

## HOW TO PAN-SHARPEN IMAGES USING THE GRAM-SCHMIDT PAN-SHARPEN METHOD – A RECIPE

T. Maurer

Esri, 380 New York Street, Redlands, CA 92373, USA - tmaurer@esri.com

**KEY WORDS:** High Resolution Imagery, Gram-Schmidt Pan-Sharpener

### ABSTRACT:

Since its publication in 1998 (Laben and Brower, 2000), the Gram-Schmidt pan-sharpen method has become one of the most popular algorithms to pan-sharpen multispectral (MS) imagery. It outperforms most other pan-sharpen methods in both maximizing image sharpness and minimizing color distortion. It is, on the other hand, also more complex and computationally expensive than most other methods, as it requires forward and backward transforming the entire image. Another complication is the lack of a clear recipe of how to compute the sensor dependent MS to Pan weights that are needed to compute the simulated low resolution pan band. Estimating them from the sensor's spectral sensitivity curves (in different ways), or using linear regression or least square methods are typical candidates which can include other degrees of freedom such as adding a constant offset or not. As a result, most companies and data providers do it somewhat differently.

Here we present a solution to both problems. The transform coefficients can be computed directly and in advance from the MS covariance matrix and the MS to Pan weights. Once the MS covariance matrix is computed and stored with the image statistics, any small section of the image can be pan-sharpened on the fly, without having to compute anything else over the entire image. Similarly, optimal MS to Pan weights can be computed directly from the full MS-Pan covariance matrix, guaranteeing optimal image quality and consistency.

### 1. INTRODUCTION

#### 1.1 Multispectral pan-sharpening

One of the common problems in remote sensing and high resolution image processing is the need for somehow fusing lower resolution, multispectral (MS) bands such as Red, Green, Infrared, etc., with the single, higher resolution Pan band. Ideally, the MS bands can be up-sampled to the full Pan resolution without altering their spectral properties. In practice however, this is hard to achieve. Many pan-sharpening algorithms exist, differing in the degree, to which they maximize the sharpness, and at the same time minimize the color or spectral distortion of the pan-sharpened output image. Older and simpler algorithms, such as the Intensity-Hue-Saturation (IHS) transformation or the Brovey method only work for up to three MS bands, while more modern algorithms work for four or more bands. For a recent survey of pan-sharpening methods see for instance (Amro, 2011). Since its publication in 1998 (and patented by Kodak in 2000), the Gram-Schmidt pan-sharpen method has become one of the most widely used high quality methods. Many variations and enhancements have been studied and published, e.g. (Aiazzi, 2007). The Gram-Schmidt method is also offered by companies such as Esri, ENVI, and others, in their software packages.

#### 1.2 The Gram-Schmidt pan-sharpen method

The Gram-Schmidt pan-sharpen method in a nutshell:

- 1) Compute a simulated low resolution Pan band as a linear combination of the n MS bands:

$$Pan_{sim} = \sum_{k=1}^n w_k MS_k \quad (1)$$

- 2) Treating every band as a high dimensional vector and starting with the simulated pan band as the first vector, make all bands orthogonal using the Gram-Schmidt vector orthogonalization, or, more precise, a modified version of it. For the Gram-Schmidt pan-sharpening, both the incoming bands and the arguments of the scalar products are made dc free first (get their means subtracted). This turns the original Gram-Schmidt scalar products into covariances. The iterative procedure stays the same: Compute the angle between the Red band and the Pan band, rotate the Red band to make it orthogonal to the Pan band. In the next step, compute the angles between the Green band and the Pan band and rotated Red band, then rotate the Green band and make it orthogonal to both the Pan band and rotated Red band. And so forth. This Gram-Schmidt forward transform de-correlates the bands.
- 3) Replace the low resolution simulated Pan band by the gain and bias adjusted high resolution Pan band. Up-sample all MS bands accordingly.
- 4) Reverse the forward Gram-Schmidt transform using the same transform coefficients, but on the high resolution bands. The result of this backward Gram-Schmidt transform is the pan-sharpened image in high resolution.

There are two practical problems here: First, how do we compute the MS to Pan weights needed for step 1? Especially if we don't know the sensor's spectral sensitivity curves or we even don't know the sensor. Second, how can we pan-sharpen a small section of the (potentially huge) image without having to forward and backward transform the entire image? We will address the second question first.

## 2. THE RECIPE

### 2.1 Compute the Gram-Schmidt transform coefficients from the MS covariance matrix and the MS to Pan weights

Here we show that the Gram-Schmidt transform coefficients or matrix (let's call it the GS matrix), which is usually computed iteratively together with the forward transformed (or "rotated") MS bands, can be computed directly from the MS covariance matrix and the MS to Pan weights. We start by rewriting the original equations (10)-(12) in (Laben and Brower, 2000). First, for simplicity, we drop the subtracted mean terms, or, in other words, assume the incoming bands are already dc free. We will show further below that this has no effect on the mathematical result. Then, omitting the pixel indexes for brevity, and explicitly taking the first band (0) as the simulated Pan band and the remaining bands as MS bands 1 to  $n$ , we get

$$\begin{aligned}
 Pan'_{sim} &= Pan_{sim} \\
 MS'_1 &= MS_1 - \frac{\langle Pan'_{sim} | MS_1 \rangle}{\langle Pan'_{sim} | Pan'_{sim} \rangle} Pan'_{sim} \\
 MS'_2 &= MS_2 - \frac{\langle Pan'_{sim} | MS_2 \rangle}{\langle Pan'_{sim} | Pan'_{sim} \rangle} Pan'_{sim} \\
 &\quad - \frac{\langle MS'_1 | MS_2 \rangle}{\langle MS'_1 | MS'_1 \rangle} MS'_1 \\
 &\quad \vdots \\
 MS'_n &= MS_n - \frac{\langle Pan'_{sim} | MS_n \rangle}{\langle Pan'_{sim} | Pan'_{sim} \rangle} Pan'_{sim} \\
 &\quad - \sum_{k=1}^{n-1} \frac{\langle MS'_k | MS_n \rangle}{\langle MS'_k | MS'_k \rangle} MS'_k
 \end{aligned} \tag{2}$$

The apostrophe denotes the transformed or rotated band, and  $\langle a | b \rangle$  means the covariance of bands  $a$  and  $b$ . Note that the coefficients above depend on the transformed bands and can only be computed after the bands with lower indexes have been transformed. Therefore our next step is to substitute equation (1) into the above Gram-Schmidt transform equations. We compute the coefficients one column at a time, from left to right, in increasing difficulty. All transform coefficients  $GS_{kl}$  can be written in the form:

$$GS_{kl} = \frac{\langle MS'_k | MS_l \rangle}{\langle MS'_k | MS'_k \rangle} = \frac{\sum a_i^l \langle MS_i | MS_k \rangle}{\sum \sum a_i^l a_j^l \langle MS_i | MS_j \rangle} \tag{3}$$

With their weight vectors  $a^l$  computed also iteratively only taking turns with the transform coefficients, in vector notation, as:

$$\begin{aligned}
 a^0 &= w \\
 a^1 &= e_1 - GS_{10} a^0 \\
 a^2 &= e_2 - GS_{20} a^0 - GS_{21} a^1 \\
 &\vdots \\
 a^n &= e_n - \sum_{i=0}^{n-1} GS_{ni} a^i
 \end{aligned} \tag{4}$$

Here  $a^0$  is the MS to Pan weight vector,  $a^1$  the MS to rotated red band weight vector,  $a^2$  the MS to rotated green band weight vector, etc. In other words, from the initial MS to Pan weight vector  $a^0$  we calculate the first Gram-Schmidt transform coefficient  $GS_{10}$ , then Gram-Schmidt transform this initial weight vector to compute the next weight vector  $a^1$  which allows us to compute the Gram-Schmidt transform coefficients for the next row and so on.

The result is a function that takes as input the MS covariance matrix and the MS to Pan weights, and outputs the Gram-Schmidt transform coefficients. The MS covariance matrix can be computed once for the MS raster and stored as part of the raster statistics. The MS to Pan weights can be assumed to be mainly sensor dependent and having been computed in advance. The direct computation of the GS matrix allows to pan-sharpen any small portion of an image without having to compute any other property over the entire image first. The resulting pan-sharpened image is exactly the same as if globally computed. Another practical use case is that the MS to Pan weights can be varied and only an area of interest pan-sharpened on-the-fly, again without having to compute anything globally. Last but not least the direct computation of the GS matrix allows for analyzing the propagation of errors from either the MS to Pan weights or the MS covariance to the GS matrix.

In the beginning of this derivation we had simply dropped the band mean offsets, in contrast to Laben and Brower. This does not make a difference for the end result. The MS covariance does not depend on the band mean. The covariance of any two bands  $a$ ,  $b$ , with one of the two bands constant is 0, which means that adding or subtracting constant mean terms cancels out in the Gram-Schmidt transform coefficients. With the GS matrix being independent of the band means, adding or subtracting these means in the above equations only results in constant, position independent offsets in the rotated bands. When doing the backward transform, the final (forward and backward transformed) MS bands will have the same band means as the original MS bands, as long as the substituted high resolution Pan band has the same mean as the low resolution simulated one. This is the case, thanks to the gain bias transform of the high resolution Pan band before it can replace the simulated one (equations (1)-(3), (Laben and Brower, 2000)). For the same reason any constant offset in the simulated Pan band would have no effect on the result.

### 2.2 Compute optimal MS to Pan weights

Now we come to the second topic of interest, the computation of optimal MS to Pan weights. Optimal here means optimal for

Gram-Schmidt pan sharpening. As it turns out, we get this as a by-product of the above calculations. Part of the first column Gram-Schmidt transform coefficients was

$$\langle Pan'_{sim} | MS_k \rangle = \sum_{i=1}^n w_i \langle MS_i | MS_k \rangle \quad (5)$$

We had used this formula to compute the left side from the (known) MS to Pan weights and the MS covariance matrix. All we do now is replace the simulated Pan band on the left by the low pass filtered and down sampled real high resolution Pan band, compute the left side, and solve for the weights  $w$ . The MS covariance matrix is a real symmetric square matrix with its size equal to the number of MS bands and usually directly invertible. So we get, with index  $ds$  for down sampled, and  $C$  for covariance matrix,

$$w_k = \sum_{i=1}^n C_{ki}^{-1} \langle Pan_{ds} | MS_i \rangle \quad (6)$$

In case  $C$  should turn out to be singular (not directly invertible), this equation can also be solved in a least square fashion by Singular Value Decomposition (SVD). Decompose  $C$  into its SVD components, compute its pseudo inverse, and then solve (6).

This formula has three advantages over any other approach. First, because it only contains covariances, it is inherently independent of any kind of offset or bias from either Pan or MS bands. Second, it is fully consistent with the Gram-Schmidt framework. Third, it is computed solely from image statistics, with the MS covariance matrix usually already pre-computed and stored with the MS image.

### 3. OTHER CONSIDERATIONS

#### 3.1 Compute the MS to Pan weights per sensor or per scene?

There are still two major options left to choose from. The above formula (6) allows computing MS to Pan weights for each scene or image individually. Or, we can pick a representative scene for each sensor, compute weights for it, and use those weights for all other images coming from the same sensor. Both workflows have their advantages and disadvantages. For example, in scenes with only brown desert or blue ocean, in icy areas, or areas with full cloud cover, computing the weights again and again not only seems like a waste of CPU time, but also may lead to problems if the image has a constant band (e.g., all white). Working with representative scenes allows to hand-select them, choose relevant content (such as urban areas), and make sure they don't have any alignment problem. Note that the Gram-Schmidt method is more robust to spatial misalignment of the bands than most other pan-sharpening methods because all transform coefficients are computed in the low MS resolution. Good practice is to have good sets of weights pre-computed and use them as a starting point, and only re-compute MS to Pan weights as a refinement for certain scenes as needed.

#### 3.2 MS to Pan weights can be normalized to 1

The MS to Pan weights are relative weights. As we can see from the Gram-Schmidt transform equations (2) above, any scale factor applied to all weights cancels out. The resulting forward transformed MS bands are independent of such a scale factor. Finally, when we back transform them, as long as the substituted high resolution Pan band got gain adjusted to the simulated Pan band, such a scale factor cancels out there as well. Again, same as with the argument about the band means above, this is why the gain bias adjustment of the high resolution Pan band is important, before it can replace the simulated one.

All this gives us the liberty of normalizing the weights to 1, without changing the results.

#### 3.3 How about negative weights?

Another comment to be made is about dealing with negative weights. Although only rarely observed so far, it can happen that a weight can come out slightly negative. For instance, we got -0.01 for the IR weight for the UltraCam sensor which basically has no IR in the Pan band. Another case was -0.11 for the Blue weight for Ikonos, where the insufficient coverage of the Pan bands IR region by just one NIR band on the MS side caused some tilting of the weight vector.

At this point the less sensitive reader may not bother about negative weights and just use them. However, negative weights are not possible when they are read off the sensor's spectral sensitivity curves as suggested by Laben and Brower (Laben and Brower, 2000). So they are hard to justify. For this reason we set negative weights back to 0 and renormalize the weights. This has worked fine for us so far. However, for the sake of completeness, we will now outline a more rigorous treatment. Let's rewrite our MS to Pan weight formula as a minimization problem with additional inequality constraints using matrix vector notation (calling the MS-Pan covariance vector  $p$ ) as

$$\|C w - p\| = \min, \quad w_i \geq 0 \quad (7)$$

Making use of the symmetry of  $C$ , we get

$$w^T C^2 w - 2 p^T C w = \min, \quad w_i \geq 0 \quad (8)$$

With obvious substitutions this can be turned into

$$\frac{1}{2} w^T Q w + b^T w = \min, \quad w_i \geq 0 \quad (9)$$

which is a classical quadratic programming problem. On how to solve it, see for instance (Wikipedia, 2013).

#### 3.4 Pan-sharpen on the fly without computing statistics

Consider an emergency response scenario, for example, a wild fire or an earthquake. Satellite images need to be processed and analyzed as quickly as possible to find out where the most damage is and where the rescue personnel should be sent first. Besides RGB, other three band combinations involving an infrared or near infrared channel, and two of the visible bands can be used, too. Such pseudo color images can highlight other features, which are not so easy to see in the regular RGB image.

In an emergency or disaster situation, no one wants to wait until all the statistics including the covariance matrix have been computed, for all images involved.

Instead, we use the MS to Pan weights pre-computed for the current sensor. In the absence of image statistics, we pan-sharpen only the area needed to satisfy the current request (e.g., the screen), but guaranteeing a minimal pixel area. I.e., we enlarge the processed area as needed to avoid strange results that we might get with too few pixels. This way pan-sharpened imagery is available instantaneously, suitable for panning and zooming. On the downside this way of processing is not globally consistent: Exporting the entire image this way would lead to small tiling artefacts, as the Gram-Schmidt transformation matrix will somewhat vary from tile to tile.

#### 4. EXAMPLE

Let us look at one example. We have picked a WorldView-2 satellite scene of London and look at the London tower bridge, see Fig. 1-4 below. Here we show:

1. The original MS image around the tower bridge.
2. The original Pan image.
3. The MS image up-sampled to full Pan resolution.
4. The result of the Gram-Schmidt pan-sharpening.

Original imagery courtesy of Digital Globe. The images have been processed using Esri's ArcGIS for Desktop 10.2 and then cropped to fit in here. Compare the sharpness of the pan-sharpened result image to the sharpness of the original Pan image and the colors of the pan-sharpened result image to the colors of the up-sampled MS image.

#### 5. SUMMARY AND CONCLUSION

In this paper, we have shown the following for the Gram-Schmidt pan-sharpen method:

1. If the MS covariance matrix and the MS to Pan weights are known, the Gram-Schmidt transform coefficients can be directly computed. This turns the Gram-Schmidt pan-sharpen method from a global into a local method, meaning that any small region of the image can be pan-sharpened on the fly without having to compute anything globally over the entire image first.
2. If the MS covariance matrix and the MS to Pan covariance vector are known, optimal MS to Pan weights can be directly computed. Then the Gram-Schmidt transform coefficients can be computed. So the entire Gram-Schmidt transform is fully determined by the MS and MS to Pan covariances.
3. Once the MS to Pan weights have been computed on a representative scene of a given sensor, other images from the same sensor can be pan-sharpened even without image statistics, but then only approximately, by treating every requested tile or area as a full image. Such a local or dynamic mode can be useful for instantaneously pan-sharpening freshly incoming images without having to compute full image statistics first.

We hope that this description or recipe makes it easier for others to go with the Gram-Schmidt pan-sharpen method and make optimal use of its potential.

#### 6. REFERENCES

- Aiazzi, B., et al, 2007. MS + Pan image fusion by an enhanced Gram-Schmidt spectral sharpening. In *New Developments and Challenges in Remote Sensing*, Z. Bochenek (ed.), 2007 Millpress, Rotterdam, ISBN 978-90-5966-053-3, pp. 113-120.
- Amro, I., et al, 2011. A survey of classical methods and new trends in pansharpening of multispectral images. *EURASIP Journal of Advances in Signal Processing* 2011, 2011:79, Granada, Spain, <http://asp.eurasipjournals.com/content/2011/1/79/> (30 Sep 2011).
- Wikipedia, 2013, [http://en.wikipedia.org/wiki/Quadratic\\_programming](http://en.wikipedia.org/wiki/Quadratic_programming) (20 Mar 2013).
- Laben, C.A. and Brower, B.V., 2000. Process for Enhancing the Spatial Resolution of Multispectral Imagery using Pan-Sharpener. U.S. Patent 6,011,875. Jan 4, 2000.



**Figure 1:** WorldView-2 image of the London tower bridge. MS image at source resolution. Courtesy of Digital Globe.



**Figure 2:** Pan image at source resolution. Courtesy of Digital Globe.



**Figure 3:** MS image up-sampled to Pan resolution using bilinear resampling.



**Figure 4:** MS image pan-sharpened using Gram-Schmidt pan-sharpening in Esri's ArcGIS for Desktop 10.2.