reversing the REDUCE process. We shall call this an EXPAND operation.

$$G_{l,k} = \mathrm{EXPAND}[G_{l,k\text{-}1}].$$

By EXPAND, we mean

$$G_{l,k}(i,j) = 4\sum \sum_{n,n=-2}^{2} G_{l,k-1}\left(\frac{2i+m}{2}, \frac{2j+n}{2}\right).$$

*Only terms for which (2i + m)/2 and (2j + n)/2 are integers contribute to the sum

We now define a sequence of band-pass images $L_0, L_1, \ldots, L_N$. For $0 < l < N$,

$$L_l = G_l - \mathrm{EXPAND}[G_{l+1}] = G_l - G_{l+1,1}.$$

Because there is no higher level array to subtract from G_N, we define **$L_N = G_N$** .

- Just as the value of each node in the Gaussian pyramid could have been obtained directly by convolving the weighting function W_l with the image, each node of L_i can be obtained directly by convolving $W_i - W_{i+1}$ with the image.
- This difference of Gaussian-like functions resembles the Laplacian operators commonly used in the image processing, so we refer to the sequence $L_0, L_1, \ldots, L_N$ as the Laplacian pyramid.

Summation Property:

- The steps used to construct the Laplacian pyramid may be reversed to recover the original image G 0 exactly:
- The top pyramid level, L N , is first expanded and added to L N-1 to recover G N-1
- This array is then expanded and added to L N-2 torecover G N-2 , and so on.

Alternatively, we may write

$$G_0 = \sum_{l=0}^{N} L_{l,l}.$$

The expand and sum procedure will be used to construct a mosaic image from its set of splined band-pass components.

Boundary Conditions

In both the REDUCE and EXPAND operations, special attention must be given to edge nodes.
For example, when a REDUCE is performed, the generating kernel for an edge node at level G l=1 extends beyond the edge of level G l by two nodes. Therefore, before the REDUCE (or EXPAND) is performed, G l is augmented by two rows of nodes on each side.

Values are assigned to these nodes by reflection and inversion across the edge node. Thus, if G l (0, j) is a node on the left edge of G l , we set

$$G_l(-1, j) = 2G_l(0, j) - G_l(1, j),$$

and

$$G_l(-2, j) = 2G_l(0, j) - G_l(2, j).$$

This treatment of boundaries has the effect of extrapolating the images in such a way that the first derivative is constant at the edge node (the second derivative is zero).

Multiband Spline

The multiresolution spline algorithm may be defined rather simply in terms of
the basic pyramid operations.

1. Splining Overlapped Images

1. Laplacian pyramids LA and LB are constructed for images A and B respectively.
2. A third Laplacian pyramid LS is constructed by copying nodes from the left half of LA to the corresponding nodes of LS, and nodes in the right half of LB to the right half of LS.
3. The splined image S is obtained by expanding and summing the levels of LS

Nodes along the center line of LS are set equal to the average of corresponding LA and LB nodes.

The center line for level l of a Laplacian pyramid is at i = 2 N-1 . Thus, for
all i, j, l,

$$LS_l(i,j) = \begin{cases} LA_l(i,j) & if \quad 1 < 2^{N-1} \\ (LA_l(i,j) + LB_l(i,j))/2 & if \quad i = 2^{N-1} \\ LB_l(i,j) & if \quad i > 2^{N-1} \end{cases}$$

2. Splining Regions of Arbitrary Shape

The steps outlined above can be generalized for constructing a mosaic from image regions of arbitrary shape. Again, we assume that the regions to be splined are contained in images A and B and that these completely overlap.

As before, nodes of the Laplacian pyramids LA and LB for the component images will be combined to form the Laplacian pyramid LS of the image mosaic S. We introduce an additional pyramid structure in order to determine which nodes of LS should be taken from LA, which from LB, and which should be an average of the two.

Let R be a binary image of the same size as A and B, in which all pixels inside the region of A to be splined with B are 1 and all those outside the region are 0. GR is the Gaussian Pyramid formed from R.

1. Build Laplacian pyramids LA and LB for images A and B respectively.
2. Build a Gaussian pyramid GR for the region image R.
3. Form a combined pyramid LS from LA and LB using nodes of GR as weights. That is, for each l, i and j: LS(i, j) = GR(i, j)LA(i, j) + (1 - GR(i, j))LB(i, j).
4. Obtain the splined image S by expanding and summing the levels of LS.