Fast Colour2Grey

***Ali Alsam*[1] *and Mark S. Drew*[2]**
[1]**The Scientific Department, The National Gallery London, Trafalgar Square, London WC2N 5DN, U.K., ali.alsam@gmail.com**
[2]**School of Computing Science, Simon Fraser University, Vancouver, British Columbia, Canada V5A 1S6, mark@cs.sfu.ca**

Abstract

A standard approach to generating a greyscale equivalent to an input colour image involves calculating the so-called structure tensor at each image pixel. Defining contrast as associated with the maximum-change direction of this matrix, the grey gradient is identified with the first eigenvector direction, with gradient strength given by the square root of its eigenvalue. However, aside from the inherent complexity of such an approach, each pixel's gradient still possesses a sign ambiguity*, since an eigenvector is given only up to a sign. This is ostensibly resolved by looking at how one of the R,G,B colour channels behaves, or how the the luminance changes. Instead, we would like to circumvent the sign problem in the first place, and also avoid calculating the costly eigenvector decomposition. So here we suggest replacing the eigenvector approach by generating a greyscale gradient equal to the maximum gradient amongst the R,G,B gradients, in each of* x, y*. But in order not to neglect the tensor approach, we consider the relationship between the complex and the simple approaches. We also note that, at each pixel, we have both forward-facing and backward-facing derivatives, which are different, and we consider a tensor formed from both. Then, over a standard training set, we ask for an optimum set of weights for all the maximum gradients such that the simple maxima scheme generates a greyscale structure tensor to best match the original, colour, one. We find that a simple scheme that facilitates fast solutions is best. Greyscale results are shown to be excellent, and the algorithm is very fast.*

1. Introduction

Colour images contain information about the intensity, hue and saturation of the physical scenes that they represent. From this perspective, the conversion of colour images to black and white has long been defined as: The operation that maps RGB colour triplets to a space which represents the luminance in a colour-independent spatial direction. As a second step, the hue and saturation information are discarded, resulting in a single channel which contains the luminance information.

In the colour science literature, there are, however, many standard colour spaces that serve to separate luminance information from hue and saturation. Standard examples include: CIELab, HSV, LHS, YIQ etc. But the luminance obtained from each of these colour spaces is different.

Assuming the existence of a colour space that separates luminance information perfectly, we obtain a greyscale image that preserves the luminance information of the scene. Since this information has real physical meaning related to the intensity of the light signals reflected from the various surfaces, we can redefine the task of converting from colour to black and white as: An operation that aims at preserving the luminance of the scene.

In recent years, research in image processing has moved away from the idea of preserving the luminance of a single image pixel to methods that include spatial context, thus including simultaneous contrast effects. Thus we need to generate the intensity of an image pixel based on its neighbourhood. Further, for certain applications, preserving the luminance information per se might not result in the desired output. As an example, an equiluminous image may easily have pixels with very different hue and saturation. However, equating grey with luminance results in a flat uniform grey. So we wish retain colour regions while best preserving achromatic information. Finally, in a recent study, Connah et al. [1] compared six algorithms for converting colour images to greyscale. Their findings indicate that the use of spatial algorithms results in visually preferred rendering.

To proceed, we state that a more encompassing definition of colour to greyscale conversion is: An operation that reduces the number of channels from three to one while preserving certain, user defined, image attributes. As an example, Bala and Eschbach [2], introduced an algorithm to convert colour images to greyscale while preserving colour edges. This is achieved by transforming the RGB image to an opponent colour space, extracting the luminance image and adding chrominance edges that are not present in the luminance. In this sense, the algorithm aims at preserving chrominance edges.

In this paper, we present an algorithm that builds upon the work of Socolinsky and Wolff [3, 4], who developed a technique for multichannel image fusion with the aim of preserving *contrast*. In their work, these authors use the Di Zenzo structure-tensor matrix [5] to represent contrast in a multiband image. The interesting idea added to [5] was to suggest re-integrating the gradient produced in Di Zenzo's approach into a single, representative, grey channel encapsulating the notion of contrast.

We briefly recapitulate this method here, and start by defining the contrast in a greyscale image as the directional derivative of the image intensity map in the maximum-change direction joining the given pixel with its neighbours. In other words, greyscale contrast is the maximum intensity variation due to an infinitesimal spatial variation. Socolinsky and Wolff's work shows that it is possible to use the Di Zenzo matrix definition of contrast in an n-dimensional image to formulate an equivalent greyscale output.

The Di Zenzo matrix allows us to represent contrast at each image pixel by utilising a 2×2 symmetric matrix whose elements are calculated based on the derivatives of the colour channels in the horizontal and vertical directions. Socolinsky and Wolff defined the maximum absolute colour contrast to be the square root of the maximum eigenvalue of the Di Zenzo matrix along the direction of the associated eigenvector. In [3], Socolinsky and Wolff noted that the key difference between contrast in the greyscale case and that in a multiband image is that, in the latter, there is no preferred *orientation* along the maximum contrast direction. In other words, contrast is defined along a line, not a vector. To resolve the resulting sign ambiguity, the authors suggest having the orientation of one of the colour channels, or alternatively the luminance function, serve as a representation of a smooth function indicating vector sense.

The contribution of this paper is twofold. Firstly, we present an algorithm to calculate contrast in colour and multiband images that *does not require the calculation of the eigen-decomposition of the Di Zenzo matrix*. Instead, we make use of simple gradient maxima of each of the colour channels, both in the forward-facing gradient direction and in the backward-facing one. This results in a much faster achromatic approximation of n-dimensional contrast but also, and most importantly, obviates the sign-ambiguity problem and as a result removes unpleasant artifacts that result from getting the sign wrong. The second contribution is that our algorithm incorporates a definition of orientation that is independent of any particular single colour channel. Finally, using non-linear optimisation we show that there is a mathematical equality between the new definition of contrast and that obtained by Di Zenzo's matrix: we generate a greyscale representation that produces a Di Zenzo matrix that, in a least-squares sense over a set of training images, best matches the original tensor that derives from the colour image. Thus we import into the grey regime the same definition of colour contrast, but generated in a simple involving color-channel gradient maxima. The resulting scheme can be implemented as a very fast Fourier-based algorithm.

2. Colour Contrast and Grey Contrast

Suppose a colour image is denoted $\boldsymbol{\rho}$, with components $k = 1..3$. Let the two components of the gradient image for channel $k = 1..3$ be $\rho^k_{,x}$, $\rho^k_{,y}$: respectively the partial derivatives $\partial\rho^k/\partial_x$ and $\partial\rho^k/\partial_y$. Then we can form Di Zenzo's *structure tensor* $\boldsymbol{Z}$ as the symmetric 2×2 matrix

$$\boldsymbol{Z} \;=\; \begin{pmatrix} \sum_k \rho^k_{,x}\rho^k_{,x} & \sum_k \rho^k_{,x}\rho^k_{,y} \\ \sum_k \rho^k_{,x}\rho^k_{,y} & \sum_k \rho^k_{,y}\rho^k_{,y} \end{pmatrix} \tag{1}$$

Since $\boldsymbol{Z}$ is symmetric, its eigenvectors form an orthogonal matrix, $\boldsymbol{V}$, with columns $\boldsymbol{v}$ such that

$$\boldsymbol{Z}\,\boldsymbol{v}_i = \lambda \boldsymbol{v}_i, \quad i = 1..2 \tag{2}$$

The eigenvector associated with the largest eigenvalue points in the (unsigned) direction of of maximum contrast [5]. So the gradient direction for generating an output greyscale image is taken to be the maximum-eigenvalue direction $\boldsymbol{v} = \boldsymbol{v}_1$.

Now for a *grey* image g, the Di Zenzo matrix becomes instead

$$\boldsymbol{Z}_g \;=\; 3 \begin{pmatrix} g_{,x}g_{,x} & g_{,x}g_{,y} \\ g_{,x}g_{,y} & g_{,y}g_{,y} \end{pmatrix} \tag{3}$$

with the 3 to match the $\boldsymbol{Z}$ from eq. (1) for a colour image.

The question we would like to answer is: Given $\boldsymbol{Z}$, belonging to the original colour image, what greyscale representation g produces a Di Zenzo matrix $\boldsymbol{Z}_g$ that best matches the matrix $\boldsymbol{Z}$ for the input colour image?

Clearly, we could use the maximum-contrast direction $\boldsymbol{v}$ above, and generate a greyscale g from $\boldsymbol{v}$ and the gradient strength given by the square root of the first eigenvalue. However, we would like to in the first place generate a grey gradient without the speed penalty of finding eigenvectors, and as well we would like to circumvent the issue of choosing a gradient vector sense — i.e., the problem that sign is not defined for an eigenvector.

We argue that we can solve both these problems by the straightforward scheme of using the maximum change over all the colour channels, R,G,B, as a simple but effective approximation to the eigenvector approach. Moreover, we notice that at each pixel there are two possible choices for the maximum gradient, since we can use either front-facing and backward-facing derivatives (speaking of the horizontal direction; and similarly for the vertical direction). Is there a combination of *all* the gradient information available in all these colour gradients derived from maxima, such that we can get a best approximation to Di Zenzo's definition of contrast?

3. Fast Maximum Algorithm

In Di Zenzo's paper [5] the author states that simpler representative grey versions of the colour gradient than the eigenvector-based one above could be used. E.g., the root-mean-square, over colour, of the colour-channel x-gradients could possibly be used as the x-component of the representative grey gradient. Alternatively, "the RMS could be replaced by the sum, or even the maximum, of the absolute values of the differences involved."

Here, our motivation is to make Di Zenzo's method *fast*, by omitting the eigenvalue determination step entirely, and replacing it by a simple calculation of the *maximum*, as suggested. However, at each pixel we note that we have in fact two possible maxima in R,G,B: one forward-facing and (the negative of) another, backward-facing, as in Fig. 1. So far as we are aware, no-one has made use of a simple maximum scheme over colour channels, or of all the information available at each pixel of a multi-valued image. Here, we look at both differences ending at a pixel, West and East for the horizontal maximum, and North and South for the vertical.

Fig. 1 shows the Cartesian direction gradients for each of the R,G,B colour channels. Here, in the East direction, we would choose Red as the maximum horizontal component. And for the vertical South direction, we would choose Blue. Cycling through Red, Green, and Blue we would thus take as the correct grey gradient component the maximum over all of R,G,B, separately in each of the two coordinate directions, x and y and separately for the forward-facing and backward-facing directions.

In the simplest incarnation of the idea now, we could simply use the East and South maxima as the grey gradient components, and re-integrating these into a single scalar field, we arrive at a greyscale output.

But suppose we now ask: In each direction, what combination of the front-facing and backward-facing maxima over the colour channels, which we just chose, gives a matrix $\boldsymbol{Z}_g$ that best matches the matrix $\boldsymbol{Z}$ for the input colour image?

4. Learning Maximum-Gradient Weights

In Fig. 1, we show the representative grey gradient as a dashed vector, centered on the current pixel. This is the gradient that would be found using an eigenvalue decomposition of eq. (1). If we use a simple forward-difference definition of derivative, then, for the x direction, we consider the difference between the pixel to the right and the current pixel. Usually, we take that value to be the x-derivative and leave it at that. However, here we wish to match the (dashed in Fig. 1) gradient, determined by the Di Zenzo construction, as well as possible and under the constraint that we utilise only the maxima in each channel. Since we in fact have available the colour-differences in all four directions, we make use of these to obtain a best approximation of the

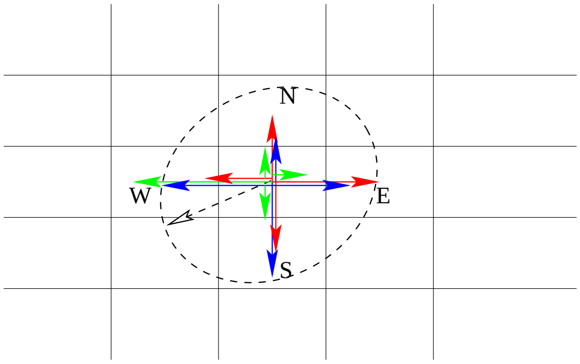


Figure 1. Using maximum horizontal and vertical gradients, over colour channels, versus using the maximum-contrast eigenvector (dashed) of the Di Zenzo matrix. Here, Red is max in the N direction, Blue is max for S, Red is max for E, and Green is max for W.

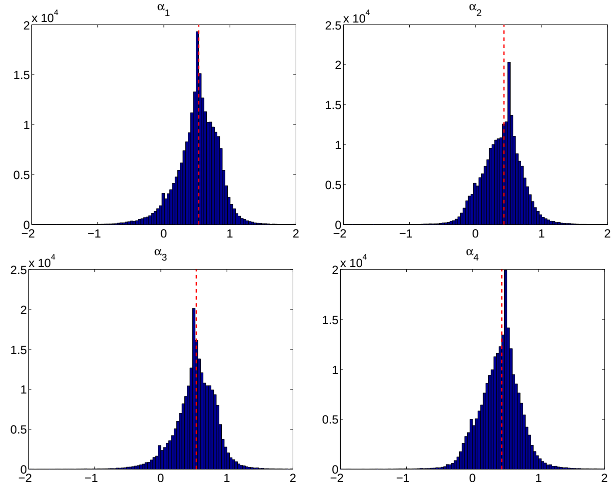


Figure 2. Optimum weights at each pixel, over random pixels in training images.

Di Zenzo gradient shown in Fig. 1 from the available maximum differences shown as the longest undashed arrows.

To do so, we randomly select pixels from various images. For each location, we generate a Di Zenzo matrix $\boldsymbol{Z}_g$ associated with the grey image produced by using maxima from colour channel gradients. We select weights $\alpha, \beta, \gamma, \delta$ at each pixel location so as to to optimally combine the N,S,E,W colour gradients, where in each of the four directions, the gradient in that direction is taken to be the largest change over R or G or B. An optimum is defined as corresponding to that set of weights multiplying these four maximum change vectors that produces the best match to the Di Zenzo matrix $\boldsymbol{Z}$ for the original, colour image composed of both front- and back-facing derivatives.

That is, at a particular x, y, we look for weights $\boldsymbol{\alpha} = \{\alpha, \beta, \gamma, \delta\}$ such that the gradient ∇g satisfies the following optimisation:

$$\begin{array}{l}
\text{with colour gradient } \nabla\boldsymbol{\rho} \;=\; \{\rho^k_x\,,\, \rho^k_y\},\;\; k = 1..3, \\[1em]
\text{find scalar-field gradients} \\[1em]
\nabla g^E_{,x} \;=\; \overset{\max_abs}{k} \left(\rho^{k,E}_{,x}\right) \\
\text{where } E \text{ is the East-facing derivative;} \\
\text{and similarly for W,S,N.}
\end{array} \tag{4}$$

Now combine the candidate gradient terms into a single gradient pair

$$\begin{aligned}
g_{,x} &= \alpha g^E_{,x} + \beta g^W_{,x} \\
g_{,y} &= \gamma g^S_{,y} + \delta g^N_{,y}
\end{aligned} \tag{5}$$

where $\boldsymbol{\alpha}$ is chosen by the minimisation

$$\overset{\min}{\boldsymbol{\alpha}} \; \|Z - Z_g\| \tag{6}$$

with $\boldsymbol{Z}_g$ formed via eq. (3) using $\nabla g = \{g_{,x}, g_{,y}\}$.

Here we used a standard set of colour images [1] and sampled 10,000 pixel locations from each image.

[1] We used the KodakCD images from http://www.cipr.rpi.edu/resource/stills/kodak.html.

We had expected to find that $\boldsymbol{\alpha} = 0.5$ for all four values. Since the histograms for each component of $\boldsymbol{\alpha}$ is quite peaked, we simply used the median for each (see Fig. 2.) We found median values $\boldsymbol{\alpha} = \{0.530, 0.430, 0.537, 0.443\}$; these values indicate a property of natural image statistics. The optimisation is very quick since it is a quadratic program (and is offline in any event).

Once we have a set of gradients, these must be combined in order to re-integrate the derivatives ∇g into a single, unified grey image g. I.e., at pixel (i, j), suppose our maximum-change gradients are $g^E_{,x}$, $g^W_{,x}$, $g^S_{,y}$, $g^N_{,y}$, where we are careful to have all definitions of the sign of gradients facing the same way. We wish to find greyscale image $g(i, j)$ such that

$$\begin{aligned}
&\alpha[g(i+1,j) - g(i,j)] + \beta[g(i,j) - g(i-1,j)] + \\
&\gamma[g(i,j+1) - g(i,j)] + \delta[g(i,j) - g(i,j-1] \\
&= \alpha g^E_{,x} + \beta g^W_{,x} + \gamma g^S_{,y} + g^N_{,y} \\
&\equiv RHS
\end{aligned}$$

or in other words

$$\begin{aligned}
g(i,j) \;=\; &\{\alpha g(i+1,j) - \beta g(i-1,j) \\
&+\gamma g(i,j+1) - \delta g(i,j-1) - RHS \\
&\}/(\alpha - \beta + \gamma - \delta)
\end{aligned} \tag{7}$$

Clearly, this is an extension of the spatial domain solution of a Poisson equation. It is an extension because we weight the forward-facing and backward-facing derivatives differently. We solve it using Jacobi iteration, with homogeneous Neumann boundary conditions to ensure zero derivative at the image boundaries.

Overall, the idea of using the colour structure tensor $\boldsymbol{Z}$ is useful, in that we can determine just what combination of maximum gradient information best matches the colour gradient. However, output greyscale images using the above weights $\boldsymbol{\alpha}$ are in practice quite close to those simply using one-sided gradients, i.e., $\boldsymbol{\alpha} = \{1, 0, 1, 0\}$. Therefore while in theory the two-sided gradients do provide more information, in fact the one-sided gradients are faster to calculate and already produce excellent results.

Alternatively, if we were willing to adaptively select weights $\boldsymbol{\alpha}$ for each image, we could certainly obtain the optimum result compared to the simplified, one-sided derivative method.

Generally, the weighted method will produce a smoother output greyscale than the simplified method, and this may be useful.

5. Fourier-Based Re-Integration of Gradients

Instead of solving in the spatial domain for the greyscale corresponding to a putative set of gradients, as in eq. (7), we can employ a much faster method, employing the transform equation in the frequency domain. Here we make use of the Frankot-Chellappa algorithm [6], which works by projecting the putative set of gradients over an image onto an integrable convex set of gradients written in Fourier space. This algorithm then essentially takes a second derivative, by multiplying by the transform of the derivative operator in the frequency domain, and solves the resulting Poisson equation by going back to the spatial domain. Again, here we use homogeneous Neumann boundary conditions. To employ this algorithm, we must make use of one-sided derivatives only, since the transform in frequency space is defined for simple, unweighted pixel difference operations. Much more complicated methods are indeed available [7], but these are comparatively very slow indeed.

Overall, greyscale output image results are very close, for either an iterative or a Fourier-based approach. The main emphasis is that we make use of the maximum over colour gradients, in each direction. Results shown are from using the simplest and fastest method — single-sided derivatives and Fourier integration.

5. Experiments

Fig. 3 shows several examples of the new, fast method, compared to using either luminance or Socolinsky and Wolff's original method. Results are seen to be an improvement over the luminance, and also over the standard approach. In particular, in any situation where we can easily see colour edges, as in the painting in the top row of Fig. 3, it is easy to discern integrability errors associated with Socolinsky and Wolff's method (as discussed in [1]), whereas the present method does not produce these. As well, the method tends to change the dynamic range so that dark and bright areas are more visible, since it relies on not just the luminance change, but on the change over all three colour channels to generate its approximation of contrast.

6. Conclusions

We have outlined a new, fast method for converting still colour image data to greyscale, based on but not tied to the Di Zenzo structure tensor approach. Instead of having to calculate eigenvectors at each pixel, we calculate a simple maximum gradient based on the maximum over all colour channels. Results are seen to be an improvement over the luminance and also better than the standard, Socolinsky-Wolff approach (and, parenthetically, better than simply using histogram-equalisation on the luminance). As well, the algorithm is very fast.

References

[1] D. Connah, G.D. Finlayson, and M. Bloj. Seeing beyond luminance: A psychophysical comparison of techniques for converting colour images to greyscale. In *15th Color Imaging Conference: Color, Science, Systems and Applications.*, pages 336–341, 2007.

[2] R. Bala and R. Eschbach. Spatial color-to-grayscale transform preserving chrominance edge information. In *14th Color Imaging Conference: Color, Science, Systems and Applications.*, pages 82–86, 2004.

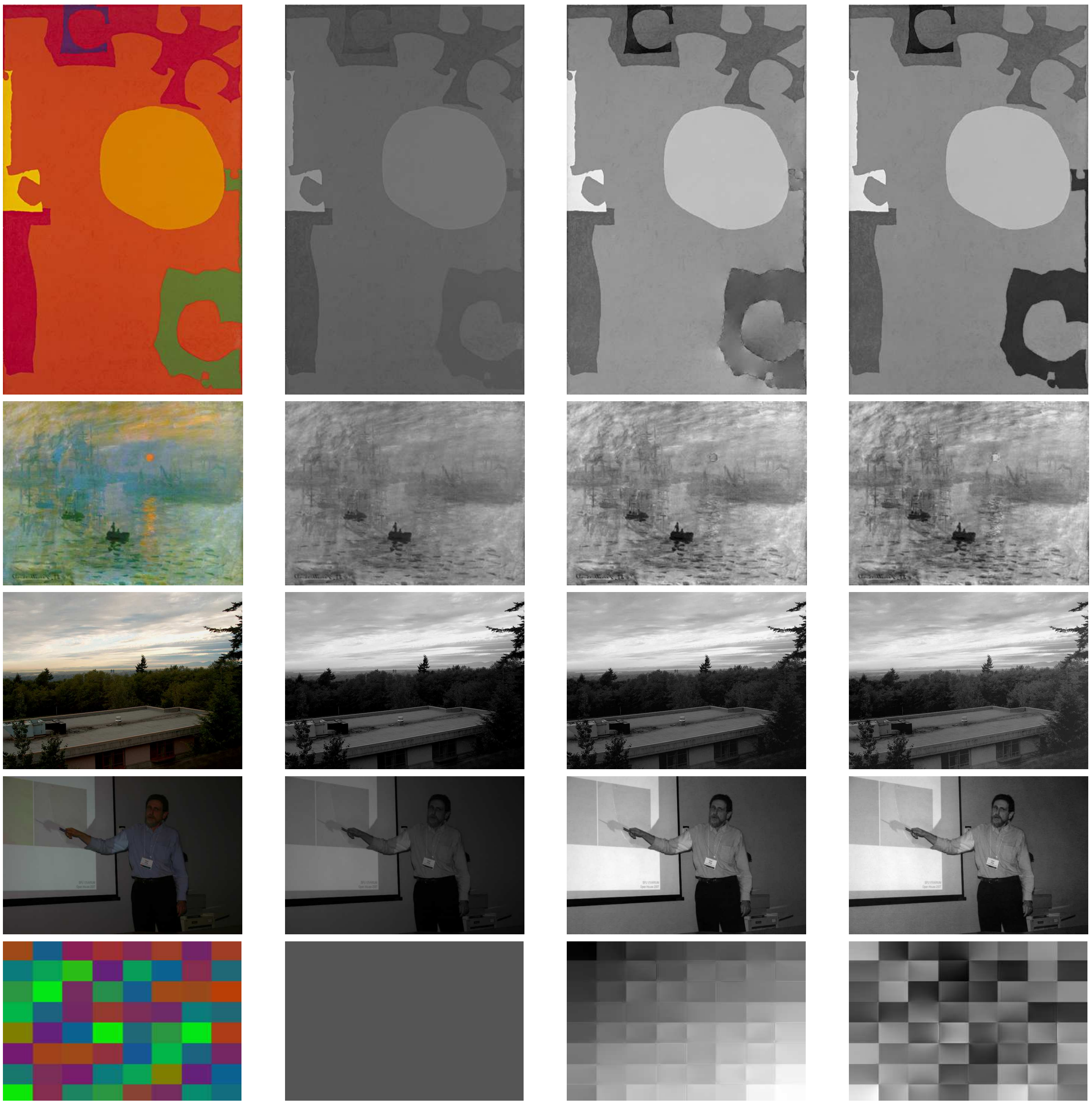
[3] D.A. Socolinsky and L.B. Wolff. A new visualization paradigm for multispectral imagery and data fusion. In *CVPR*, pages I:319–324, 1999.

[4] D.A. Socolinsky and L.B. Wolff. Multispectral image visualization through first-order fusion. *IEEE Trans. Im. Proc.*, 11:923–931, 2002.

[5] S. Di Zenzo. A note on the gradient of a multi-image. *Comp. Vision, Graphics, and Image Proc.*, 33:116–125, 1986.

[6] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 10:439–451, 1988.

[7] A.K. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In *ECCV 2006*, pages 578–591, 2006.



(a) (b) (c) (d)

Figure 3. (a): Original colour image. (b): Luminance. (c): Socolinsky-Wolff greyscale. (d): Present method.