

Probabilistic Model of Error in Fixed-Point Arithmetic Gaussian Pyramid

Antoine Méler

John A. Ruiz-Hernandez
INRIA Grenoble - Rhône-Alpes
655 avenue de l'Europe
38 334 Saint Ismier Cedex France

James L. Crowley

{Antoine.Meler, John-Alexander.Ruiz-Hernandez, James.Crowley}@inrialpes.fr

Abstract

The half-octave Gaussian pyramid is an important tool in computer vision and image processing. The existence of a fast algorithm with linear computational complexity makes it feasible to implement the half-octave Gaussian pyramid in embedded computing systems using only integer arithmetic. However, the use of repeated convolutions using integer coefficients imposes limits on the minimum number of bits that must be used for representing image data. Failure to respect this limits results in serious degradation of the signal to noise ratio of pyramid images.

In this paper we present a theoretical analysis of the accumulated error due to repeated integer coefficient convolutions with the binomial kernel. We show that the error can be seen as a random variable and we deduce a probabilistic model that describes it. Experimental and theoretical results demonstrate that the linear complexity algorithm using integer coefficients can be made suitable for video rate computation of a half-octave pyramid on embedded image acquisition devices.

1. Introduction

Over the last years, the pyramid and Gaussian derivative features have emerged as a powerful source of image features for computer vision and pattern recognition. Gaussian derivative features have been used for 3D reconstruction, image stitching, object recognition [9] [10] and detection of faces [8] and other categories of image objects. Gaussian derivative features are the basis for SIFT descriptor [7], the histogram of gradients [4] and the second local order structure image solid [6]. Such features are also used in biological vision models [5].

An inconvenience of Gaussian derivative features is that most algorithms require relatively expensive floating-point computations with algorithmic complexities of N^2 or worse. It is possible to compute such features with a linear

complexity algorithm using only integer coefficient convolutions. However, the use of this algorithm requires careful attention to the signal noise introduced by roundoff error during repeated convolutions with integer coefficient filters.

Crowley and Stern [3] have proposed a linear complexity algorithm for computing an exactly scale invariant Gaussian pyramid using a technique referred to as Cascade convolution with resampling. This algorithm has an $O(N)$ complexity where N is the number of pixels in the image and produces $2\log_n(N)$ resampled images in which each image has an identical impulse response at a logarithmically increasing set of scales.

An integer coefficient version of this algorithm, the half-octave binomial pyramid was proposed by Crowley and Riff [2] in 2003. A pyramid computed with this algorithm can provide Gaussian derivative image features using only simple sums and differences of pixels. As a result, this algorithm is suitable for calculation of Gaussian derivative features on embedded computing devices. However restrictions in the available memory on such devices can lead designers to implement the algorithm with as few as 8 bits per pixel. When used with repeated convolution, this a restricted bit length introduces serious degradation in the signal to noise ratio of the resulting pyramid. Such errors can cause important loss of reliability and precision of image descriptors based on the pyramid.

In this paper we present a probabilistic model for the error resulting from the computation of the half-octave Gaussian pyramid fixed point integer coefficients. This model describes the error generation and propagation at each pyramid level due to recursive convolutions with a binomial kernel. Our results show that the half-octave pyramid can be accurately computed with short fixed-point numbers.

This paper is organized as follows. Section 2 explains the half-octave Gaussian pyramid algorithm. Our probabilistic error model is developed in section 3, followed experimental verification in section 4. Section 5 concludes with discussion of the results and its application for embedded image processing.

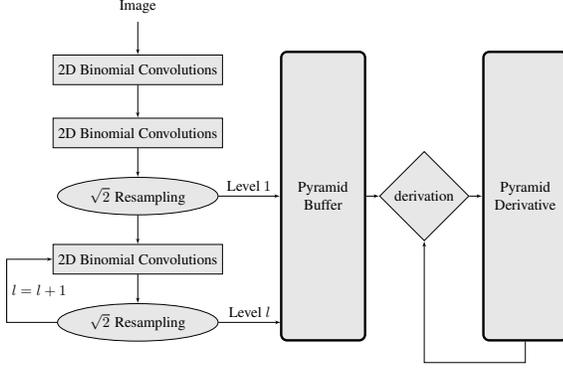


Figure 1. The $O(N)$ cascade convolution pyramid computation scheme

2. The Half-Octave Gaussian Pyramid

In this section we describe the calculation of the half-octave Gaussian pyramid and its derivatives.

A linear complexity pyramid algorithm for the calculation of Gaussian derivatives has been known since the 1980's [1] (figure 1). This algorithm involves alternatively convolving with a Gaussian support, and resampling the resulting image. The result of this algorithm is a half-octave Gaussian pyramid. An integer coefficient version of this algorithm [2] has been demonstrated using repeated convolutions of the binomial kernels $[1, 2, 1]$ and $[1, 2, 1]^t$. Implementations that compute such pyramids on PAL sized images and bigger exist for the current generation of computer stations, and can be embedded in dedicated signal processor.

The half-octave Gaussian pyramid for an $N * N$ pixels image is composed of up to $L = 2 \log_2(N)$ images. The effect of cascade convolution is to sum the variances of the filters, so that the cumulative variance for the image $l \in \llbracket 1, L \rrbracket$ is $\sigma_l = 2^l$. That for, the convolution step involves 2 convolutions by $[1, 2, 1]$ in each direction. Interleaving resampling with convolutions decreases the number of image samples while expanding the distance between samples. This has the effect of dilating the Gaussian support without increasing the number of samples used for the Gaussian, effectively increasing the scale. The resampling rate is set to $\sqrt{2}$, resulting in a constant ratio of scale over sample distance.

The resulting pyramid represents the original $N * N$ pixels image with a sequence of $2 \log_2(N)$ images at a geometric progression of scales each with twice as less samples as the previous, resulting in a total of $2N^2$ samples (figure 2). The 2-dimensional derivative of order (d_X, d_Y) is then obtained by convolving d_X times the resulting pyramid with the derivative kernel $[-1, 0, 1]$, and d_Y times with $[-1, 0, 1]^t$.

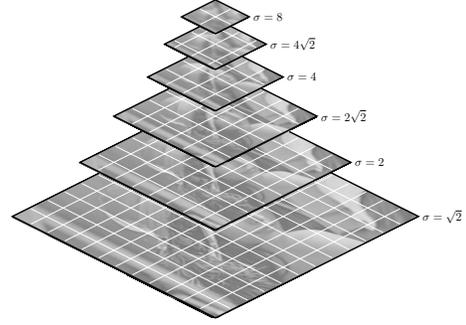


Figure 2. The half-octave pyramid algorithm result

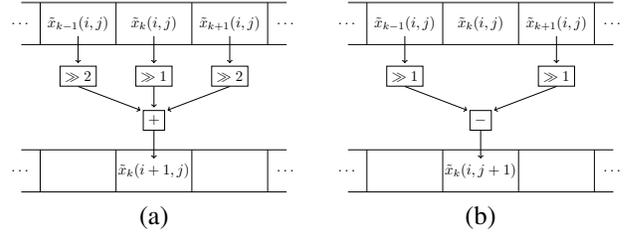


Figure 3. Convolution with the two base kernels (smoothing (a) and differentiation (b)). The $\gg n$ operator stands for right n bits shift with zero padding

3. Probabilistic Formulation

3.1. Random Error Model

In this section, we present our model to analyze the noise effect of using fixed point coefficients for the pixel values in the pyramid.

Let $\tilde{x}_{\vec{k}}(\vec{i}, \vec{j})$ be the value of the pixel of coordinate \vec{k} after \vec{i} smoothing convolutions and \vec{j} differentiations in directions X and Y. $\tilde{x}_{\vec{k}}(\vec{i}, \vec{j})$ is the sum of an exact theoretical value $x_{\vec{k}}(\vec{i}, \vec{j})$ and an error $E_{\vec{k}}(\vec{i}, \vec{j})$.

$$\tilde{x}_{\vec{k}}(\vec{i}, \vec{j}) = x_{\vec{k}}(\vec{i}, \vec{j}) + E_{\vec{k}}(\vec{i}, \vec{j}) \quad (1)$$

We model the error E by a random variable.

In order to simplify the problem, the error part of contiguous pixels is supposed independent. This approximation is verified to be very accurate in section 4. Thus, we will simply consider $E(i, j)$ where i and j are the number of smoothing and differentiation convolutions respectively applied to one pixel, including both directions X and Y. We consider only two operations: convolution with the normalized kernels $K_s = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$ and with $K_d = [\frac{1}{2}, 0, -\frac{1}{2}]$ as schematized on figure 3. We also suppose that all the convolutions with K_s are applied before those with K_d . Thus, we express $E(i, j = 0)$ as a function of $E(i - 1, j)$, then $E(i, j > 0)$ as a function of $E(i, j - 1)$.

3.2. Error Generation and Propagation

\tilde{x} is a fixed point number with n_i bits of integer part and n_f bits of fractional part. Furthermore, \tilde{x} is supposed to be initially an integer (i.e. its n_f least significant bits are 0).

Because pyramid computation is recursive, the error of one value is composed of two parts. One is a new error, generated by the current computation E^{new} , and the other, E^{prop} , is accumulated during the previous computations and propagated to the new value.

In this section, we determine, for a given number of fractional bits n_f , the expressions of the random variables $E_{n_f}^{new}$ and $E_{n_f}^{prop}$.

Error generation $E_{n_f}^{new}$ is the error generated by the calculation of a new pyramid value.

As shown in figure 3, a convolution with K_s or K_d starts with divisions by a number $2^{n_{shift}}$, which, in fixed point arithmetic, corresponds to a bit shift with the loss of the n_{shift} least significant bits. Such a division can be modeled with $U_{V_{n_f}(n_{shift})}$, the uniform discrete random variable with the set of possible values $V_{n_f}(n_{shift}) = \left\{ \frac{-i}{2^{n_f+n_{shift}}} \right\}_{i \in [0, 2^{n_{shift}}-1]}$. In the case of values being initially integers, this model is well-suited for $2i + j > n_f$. Before this condition is true, the divisions do not lead to a loss of information. Thus, the generated error is equal to:

- If $2i + j \leq n_f$:

$$E_{n_f}^{new}(i, j) = 0 \quad (2)$$

- Otherwise:

$$\begin{cases} E_{n_f}^{new}(i, j = 0) = U_{V_{n_f}(2)} + U_{V_{n_f}(1)} + U_{V_{n_f}(2)} \\ E_{n_f}^{new}(i, j > 0) = U_{V_{n_f}(1)} - U_{V_{n_f}(1)} \end{cases} \quad (3)$$

Error propagation This paragraph considers the error that propagates from a state (i, j) to the next $(i + 1, j)$ or $(i, j + 1)$.

We assume the number of integer bits, n_i , is sufficient to avoid overflow. In this case, the error $E_{n_f}^{prop}(i, j)$ propagated during a convolution with K_s or K_d is a random variable defined by recurrence:

$$\begin{cases} E_{n_f}^{prop}(0, 0) = 0 \\ E_{n_f}^{prop}(i, j = 0) = \frac{E_{n_f}(i-1, j)}{2} + \frac{E_{n_f}(i-1, j)}{2} \\ \quad + \frac{E_{n_f}(i-1, j)}{4} \\ E_{n_f}^{prop}(i, j > 0) = \frac{E_{n_f}(i, j-1)}{2} - \frac{E_{n_f}(i, j-1)}{2} \end{cases} \quad (4)$$

3.3. Total Accumulated Error

The total error is the sum of the propagated part and the generated part:

$$E_{n_f}(i, j) = E_{n_f}^{prop}(i, j) + E_{n_f}^{new}(i, j) \quad (5)$$

We deduce from the above model the error characteristics (standard deviation and variance) as a function of the number of smoothing convolutions (i) and differentiations (j). We first express the error with recurrence formulation, and then give the solution.

Error evolution The values are initially integers, $E_{n_f}(i, j) = 0$ for $2i + j \leq n_f$. Thus we express the expected value $\mathbb{E}[E_{n_f}(i, j)]$ and the variance $Var[E_{n_f}(i, j)]$ of the complete error random variable for $2i + j \leq n_f$. In this condition, $E_{n_f}^{prop}(i, j)$ and $E_{n_f}^{new}(i, j)$ can be supposed independent.

Smoothing convolutions are computed before computing image derivatives from image differences. Thus we first give the recurrence law for smoothing ($j = 0$) and then for derivative computation ($j > 0$).

- Smoothing:

$$\begin{aligned} \mathbb{E}[E_{n_f}(i, j = 0)] &= \mathbb{E}[E_{n_f}^{prop}(i, j)] + \mathbb{E}[E_{n_f}^{new}(i, j)] \\ &= \mathbb{E}[E_{n_f}(i-1, j)] - \frac{1}{2^{n_f}} \end{aligned} \quad (6)$$

$$\begin{aligned} Var[E_{n_f}(i, j = 0)] &= Var[E_{n_f}^{prop}(i, j)] + Var[E_{n_f}^{new}(i, j)] \\ &= \frac{3}{8} Var[E_{n_f}(i-1, j)] + \frac{5/3}{2^{2(n_f+5)}} \end{aligned} \quad (7)$$

- Derivation:

$$\begin{aligned} \mathbb{E}[E_{n_f}(i, j > 0)] &= \mathbb{E}[E_{n_f}^{prop}(i, j)] + \mathbb{E}[E_{n_f}^{new}(i, j)] \\ &= 0 \end{aligned} \quad (8)$$

$$\begin{aligned} Var[E_{n_f}(i, j > 0)] &= Var[E_{n_f}^{prop}(i, j)] + Var[E_{n_f}^{new}(i, j)] \\ &= \frac{1}{2} Var[E_{n_f}(i, j-1)] + \frac{2/3}{2^{2(n_f+2)}} \end{aligned} \quad (9)$$

Error expression We can deduce the absolute expressions of $\mathbb{E}[E_{n_f}(i, j)]$ and $Var[E_{n_f}(i, j)]$ as a function of i, j and n_f .

$$\mathbb{E}[E_{n_f}(i, j)] = \begin{cases} \frac{-1}{2^{n_f}} \left(i - \frac{n_f}{2} \right) & \text{if } j = 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$Var[E_{n_f}(i, j)] = \frac{1}{3} \left(\frac{1 - \left(\frac{3}{8}\right)^{i - \frac{n_f}{2}}}{2^{2n_f + j + 2}} + \frac{1 - \left(\frac{1}{2}\right)^j}{2^{2n_f + 2}} \right) \quad (11)$$

Note that the error variance increases with i and j toward a limit:

$$Var[E_{n_f}(\infty, 0)] = Var[E_{n_f}(i, \infty)] = \frac{1/3}{2^{2n_f + 2}}$$

4. Experiments and validation

In this section we compare the error predicted by our model with the one measured by subtracting an image pyramid with varying precision ($0 \leq n_f \leq 12$) and a 64 bits one. We perform this analyse on synthetic images (Perlin noise) and photos randomly downloaded from the internet.

4.1. Error Distribution

To validate our model, we compare predicted and measured error. There is no simple mathematic expression for the distribution of the error as modeled in this paper. However we can estimate it as accurately as needed by running simulations of our random variable.

As shown figure 4, our model accurately matches the error distribution measured in pyramids built from highly-informative images like Perlin noise. In the case of photographs, an unpredicted null-error peak can be observed. The figure 5 demonstrates that in homogeneous areas (sky, unfocused background, over/under-exposition, ...), smoothing convolutions do not alter pixel values which thus stay integers. In such areas, our model for the loss of information as an addition by a random uniform variable fails while the pyramid level is under the singular zone characteristic scale. Thus, our theoretical analysis provides information about the accumulated error in the worst case.

One can also observe a larger spread in the measured distributions than in the theoretical one. This difference is due the independence hypothesis which underestimate the variance by neglecting a positive covariance term. However, it appears that this term play a minor part.

4.2. Error Parameters

In this section, we graphically display the parameters of the random error variable.

Due to the presence of homogeneous areas in photos, experimental standard deviation and variance are very data-dependant. It is thus more interesting to visualize the theoretical values which correspond to the worst case (figure 6). It can be seen that the standard deviation convergence in the levels is quite fast. We thus can simply study the limit value (figure 7).

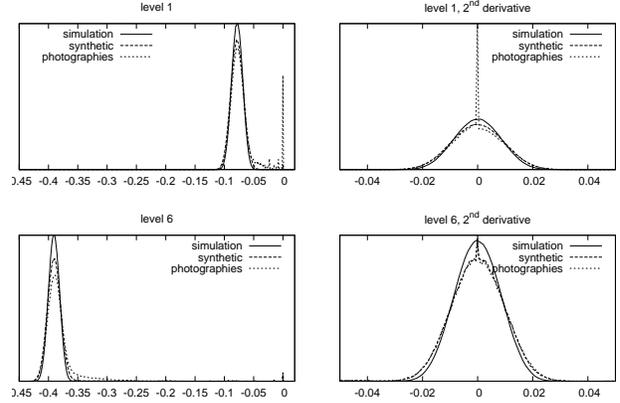


Figure 4. Distribution of the error of the pyramid values with $n_f = 6$. abscissa: error value, ordinate: pixel rate with this error.

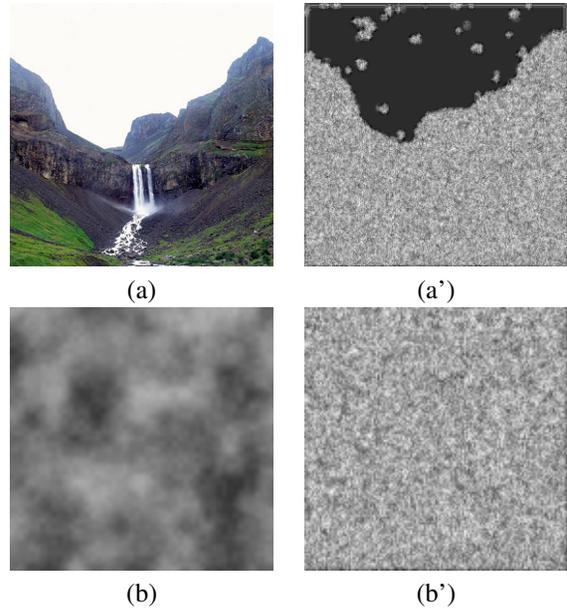
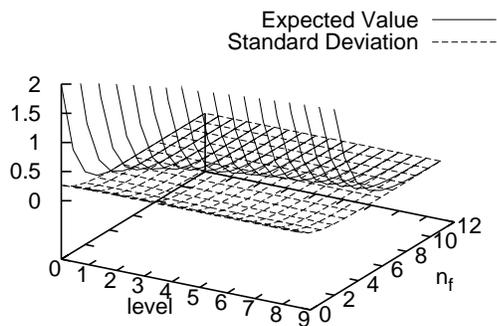


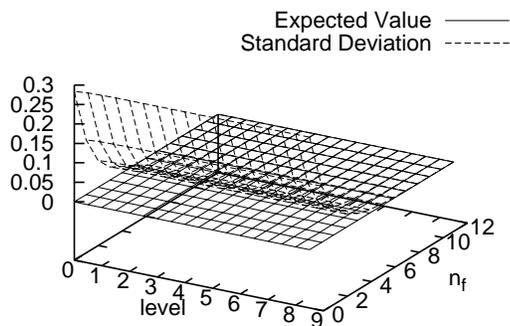
Figure 5. Over-exposed photography (a) and associated error image at pyramid level 3 with $n_f = 6$ (a'). The saturated area corresponds to a null-error area which is very poorly modeled by our random error variable. Perlin noise (b) and its associated error image (b').

5. Conclusion

This paper has presented a model for the error due to recursive truncations by a random noise in computing a Gaussian pyramid using a linear complexity algorithm involving cascaded convolution with integer coefficient binomial filters. This model predicts the additive image error as a function of the number of fractional bits used, the scale of the pyramid and the order of the derivative. We have validated our model by comparing a simulation of our random variable and measurements performed in pyramids applied



(a)



(b)

Figure 6. Expected value and standard deviation of the error variable as a function of the level and the number of fractional bits n_f before differentiations (a) and after 2 differentiations (b).

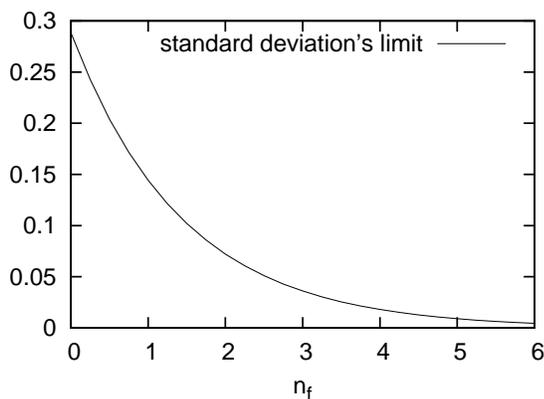


Figure 7. Limit value of the error standard deviation as a function of the number of fractional bits n_f .

to synthetic images and photos. We have shown that the noise variance rapidly converges to a fixed value as a func-

tion of the pyramid scale. However, the divergence is constant and grows rapidly for small numbers of fractional bits in the fixed point representation. This divergence becomes zero when the signal is differentiated, thus making it irrelevant for many applications. We can conclude by observing that binomial convolution kernels tends to smooth the error noise, thus compensating for random noise added by integer truncation in cascade convolution. This makes the Gaussian binomial pyramid a good multi-scale image representation solution for use in embedded computer vision systems.

References

- [1] J. Crowley and A. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE PAMI*, 6(2):156–169, March 1984. 2
- [2] J. Crowley and O. Riff. Fast computation of scale normalised gaussian receptive fields. pages 584–598, 2003. 1, 2
- [3] J. Crowley and R. Stern. Fast computation of the difference of low-pass transform. *PAMI*, 6(2):212–222, March 1984. 1
- [4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE CVPR*, pages 886–893, June 2005. 1
- [5] M. A. Georgeson, K. A. May, T. C. Freeman, and G. S. Hesse. From filters to features: scale-space analysis of edge and blur coding in human vision. *Journal of vision*, 7(13), 2007. 1
- [6] L. Griffin. The second order local-image-structure solid. *IEEE PAMI*, 29(8):1355–1366, 2007. 1
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1
- [8] J. A. Ruiz Hernandez, A. Lux, and J. L. Crowley. Face detection by cascade of gaussian derivatives classifiers calculated with a half-octave pyramid. *IEEE Conference on Automatic Face and Gesture Recognition*, Amsterdam, Sep 2008. 1
- [9] B. Schiele and J. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36:31–50, 2000. 1
- [10] J. Yokono and T. Poggio. Oriented filters for object recognition: an empirical study. In *Proceedings of the IEEE FG2004. Seoul, Korea*, page 755, 2004. 1