Image processing with JPEG2000 coders

Przemysław Śliwiński[a] and Czesław Smutnicki[a] and Artur Chorażyczewski[a]

[a]The Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, Wrocław, Poland

ABSTRACT

In the note, several wavelet-based image processing algorithms are presented. Denoising algorithm is derived from the Donoho's thresholding. Rescaling algorithm reuses sub-division scheme of the Sweldens' lifting and a sensor linearization procedure exploiting system identification algorithms developed for nonlinear dynamic systems. Proposed autofocus algorithm is a passive one, works in wavelet domain and relies on properties of lens transfer function. The common advantage of the algorithms is that they can easily be implemented within the JPEG2000 image compression standard encoder, offering simplification of the final circuitry (or the software package) and the reduction of the power consumption (program size, respectively) when compared to solutions based on separate components.

Keywords: Wavelets, image processing, JPEG2000, denoising, auto-focusing, scaling, high dynamic range imaging, focus stacking, demosaicking

1. INTRODUCTION

The paper presents several wavelet-based image processing algorithms for:

- denoising,
- rescaling and demosaicking,
- sensor linearization, and
- auto-focusing.

There are also proposed exemplary applications of the algorithms:

- focus tracking and stacking, and
- high dynamic range imaging

Implementations of the algorithms will be depicted during the poster session at the conference.

Denoising algorithm is derived from the Donoho's thresholding idea (*cf.* [1, 2]) and applied to reduce the amount of a noise produced by an image sensor. Rescaling algorithm reuses a sub-division scheme of the Sweldens' lifting algorithm (see *e.g.* [3,4], which is an *in-situ* version of the fast wavelet transform and thus commonly used in JPEG2000 encoder implementations; [5,6]), and can be used in *e.g.* live-view applications. Sensor linearization procedure exploits Greblicki's system identification algorithms developed for nonlinear dynamic systems; *cf.* [7,8]. The autofocus algorithm is a passive one, works in wavelet domain and exploits the properties of lens transfer function; *cf.* [9, 10, 11, 12]. The convergence of autofocusing is shown under noisy environment assumptions for a wide class of input images.

The common advantage of the algorithms is that they can easily be implemented within the existing JPEG2000 image compression standard encoders and thus can offer a significant simplification of the final circuitry and reduction of the power consumption (or a smaller footprint of the respective software implementation) when compared to solutions based on separate components.

1.1 A camera model and sensor linearization

The pivotal for further findings is the observation that the digital camera can be modelled as a Wiener system (*cf. e.g.* [13,14]), *viz.* the nonlinear dynamic system composed of a linear dynamic block (corresponding to the lens and the anti-aliasing filter; see *e.g.* [15]) followed by a static nonlinearity (representing the image sensor; [16]).

ALGORITHM: Due to the observation in [7] the inverse of the sensor nonlinearity is the *regression function* of the input (a real scene) on the output (a captured image) and thus can be estimated using a regression estimate (*e.g.* the nonparametric one, as proposed in the cited papers [7] and in [8] or [17]).

1.2 JPEG2000 encoder

JPEG2000 is the newest ISO standard for images. It has been developed with the help of the recent achievements in harmonic analysis, image and signal processing and information theory (like wavelets theory and fast transform algorithms, arithmetic coding and multidimensional embedded data coding); see *e.g.* [18, 19, 20] and the source books [21, 22, 23]. The JPEG2000 image compression standard has already found its application in digital cinema and in medicine. Its adoption in consumer level devices like digital cameras, personal computers and in Internet has been hindered by the standard's complexity: both in terms of the theory supporting the compression algorithm and of the resulting implementation (in either hardware or software manner; see *e.g.* [24, 25]).

While this is the openness of the standard which can partially be blamed for its complexity, it simultaneously is one of its biggest advantage as it allows to adjust the standard to the particular applications requirements; *cf.* [26, 27]. From this paper vantage point we would like to point out:

- a possibility of selection of (predefined or user-defined) wavelet transform,
- a availability of embedding of the pointwise nonlinear transform, and
- a flexible quantization scheme, see *e.g.* [28], as the standard's means to be exploited in our algorithms.

Remark 1. *One can treat a JPEG2000 encoder as a (powerful) co-processor, which effectively performs (forward and inverse) wavelet transform for a given wavelet family. In particular, the encoder compatible with the core standard, implements two bi-orthogonal transforms, based on 5/3 LeGall and 9/7 Daubechies-Cohen-Feauveau families; [5,27]. The extended standard version (Part II: ISO/IEC 15444-2) allows implementation of arbitrary, bi-orthogonal or orthogonal wavelet transform. In the note, Haar wavelets are used in the proposed autofocus algorithm.*

Remark 2. *It should also be emphasized that a wavelet transform can be computed losslesly when a proper wavelet family is applied (*i.e.* when the transform coefficients have finite binary representations). The prominent examples are Haar and LeGall wavelets. There is a recipe for obtaining a custom designed wavelets in the paper [3].*

Remark 3. *The scalar quantization scheme (a TCQ (trellis coded quantization) based one is a part of the standard extension) can either be uniform or uniform with a deadzone. In the latter, one can manipulate the range within from the coefficients are zeroed; [5].*

2. ALGORITHMS

2.1 Denoising

We assume that there is a single source of the noise situated in a image sensor and is of random nature. Thus we neglect quantization noise generated during an A/D conversion (so called *banding* and *fixed pattern noise*). A sensor sensitivity is obtained by a amplification increasing of its output. Thus, since the sensor noise is of 'thermal nature' and – to much extent – of a constant amplitude, then signal-to-noise ration diminishes and the noise is especially apparent at high ISO levels.

We exploit the following observations:

- the first is from Donoho's paper [1] where it is shown that for orthogonal transforms (or linear, in general) the white Gaussian noise is transformed from the image representation to all wavelet coefficients of the image. A hard-threshold operator (zeroing the coefficients smaller than a given – noise variance-dependent – level) is supposed to remove the noise from these coefficients.

- There is usually an anti-aliasing (lowpass) filter placed before image sensor (in order to avoid *moire* – the visual artifacts appearing when Nyquist condition is not satisfied). Thus the part of the noise (corresponding to the frequencies blocked by AA filter) can be removed.
- Wavelet transform coefficients (due to their property of possessing of (at least) one vanishing moment) are invariant to the noise expected value (*i.e.* the noise do not need to have zero-mean).

The denoising algorithm (in its most crude form) is the following:

ALGORITHM: Compute the wavelet transform of the image (for each of its components separately) and perform hard thresholding on the finest wavelet coefficients only be setting the width, $0 \leq \xi < 1$, of the quantization deadzone, yielding a following quanitzer-dequantizer pair (*cf.* [5, p. 107])

$$q = Q(x) = \begin{cases} sign(x) \left\lfloor \frac{|x|}{\Delta} + \xi \right\rfloor & \frac{|x|}{\Delta} + \xi > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{and } \hat{x}_q = \begin{cases} sign(q) \lfloor |q| - \xi + \delta \rfloor & q = 0 \\ 0 & q \neq 0 \end{cases}$$

which can easily be implemented in C++

```
template <typename input_t, typename output_t>
class dead_zone_scalar_quantizer: public scalar_quantization_base<input_t, output_t>
{
/* ... */
 output_t operator ()(input_t const x)
 {
  return std::abs(x)/Delta + xi >= 0
         ? static_cast<output_t>(sgn(x)) * std::floor(std::abs(x)/Delta + xi)
         : output_t(0);
 }
};
template <typename input_t, typename output_t>
class dead_zone_scalar_dequantizer: public scalar_quantization_base<input_t, output_t>
{
/* ... */
 output_t operator ()(input_t const q)
 {
  return static_cast<output_t>(sgn(q)) * (std::abs(q) - xi + delta) * Delta;
 }
};
```

REMARK 4. *The algorithm does not take into account perceptual issues. There is, for instance, well know fact that the typical for sensors with CPA (color pattern array – usually a Bayer one; see [29, 30]) color noise is 'less natural' since, in a human eye, color detectors are turned of under low light conditions; see* e.g. *[31]. It can however be easily extended to this case after color conversion from RGB to YC_BC_R color space. For other approaches see* e.g. *[32, 33].*

2.2 Scaling

Scaling raster images (bitmaps) is usually performed with the help of interpolations. The simplest one is based on nearest-neighbor principle, however, it produces blocking artifacts (during enlarging) and *moire* (during shrinking the image). These effects are diminished when higher order interpolators are employed (like *e.g.* linear or cubic splines; see *e.g.* [34] for a tutorial and also [35, 36, 37]).

The following observations constitute the JPEG2000-based scaling algorithm:

- image enlarging can be performed by the inverse wavelet transform.

- the image energy is preserved if the transform is orthonormal or is bi-orthogonal with properly set amplification constants; see [38].

ALGORITHM: Take take the original image and interleave its points with black ones and perform a lifting steps corresponding to the employed wavelet family as if it is a normal inverse transform procedure; *cf.* [39]. For a L^2-normalized transforms, the pivotal code snippet can look like

```
template<typename T>
void wavelet_transform<T>::L2_Haar_upscaling(matrix<T> &image, int M)
{
 int image_size = image.height() >> (M - 1);
 for(int m = M - 1; m >= 0; --m)
 {
  int step = 1 << m;
  for(int i = 0; i < image_size; ++i)
   for(int j = 0; j < image_size; j += 2)
   {
    T &s = image[i * step][j * step];
    s *= sqrt_2;
   }
  image_size <<= 1;
 }
}
template<typename T>
void wavelet_transform<T>::L2_Haar_downscaling(matrix<T> &image, int M)
{
 int image_size = image.height();
 for(int m = 0; m < M; ++m)
 {
  int step = 1 << m;
  for(int i = 0; i < image_size; ++i)
   for(int j = 0; j < image_size; j += 2)
   {
    T &s = image[i * step][j * step]; T &d = image[i * step][(j + 1) * step];
    d -= s;      d /= sqrt_2;
    s *= sqrt_2; s += d;
   }
  image_size >>= 1;
 }
}
```

REMARK 5. *Clearly, the direct application of the algorithm allows scaling the image by the factors equal to power of two only. It seems however to be sufficient in live-view applications for quick preview of the images.*

REMARK 6. *Note that the image, when downscaled (which, effectively, is tantamount to performing forward transform), is also averaged, [39], and therefore there is no risk of adding a* moire*.*

2.3 Demosaicking

After the signal denoising the next step of the processing pipe is usually a conversion of the sensor signal to a given color space. Since the Bayer color filter (*i.e.* the RGBG one) is usually used then the effective number of sensor pixels is four times smaller than the 'advertized' one. Since, essentially demosaicking is an interpolation operation, see *e.g.* [40, 41, 32] our algorithms is a simple application of the above scaling algorithm based on Sweldens lifting scheme:

ALGORITHM: Assign the green color to the component with the resolution twice as much as for other red and blue colors (by manipulation of the JPEG2000's components sub-sampling factors – see [5, p. 420]) and perform on these two latter colors one more step of the inverse wavelet transform. Then compute a standard's inverse color transform.

REMARK 7. *Since interpolation algorithms are prone to noise (see* e.g. *[42]), then both scaling and demosaicking routines should always be performed after denoising algorithm. The latter (at least in the form derived from thresholding-based approach) applies local averaging and thus smoothens the image.*

REMARK 8. *The algorithm – if the color and wavelet transform are reversible – is also reversible as a whole. Thus the original image data from sensor can always be restored. It can be a factor in biomedical applications, [43].*

2.4 Autofocus

In imaging one can distinguish two kind of focusing algorithms: the active and the passive one. The former uses additional circuits (*e.g.* lasers or ultrasounds). The latter analyses the scene. One can further break down the passive algorithms into two types: the first employs additional optical elements (microprism, for instance) while the second relies solely on the image acquired by the sensor; [9, 10, 11]. We are considering the last one – derived from [12]. It is based on the following observations:

- Approximately, the image in focus has the smallest smoothness, *e.g.* it has the sharpest edges or steepest slopes (since for such images one can assume that lens transfer function is a Dirac delta; see *e.g.* [15]).
- Wavelet functions are differential operators, [44, 45]. That is, their coefficients grows with decreasing smoothness of the underlying image, [12]). In particular. large the coefficients in columns detects horizontal edges, those on rows vertical ones. The rest is sensitive to diagonal edges.
- Noise (if additive and stationary) is not a factor since it adds evenly to all coefficients regardless the original image smoothness.

ALGORITHM: Set the region to be focused and compute its wavelet transform. The proper focus distance is equivalent to finding the biggest wavelet coefficients. To find this maximum one can employ a modified version of Newton-Raphson algorithm. Here we provide an implementation for Canon EOS digital cameras:

```
void canon_eos::autofocus(EdsCameraRef camera, CImage &image, direction_t direction = INF)
{
   double Q, old_Q;
   double epsilon = 0.01;
   shift_t shift = MAX;
   matrix af(af_size, af_size);
   old_Q = compute_Q(camera, image, af);
   while(1)
   {
    shift_focus(camera, direction, shift);
    Q = compute_Q(camera, image, af, K0);
    if(std::fabs(Q - old_Q) <= epsilon)
     // STOP IF the shift is the smallest...
     if(shift == MIN) break;
     else
       direction = redirection(direction);
   // IF Q and old_Q are equal (a rare case, but...)
   if(Q == old_Q)
       if(shift == MIN) break;
       else
        shift = decrease_step(shift);
   // IF you went too far...
   if(Q - old_Q < 0)
```

```
        // STOP IF the shift is the smallest, it’s the peak (or at least its neighborhood)...
        if(shift == MIN)
        {
          // get back to the previous (i.e. the best) Q - this is the focus point!
          direction = redirection(direction);
          shift_focus(camera, direction, shift);
          break;
        }
        // decrease shift
        else
        {
          shift = decrease_step(shift);
          direction = redirection(direction);
        }
    old_Q = Q;
  }
}
```

Remark 9. *One can use various functions (amongst all unimodal with respect to the focus distance) to find the biggest wavelet coefficients, for instance, one* $\max\{\cdot\}$ *function or a sum of absolute values instead a sum of squared coefficients – in the application* L^2*-normalized 2D Haar transform is used and* Q *is simply a* L^2*-norm of the wavelet coefficients.*

Remark 10. *For smooth images one should use the wavelets with smallest possible vanishing numbers,* e.g. *the Haar family – they vanishes only when the focused fragment has constant colors).*

Remark 11. *The information of the point (area) of focus can be stored as a standard’s ROI (region of interest).*

3. APPLICATIONS

3.1 Focus stacking (bracketing)

In several photo imaging application, especially in microscopy and macrophotography, there is a problem of the lack of the sufficient depth-of-field to capture the whole non-blurred object. The usual countermeasure is decreasing the lens aperture. However, the drawback of this approach consists in the presence of diffraction at small apertures which results in the loss of the image details. In case when, when the object is still or slowly moving one can apply focus stacking algorithm:

ALGORITHM: Take **L/DOF** number of pictures, where **L** is the object length/depth and **DOF** is a depth-of-field for a select aperture. Then compute one wavelet transform step on each of the image and compose the resulting one from the biggest wavelet coefficients and associated with them scaling function coefficients.

Remark 12. *Typically, the lens’ focal length varies with the change of the focus distance. When the lens is of zoom type one can compensate it changing the focal length accordingly. In case of prime lenses one need to rescale images then.*

3.2 Focus tracking

Since image sensors are flat the distance information is lost after capturing. Thus, in contrast to the ’phase detecting’ methods (based on prisms) one need to emulate the phase detection by changing the lens focal distance in order to detect the direction of the object movement. If the object is comparatively slow one can use this technique for tracking moving object by employing the proposed autofocus algorithm.

3.3 High Dynamic Range imaging

Present sensors have a dynamic range smaller than human. One of the technique proposed to compensate it, is HDR (high dynamic range imaging) and has simple two steps:

ALGORITHM: Take $\mathbf{EV}_H/\mathbf{EV}_S$ images with non-overlapping exposure setting and then merge them and "compress" the resulting image to the required number of bits. (Optionally, map the image by a gamma-function.)

REMARK 13. *JPEG2000 supports this operation thanks to the presence of embedded nonlinear function and due to the capability of representation of each component by 32bits per pixel.*

4. SUMMARY

In the note several wavelet-based algorithms have been described. They are in an early stage of development and, clearly, they leave an ample space for further improvements. The algorithms are currently tested as a software package, however, they will be transferred to FPGA-based circuits.

REFERENCES

1. D. L. Donoho, "De-noising by soft thresholding," *IEEE Transactions on Information Theory* **41**(3), pp. 613–627, 1995.
2. D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Transactions on Infromation Theory* **44**(6), pp. 2435–2476, 1998.
3. A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis* **5**, pp. 332–369, 1998.
4. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *The Journal of Fourier Analysis and Applications* **4**(3), pp. 245–267, 1998.
5. D. Taubman and M. Marcellin, *JPEG2000. Image Compression Fundamentals, Standards and Practice*, vol. 642 of *The Kluwer International Series in Engineering and Computer Science*, Kluwer Academic Publishers, 2002.
6. T. Acharya and P.-S. Tsai, *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*, Wiley-Interscience, 2005.
7. W. Greblicki, "Nonparametric identification of Wiener systems," *IEEE Transactions on Information Theory* **38**(5), pp. 1487–1493, 1992.
8. W. Greblicki, "Nonparametric identification of Wiener systems by orthogonal series," *IEEE Transactions on Automatic Control* **39**(10), pp. 2077–2086, 1994.
9. A. Erteza, "Depth of convergence of a sharpness index autofocus system," *Applied Optics* **16**(8), pp. 2273–2278, 1977.
10. F. C. A. Groen, I. T. Young, and G. Ligthart, "A comparison of different focus functions for use in autofocus algorithms," *Cytometry* **6**(2), pp. 81–91, 1985.
11. E. Krotkov, "Focusing," *International Journal of Computer Vision* **1**(3), pp. 223–237, 1987.
12. J. Widjaja and S. Jutamulia, "Wavelet transform-based autofocus camera systems," *IEEE Proceedings* , 1998.
13. S. A. Billings and S. Y. Fakhouri, "Theory of separable processes with application to the identification of non-linear systems," *Proceedings of IEE* **125**, pp. 1051–1058, 1978.
14. S. A. Billings, "Identification of non-linear systems—a survey," *Proceedings of IEE* **127**(6), pp. 272–285, 1980.
15. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, Wiley-Interscience, 1978.
16. Y.-C. Chuang, S.-F. Chen, S.-Y. Huang, and Y.-C. King, "Low-cost logarithmic cmos image sensing by nonlinear analog-to-digital conversion," *IEEE Transactions on Consumer Electronics* **51**(4), pp. 1212–1217, 2005.
17. M. Pawlak, Z. Hasiewicz, and P. Wachel, "On nonparametric identification of Wiener systems," *IEEE Transactions on Signal Processing* **55**(5), pp. 482–492, 2007.

18. “Special issue on JPEG2000 standard,” *IEEE Signal Processing Magazine* **18**(5), 2001.
19. “Special issue on JPEG 2000 standard,” *Signal Processing: Image Communication* **17**, 2002.
20. “Special section on JPEG 2000 digital imaging,” *IEEE Transactions on Consumer Electronics* **49**, pp. 771–888, 2003.
21. I. Daubechies, *Ten Lectures on Wavelets*, SIAM Edition, Philadelphia, 1992.
22. S. G. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, 1998.
23. A. Cohen, *Numerical analysis of wavelets methods*, Studies in Mathematics and Its Applications, Elsevier, Amsterdam, 2003.
24. D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi, “Embedded block coding in JPEG 2000,” *Signal Processing. Image Communication* **17**, pp. 49–72, 2000.
25. G. Savaton, E. Casseau, and E. Martin, “Design of a flexible 2-D discrete wavelet transform IP core for JPEG2000 image coding in embedded imaging systems,” *Signal Processing* **86**, pp. 1375–1399, 2006.
26. J. Li, “Image compression: The mathematics of JPEG 2000,” *Modern Signal Processing* **46**, pp. 185–221, 2003.
27. M. Unser and T. Blu, “Mathematical properties of the JPEG2000 wavelet filters,” *IEEE Transactions on Image Processing* **12**, pp. 1080–1090, September 2003.
28. D. T. Lee, “JPEG 2000: Retrospective and new developments,” *Proceedings of the IEEE* **93**(1), pp. 32–41, 2005.
29. “Special issue on color image processing,” *IEEE Signal Processing Magazine* **22**(1), 2005.
30. H. J. Trussell, E. Saber, and M. Vrhel, “Color image processing. basics and special issue overview,” *IEEE Signal Processing Magazine* **22**(1), pp. 14–22, 2005.
31. Z. Liu, L. J. Karam, and A. B. Watson, “JPEG2000 encoding with perceptual distortion control,” *IEEE Transactions on Image Processing* **15**(7), pp. 1763–1778, 2006.
32. K. Hirakawa and T. W. Parks, “Joint demosaicing and denoising,” *IEEE Transactions on Image Processing* **15**(8), pp. 2146–2157, 2006.
33. K. Hirakawa and T. W. Parks, “Image denoising using total least squares,” *IEEE Transactions on Image Processing* **15**(9), pp. 2730–2742, 2006.
34. M. Unser, “Splines. a perfect fit for signal and image processing,” *IEEE Signal Processing Magazine* **16**, pp. 22–38, 1999.
35. P. Thévenaz, T. Blu, and M. Unser, “Interpolation revisited,” *IEEE Transactions on Medical Imaging* **19**(7), pp. 739–758, 2000.
36. T. Blu, P. Thévenaz, and M. Unser, “Complete parameterization of piecewise-polynomial interpolation kernels,” *IEEE Transactions on Image Processing* **12**, pp. 1297–1309, November 2003.
37. T. Blu, P. Thévenaz, and M. Unser, “Linear interpolation revitalized,” *IEEE Transactions on Image Processing* **13**, pp. 710–719, May 2004.
38. F. M. de Saint-Martin, P. Siohan, and A. Cohen, “Biorthogonal filterbanks and energy preservation property in image compression,” *IEEE Transactions on Image Processing* **8**(2), pp. 168–178, 1999.
39. W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” *Appl. Comput. Harmon. Anal.* **3**(2), pp. 186–200, 1996.
40. D. D. Muresan and T. W. Parks, “Demosaicing using optimal recovery,” *IEEE Transactions on Information Theory* **14**(2), pp. 267–278, 2005.
41. X. Li, “Demosaicing by successive approximation,” *IEEE Transactions on Image Processing* **14**(3), pp. 370–379, 2005.
42. M. Pawlak, E. Rafajłowicz, and A. Krzyżak, “Postfiltering versus prefiltering for signal recovery from noisy samples,” *IEEE Transactions on Information Theory* **49**(12), pp. 3195–3212, 2003.
43. M. Unser and A. Aldroubi, “A review of wavelets in biomedical applications,” *Proceedings of the IEEE* **84**, pp. 626–638, 1996.
44. Y. Meyer, *Wavelets and operators*, Cambridge University Press, Cambridge, 1992.
45. Y. Meyer, S. Jaffard, and R. Ryan, *Wavelets & Applications*, SIAM, Philadelphia, 2000.