International Journal of Image and Graphics

Improvement of Colorization Realism via the Structure Tensor

Mark S. Drew
School of Computing Science, Simon Fraser University,
Vancouver, British Columbia V5A 1S6, Canada
mark@cs.sfu.ca

Graham D. Finlayson
School of Computing Sciences, The University of East Anglia
Norwich, NR4 7TJ, England
graham@cmp.uea.ac.uk

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Colorization is a colour manipulation mechanism employing user-assisted colour hints for changing greyscale images into coloured ones. Several colorization algorithms have been constructed, and many of these methods are able to produce appropriately colorized images given a surprisingly sparse set of hints supplied by the user. But these colour images may not in fact look realistic. Moreover, the contrast in the colorized image may not match the gradient perceived in the original greyscale image. We argue that it is this departure from the original gradient that contributes to the un-real appearance in some colorizations. To correct this, we make use of the Di Zenzo gradient of a colour image derived from the structure tensor, and adjust the colorized image such that the Di Zenzo definition of the maximum-contrast gradient agrees with the gradient in the original grey image. We present a heuristic method to this end and guided by this approach devise an optimization-based method. Our gradient projection tends to result in more natural-looking images in the resulting adjusted colorization. To explore the proposed method we utilize minimalist sets of colour hints and find in particular that "hotspots" of un-realistic colour are subdued into regions of more realistic colour. This paper is not aimed at introducing a new basic colorization but instead our method is meant to make any colorization look more realistic; we demonstrate that this is the case for several different basic methods. In fact, we even find that a very simplistic colorization algorithm can be used provided the projection proposed here is then used to make the colorization more realistic looking.

Keywords: Colorization; Di Zenzo; gradient; contrast.

1. Introduction

To carry out colorization, regions of images or video are partially hand-coloured by the user, and then various approaches are brought to bear to generate a full-colour image from the original, greyscale one [1,2,3,4,5,6,7,8,9,10,11,12]. Colorization can always be carried out in an iterative fashion, re-colouring regions that appear incorrect. However, an iterative approach defeats the purpose of colorization as a quick, easy method of generating colour from grey. In the extreme, it could be necessary to simply hand-colour the entire image, and we wish to assist in avoiding this.

Here, we consider what can be accomplished to make a simple colorization appear more realistic in the first place. To investigate this notion, we make of use of colorizations that are quite minimalist, so as to see clearly the effect of our algorithm.

For example, the greyscale image in Fig. 1(a), colorized via the algorithm in [1,13], generates the output in Fig. 1(c): the colorization clearly performs poorly, because of the minimality of the set of colour hints. Of course, we might better succeed by applying more colour hints, or by iterating the colorization process, but here we mean to examine improvement of colorization realism by exploring how such a minimalist colorization with few user-supplied colour hints might be made more realistic (with more hints always leading to better results, naturally). While agreeing in colour with the user's control points in Fig. 1(b), nonetheless the output in (c) has a strikingly unrealistic appearance (and cf. the discussion of image realism in [14]). We argue that this is in part due to the unconstrained generation of the colour, with *contrast* not truly linked to that in the original, grey image. Instead, we suggest automatically generating a more realistic (and hence more pleasing) result, such as that in Fig. 1(d). Here, the output colours may not match every input control colour. Instead, the output is modified by requiring the gradient of the generated colour image to match that of the input greyscale image. Similar improvement is achieved for method [2] (Fig. 1(g,h,i)) and method [4,15] (Fig. 1(j,k,l)).

But what is the gradient of a colour image? Here we adopt the definition of Di Zenzo [16]: the quadratic form given by the outer product of colour channel derivatives induces a metric (cf. [17]) with maximum-gradient direction that can be taken as the maximum direction of contrast. We can go on to use this gradient (i.e., derivative) field to form a re-integrated (i.e., integral of gradient) greyscale image that best captures the contrast in a vectorial image, coded as a single greyscale output [18]. This generated greyscale image has local luminance-contrast that reflects the colour-contrast of the original image. Arguably, this greyscale image of the contrast is the most faithful reflection of our perception of gradient and edge, moving over from the colour domain to the greyscale one — the Di Zenzo definition of grey gives a good impression of the contrast that was contained in the colour image.

Here we adapt this greyscale gradient to the present case of colorization and see how to go back from grey to colour instead of from colour to grey. The gradient is found by diagonalizing the Di Zenzo "structure tensor" matrix, which is a 2×2 matrix formed from the R,G,B colour gradients. Starting from the Di Zenzo matrix for any colour image, and in particular for the output of a colorization routine, there is no guarantee that the resulting greyscale field at all matches the input grey image. Indeed, it is quite unlikely. In this paper, we mean to alter the colorized version so that its gradient exactly matches that for the input greyscale image. In doing so, we may somewhat shift the user-chosen control colours, as in Fig. 1(d). But we contend that the overall impression is a more realistic colorized image.

We present a threefold investigation of the problem. In the first instance, we examine the relationship between the structure tensor for a colorized image and that for the input, greyscale one. For the contrast to agree, the eigenvector structure of the grey image must be matched by the new, altered colour image. Here, in the first instance we adopt a simple heuristic strategy to set the two structure tensors equal. Results are strikingly more realistic (for these minimalist and therefore quick) colorizations than for the results from [1] (using its authors' supplied reference implementation [13]), or from the method [2], or from the method [4].

This discussion serves to motivate and set the stage for a second approach, in which we examine the space of *all* such alterations of the initial colorized image such that the structure tensor equals that for the grey image. By insisting that the adjusted colour gradient approximate the putative colour gradient, we arrive at a theorem which gives the desired adjustment in terms of a closed-form least squares solution. Results are again considerably more realistic, slightly better than the heurisitc method results. To better understand the performance of our algorithm we carried out the following test. We took a known image, converted it to greyscale and then made colour hints using the colours in the original colour image. Here a perfect colorization algorithm should be able to recreate an image that is close to the original. Comparing to previous work, we show that our gradient correction procedure gives a closer correspondence to ground truth.

Finally, in a third demonstration, we show that even a very simplistic colorization scheme can be used, one that yields quite unacceptable results, provided the projection scheme outlined here is then applied. Adjustment of the colorized image leads to passable colorized outputs even with a very quick and simple initial colorization.

In §2, we review the colour structure tensor and in §3 consider what colour tensors could generate an input greyscale image. In §4 we state a method for adjusting the colour gradient such that it does generate an identical luminance-contrast gradient as for the input grey image by suggesting a heuristic strategy that guarantees this result. This serves to motivate a theorem we give in §5 showing that there is in fact a best-possible solution for an adjusted colorized image, based on a rotation generated from an Orthogonal Procrustes problem. In §6 we develop a very simple colorization scheme, generating output very quickly, to determine how the colour gradient projection introduced here improves even very simple colorization algorithm results. While not acceptable in its first level of output, we show that the colour-projection scheme used here to develop realism in colorized images produces quite acceptable results, yielding a possible fast colorization scheme. In §5.2 and §7 we examine the accuracy of the three methods presented for re-generating an *input colour image* from its grey version plus colour hints at user-supplied locations. Not unexpectedly, the third, simplistic, schemes fares worst, while the scheme based on our least squares based theorem fares best.

2. Structure Tensor

2.1. *Colour to Greyscale*

In order to discuss the use of the structure tensor in guiding colorization, first we briefly recapitulate the related problem of generating a greyscale image from a colour one (the obverse of the problem at hand).

Suppose in the general case the vector-valued image C has components (colours) $k = 1..\mu$. Let the two components of the gradient image for channel k be $C^k_{,x}$, $C^k_{,y}$: respectively the partial derivatives $\partial C^k/\partial x$ and $\partial C^k/\partial y$. Then we can form the *structure tensor* $\boldsymbol{Z}$ as the symmetric 2×2 matrix

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

In differential geometry this 2×2 tensor is known as the First Fundamental Form [17] and induces a metric wherein the maximal and minimal rates of change in the vector image are given by its eigenvectors and eigenvalues. Since $\boldsymbol{Z}$ is real symmetric, its eigenvectors form an orthogonal matrix, $\boldsymbol{V}$, with columns $\boldsymbol{v}$ such that

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

The eigenvector associated with the largest eigenvalue points in the direction of of maximum contrast [16].

Let the 2×2 matrix $\boldsymbol{V} \;=\{\boldsymbol{u}\,,\boldsymbol{v}\,\}$, with $\boldsymbol{u} \;\perp \boldsymbol{v}$. If we decompose $\boldsymbol{Z}$ as

$$\boldsymbol{Z} \;=\; \boldsymbol{V}\,\boldsymbol{\Lambda}\,\boldsymbol{V}^{\,T}, \quad \boldsymbol{\Lambda} \;=\; diag(\kappa, \lambda), \;\; \kappa \leq \lambda \tag{3}$$

then we can take $\boldsymbol{v}$ as the (normalized) maximum-contrast gradient direction, with the gradient norm being $\sqrt{\lambda}$. Since the sign of an eigenvector is not unique, vector sense is taken in the direction of increasing luminance.

Re-integrating this gradient, by taking another derivative and solving Poisson's equation [18], generates a greyscale image whose gradient best matches the contrast in the original colour image. For completeness we state how re-integration of a putative, non-integrable gradient field proceeds using the Poisson equation. Suppose we identify a vector $\boldsymbol{v}$ with the gradient $g_{,x}$,$g_{,y}$ of a desired greyscale field g. Then the best least-squares solution of the optimization problem

$$\min_g \int (g_{,x} - v_x)^2 + (g_{,y} - v_y)^2 \; dx\, dy$$

is given by taking the variational derivative of the above objective function, With the solution g obtained as the solution of the resulting Euler-Lagrange equation. Here, this yields the Poisson equation in the unknown g:

$$\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \;=\; \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}$$

The solution of this equation is a classical problem, depending upon the boundary conditions chosen; here we make use of the modern formulation [19] which operates in the Fourier domain, and use homogeneous Neumann boundary conditions.

2.2. *Tensor for a Colour Image*

In the case of a 3-band colour image $\boldsymbol{\rho} \;= \{R, G, B\}$, consider the gradient colour image. We denote the gradient by $\boldsymbol{\nabla\rho}$, a 2×3 array:

$$\boldsymbol{\nabla\rho} \;=\; \begin{pmatrix} R_{,x} & G_{,x} & B_{,x} \\ R_{,y} & G_{,y} & B_{,y} \end{pmatrix} \tag{4}$$

Then eq. (1) reduces to the following:

$$\boldsymbol{Z} \;=\; \boldsymbol{\nabla\rho}\,\boldsymbol{\nabla\rho}^{T} \tag{5}$$

The maximum-eigenvalue direction $\boldsymbol{v}$ is taken to be the gradient direction for generating an output greyscale image. Note that only in the case that the gradient for each colour channel matches the gradient ∇g for grey is the maximum eigenvector sufficient to describe the colour gradient. In that case the second eigenvalue of the Di Zenzo matrix would be neglectable; but in fact that is certainly not the case in general since the ratio of the two eigenvalues ranges up to 1.0 in any typical colour image.

3. Grey to Colour

We would like to start from the structure tensor for a greyscale image and *derive* the properties that a matching colour image must have so as to deliver the same contrast as in the input, grey image. Specifically, given an input greyscale image, we certainly have the gradient direction $\boldsymbol{v}$; also, the orthogonal vector is known: $\boldsymbol{u} = \boldsymbol{v}^{\perp}$. What are the properties of a putative colour image that gives a tensor with the same $\boldsymbol{v}$ main eigenvector, or possibly the same exact tensor, as that for the greyscale image?

Firstly, we can ask what colour gradient produces the same structure tensor as a given tensor $\boldsymbol{Z}$, given the pair of directions $\boldsymbol{u}$,$\boldsymbol{v}$. Let the kth column of the 2×3 colour gradient $\boldsymbol{\nabla\rho}$ for the sought colour image be the 2-vector $\boldsymbol{\nabla}\rho_k$, $k = 1..3$. Then any $\boldsymbol{\nabla}\rho_k$ at a pixel must be a linear combination of $\boldsymbol{u}$ and $\boldsymbol{v}$:

$$\boldsymbol{\nabla\rho}_{k} \;=\; \alpha_k\sqrt{\kappa}\,\boldsymbol{u} \;+\; \beta_k\sqrt{\lambda}\,\boldsymbol{v} \tag{6}$$

Hence clearly for $\boldsymbol{\nabla\rho}\,\boldsymbol{\nabla\rho}^{T}$ to equal the $\boldsymbol{Z}$ in eq.(3), we must have

$$\begin{aligned} \boldsymbol{\nabla\rho}\,\boldsymbol{\nabla\rho}^{T} \;=\; & \textstyle\sum_{k=1}^{3}\{\alpha_k^2\kappa\boldsymbol{u}\,\boldsymbol{u}^{T} + \beta_k^2\lambda\boldsymbol{v}\,\boldsymbol{v}^{T} \\ & +\alpha_k\beta_k\sqrt{\kappa\lambda}\,(\boldsymbol{u}\,\boldsymbol{v}^{T} + \boldsymbol{v}\,\boldsymbol{u}^{T})\} \\ \equiv\; & \kappa\boldsymbol{u}\,\boldsymbol{u}^{T} \;+\; \lambda\boldsymbol{v}\,\boldsymbol{v}^{T} \end{aligned} \tag{7}$$

Thus α_k and β_k consist of a pair of orthonormal 3-vectors. I.e., in order to match a particular Di Zenzo matrix $\boldsymbol{Z}$, possible colour edges $\boldsymbol{\nabla\rho}$ can only differ from the pair of vectors $\{\sqrt{\kappa}\boldsymbol{u}\;,\;\sqrt{\lambda}\boldsymbol{v}\}$ by two rows of a 3×3 orthogonal transformation. Call this 2×3 rotation $\boldsymbol{O}$, with $\boldsymbol{O}\,\boldsymbol{O}^{T} = I$. Then we have

$$\boldsymbol{\nabla\rho} \;=\boldsymbol{V}\,\sqrt{\boldsymbol{\Lambda}}\;\boldsymbol{O}\;, \tag{8}$$

where $\boldsymbol{V} = \{\boldsymbol{u}\,,\boldsymbol{v}\}$. The $\boldsymbol{Z}$ for such colour edges $\boldsymbol{\nabla\rho}$ just equals $\boldsymbol{Z}$ in eq. (3).

Thus we know in general how to match a structure tensor. Below we first consider matching the major eigenvector, and then motivate simply matching $\boldsymbol{Z}$ exactly, via a particular strategy.

It is worth pointing out that the Conditional Image Diffusion approach [11] is also concerned with using the gradient of the input grey image to diffuse user-specified chromatic hints, with results similar to those using [17] in the same way. In fact [11] provides a theoretical study of a general PDE diffusion equation that uses one guiding field for diffusion of vector fields. The difference is that here we tie contrast to the Structure Tensor, and seek a way to post-process any colorization so as to make the result more realistic looking. As well, the proposed method is considerably simpler than PDE-based methods such as [12,11].

4. Closest Colorized Gradient to Grey Gradient

4.1. *Modified Colour Gradient: Heuristic Procedure*

Suppose we have established a tentative colorized version $\widetilde{\boldsymbol{\rho}}$ of a grey image. Let the structure tensor corresponding to that colour image be $\widetilde{\boldsymbol{Z}}$. Also, let the gradient of the original source, grey, image g be $\boldsymbol{\nabla} g = \{g_{,x}, g_{,y}\}^T$. And let the Di Zenzo matrix formed from $\boldsymbol{\nabla} g$ be $\boldsymbol{Z}_g$.

Now we would like to transform the current matrix $\widetilde{\boldsymbol{Z}}$ into one that gives as its major eigenvector the same $\boldsymbol{v}$ as for the original grey image. That is, we would like the perceived colour image gradient to match that for the grey gradient $\boldsymbol{\nabla} g$.

Suppose the current 2×3 colour edges are $\widetilde{\boldsymbol{\nabla}\boldsymbol{\rho}}$. Using Singular Value Decomposition (SVD), we can certainly factor $\widetilde{\boldsymbol{\nabla}\boldsymbol{\rho}}$ as

$$\widetilde{\boldsymbol{\nabla}\boldsymbol{\rho}} = \widetilde{\boldsymbol{V}} \sqrt{\widetilde{\boldsymbol{\Lambda}}}\, \widetilde{\boldsymbol{O}} \tag{9}$$

where $\boldsymbol{V} = \{\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{v}}\}$ approximates $\boldsymbol{V}$ and $\sqrt{\widetilde{\boldsymbol{\Lambda}}} = diag(\sqrt{\widetilde{\kappa}}, \sqrt{\widetilde{\lambda}})$ approximates $\sqrt{\boldsymbol{\Lambda}}$.

In contrast, for the original grey gradient we have simply

$$\boldsymbol{\nabla} g = \sqrt{\lambda}\boldsymbol{v}\,, \text{ with } \|\boldsymbol{v}\| = 1 \tag{10}$$

Clearly the simplest, heuristic, strategy for guaranteeing the same contrast in the colorized image as in the greyscale image is to replace $\widetilde{\boldsymbol{v}}$ by $\boldsymbol{v}$, $\widetilde{\boldsymbol{u}}$ by $\boldsymbol{u}$ (and recall: $\boldsymbol{u} = \boldsymbol{v}^{\perp}$), and $\widetilde{\lambda}$ by λ in $\widetilde{\boldsymbol{\nabla}\boldsymbol{\rho}}$, as posited in [20]. Then by eq. (8) the Di Zenzo matrix $\boldsymbol{Z}$ for the new set of colour edges, $\boldsymbol{\nabla}\boldsymbol{\rho}$, will yield precisely the original grey edges $\boldsymbol{\nabla} g$ as the solution to the Di Zenzo equations (1,3).

That is, the resulting matrix $\boldsymbol{Z}$ will have the same maximum-contrast direction $\boldsymbol{v}$ and strength $\sqrt{\lambda}$ as that of the original greyscale image.

But what should we choose for the free parameter $\widetilde{\kappa}$? By eqs. (1,3), any value will generate $\boldsymbol{v}$ provided $\widetilde{\kappa} \leq \lambda$.

If we use $\widetilde{\kappa} = 0$, we have exactly the *same* Di Zenzo matrix $\boldsymbol{Z}$ as the original $\boldsymbol{Z}_g$ for the grey image, and that is the value used in [20]. As an alternative, we found that instead leaving $\widetilde{\kappa}$ just as it arises in eq. (9) for the current colorized image does not change the colorized version greatly but has less beneficial effect on colorization artifacts we wish to attenuate. Hence as a heuristic for generating a modified colour gradient we do indeed use $\widetilde{\kappa} = 0$ and *replace the colour gradient in eq. (9)* as follows:

$$\widetilde{\boldsymbol{\nabla}\boldsymbol{\rho}} \;\rightarrow\; \boldsymbol{\nabla}\boldsymbol{\rho} \;\equiv\; \boldsymbol{V} \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix} \widetilde{\boldsymbol{O}} \tag{11}$$

The reader might think that we have implemented a rather extreme operation. But, a little thought illustrates this is not the case. The row of $\widetilde{\boldsymbol{O}}$ which we use is in fact the maximum variance direction of the 2×3 matrix of $x-$ and $y-$ colour derivatives. That is, if we were to choose one colour vector direction onto which the true derivatives were to be approximated then it would be this vector. Further, in reality, we expect the gradient direction $\boldsymbol{v}$ of the greyscale image to be of similar orientation to that found in the colour image. Indeed, where there is an *edge* in the image, the colour gradient vectors $\boldsymbol{\nabla\rho}$ will be (almost) 1-dimensional. Our derivation (6) through (8) enforces this constraint. Finally, the magnitude of the colour gradient might be different than the greyscale image. But, even here we expect similarity: we know that the brightness channel captures the most energy in the images and so expect that the energy in the brightness derivative will capture most of the energy in the colour gradients. Thus, our gradient correction procedure tends to 'tweak' the original colour gradients rather than making radical changes.

Note that this Heuristic method is *not* the same as simply deriving an R or G or B gradient closest to ∇g, with Dirichelet conditions imposed by the colour hints, as in [2], because of the colour gradient factor $\widetilde{\boldsymbol{O}}$, which is a function $\widetilde{\boldsymbol{O}}\ (\boldsymbol{x})$ of retinal location. And indeed results for [2] are quite different (Fig. 1(g,h)).

4.2. *Re-integration of Gradient*

Finally, we must generate a colour image from the colour gradients $\boldsymbol{\nabla\rho}$. To do so, we first go to a colour-opponent space $\{L, c_1, c_2\}$ [21], and use the input grey image g for the L channel. For our purpose here, any colour space separation into luminance and chrominance will work. For the chromaticity gradients, we project onto an integrable set of edges using the Frankot-Chellappa method [19] and re-integrate a Poisson's equation.

More explicitly, our colour-gradient replacement (11) generates a new colour-gradient $\boldsymbol{\nabla\rho}$. Now supposing we move into a colour-opponent colour space – the purpose is to be able to gaurd the input greyscale image from any changes. Since a colour-space transform is applied by multiplying colours by a matrix, the transform of $\boldsymbol{\nabla\rho}$ is linear and holds for the gradient as well. Now, for each colour channel we have a gradient-pair of values at each pixel, and our objective is to generate pixel values back from the now-known gradient. Instead of using the variational derivative approach as in §2.1 above, the method in [19] moves the problem into the Fourier domain and solves the same least-squares problem, but now phrased in terms of Fourier space quantities. The result is an analytic solution in Fourier space which effectively bypasses the need to actually solve Poisson's equation. (Note that this trick only work for rectangular image domains.) We apply homogeneous Neumann boundary conditions (zero derivative at the boundary) by simply flipping the image left-to-right and appending to the original image, and similarly flipping up-down.

With homogeneous Neumann boundary conditions, this leaves a free additive constant in each chroma channel (since the only input is a set of derivatives, not image values), which we set to match the mean value over the set of unique user input hint pixel colours. Alternatively, the additive constant could be set by matching a particular colour, e.g. skin tones.

4.3. *Heuristic Algorithm*

Thus, in sum the algorithm we propose here consists of implementing the best version as possible, at each pixel, of the heuristic (11): From the input grey image we know both the direction and strength of the correct contrast we are aiming at. Now supposing we have a putative colorized image $\widetilde{\boldsymbol{\rho}}$ we form its gradient $\widetilde{\boldsymbol{\nabla\rho}}$ and find its 2×3 colour part $\widetilde{\boldsymbol{O}}$ and $x-, y-$ eigenvector part $\widetilde{\boldsymbol{V}}$. We ascertain which eigenvector in $\widetilde{\boldsymbol{V}}$ is closest to $\boldsymbol{v}$ from the grey image and assign the correct sign to that eigenvector so as to match the grey gradient; this implies adjusting the sign of the corresponding row of $\widetilde{\boldsymbol{O}}$ as well. Once we replace the colour gradient with its heuristic version, we transform to color-opponent space and re-integrate the chromatic components.

Note that instead of re-integrating an achromatic channel derived from $\boldsymbol{\nabla\rho}$, we instead directly re-use the input grey channel. The reason for this replacement of a re-integrated luminance-channel gradient by the original input is that in fact we have a tensor whose eigenvectors, given perfect knowledge of sign choices, will integrate perfectly to the input grey. The algorithm generates altered colour gradients that generate the same tensor as the grey tensor exactly. However the projected colour gradient is not necessarily exactly equal to the gradient of the input grey because of the decisions we have made regarding which eigenvector is closest to the input grey gradient and also the consequent sign choices we must make. If instead we had a perfect sign-assignment, then the projected colour gradient would have, indeed, exactly the same achromatic channel as the grey gradient. Re-integrating, we would simply obtain the input grey back as our output grey channel. So instead of re-integrating over all three dimensions of $\boldsymbol{\nabla\rho}$, either in RGB or in colour-opponent space, we simply use the input grey as the output achromatic channel, replacing a possibly slightly wrong re-integration by a perfect one.

4.4. *Idempotency*

A useful property of an image processing algorithm is that it be idempotent — further applications of the method to the output do not change the result. To determine if the above heuristic algorithm is idempotent, we note that the colour-gradient projection postulated amounts to a rank-1 linear transform $\boldsymbol{P}$ as follows.

Firstly we note that we can solve for $\widetilde{\boldsymbol{O}}$ via eq. (9):

$$\widetilde{\boldsymbol{O}} = \left(\sqrt{\widetilde{\boldsymbol{\Lambda}}}\right)^{-1} \widetilde{\boldsymbol{V}}^{T} \left(\widetilde{\boldsymbol{\nabla\rho}}\right) \tag{12}$$

Hence we can rewrite the heuristic (11) as

$$\boldsymbol{\nabla\rho} = \boldsymbol{P}\, \widetilde{\boldsymbol{\nabla\rho}} \tag{13}$$

with

$$\boldsymbol{P} = (\boldsymbol{0}\ \boldsymbol{v}\ ) \sqrt{\lambda} \left(\sqrt{\widetilde{\boldsymbol{\Lambda}}}\right)^{-1} \widetilde{\boldsymbol{V}}^{T} \tag{14}$$

Putting this another way, we generate a modified colour gradient via

$$\boldsymbol{\nabla\rho} = (\boldsymbol{0}\ \boldsymbol{v}\ ) \sqrt{\lambda}\, \widetilde{\boldsymbol{O}} \tag{15}$$

From this relation, it follows that the method is indeed idempotent insofar as, if we were to substitute the output colour *gradient* $\boldsymbol{\nabla\rho}$ in place of the original colorized colour *gradient*

$\widetilde{\nabla \boldsymbol{\rho}}$ in this approach, we would arrive at the same answer, since we do not change the colour-contribution matrix $\widetilde{\boldsymbol{O}}$.

However, the method is nonetheless not idempotent. The reason is that we do not settle on a modified colour gradient and leave it at that: instead, we go on to generate a re-integrated colour image $\boldsymbol{\rho}$. Since re-integration forms a new image, one that has a changed gradient, we in fact change the input to a subsequent application of the algorithm and arrive at a changed output colour image.

The meaning of our colour-gradient projection can be gleaned from examination of eq. (9). Since we are removing a row of the colour information $\widetilde{\boldsymbol{O}}$ and keeping the main term involving strength $\sqrt{\lambda}$, we are in fact making the colorized image more grey, by removing some colour; of course, we remove colour in a careful way so as to generate the same contrast as in the input greyscale image. Fig. 1(e) shows the result after application of three repetitions of the algorithm — we can see that indeed the image has become desaturated, compared to a single application. Arbitrarily boosting the saturation back to the first projected image may help (although this often will produce negative pixel values), but does not much change this property. In the sequel, we therefore stick to a single application of the method.

Note that simply desaturating the input colorized image does not work (see Fig. 2), with saturation reduced by 40% of the input) because it does not repair the wrong colour gradient, unlike our proposed method, with the result that desaturating works quite haphazardly and also typically in the desaturated image the original colour hints can be seen too clearly.

5. Least Squares Projection of Colorized Gradient

5.1. *Optimization Approach for Colour Gradient Adjustment*

In §4.1, we settled on generation of a modified colour gradient such that the resulting Di Zenzo matrix just equals that for the input greyscale image. Given that prescription, can we phrase the gradient modification problem as an optimization, such that we arrive not just at a heuristic result but at the best result given the inputs: greyscale, plus putative colorized image.

Here, by "best" we mean that two conditions should obtain: (a) For a generated $\nabla \boldsymbol{\rho}$, $\boldsymbol{Z}$ should equal $\boldsymbol{Z}_g$; and (b) the output colour gradient $\nabla \boldsymbol{\rho}$ should approximate as closely as possible the putative colorized gradient $\widetilde{\nabla \boldsymbol{\rho}}$.

A solution obeying (a) is to keep only within the span of colour gradient $\widetilde{\nabla \boldsymbol{\rho}}$, and seek a 2×2 matrix transform $\boldsymbol{A}$ such that

$$\nabla \boldsymbol{\rho} \;=\; \boldsymbol{A}\, \widetilde{\nabla \boldsymbol{\rho}} \tag{16}$$

Therefore the desired relation between Di Zenzo matrices is as follows:

$$\boldsymbol{A}\, \widetilde{\boldsymbol{Z}}\, \boldsymbol{A}^T \;=\; \boldsymbol{Z}_g \tag{17}$$

Given this relation, we satisfy (a) above provided matrix $\boldsymbol{A}$ is any solution of (17). For example, one solution is given by

$$\boldsymbol{A}\,\sqrt{\widetilde{\boldsymbol{Z}}} \;=\; \sqrt{\boldsymbol{Z}_{\,g}} \tag{18}$$

where the matrix square root is the unique symmetric root [22] of the real positive semi-definite symmetric arguments above.

The complete set of solutions solving (17) then consists of all matrices $\boldsymbol{A}$ that are an orthogonal transform $\boldsymbol{R}$ away from (18):

$$\begin{aligned} &\boldsymbol{A}\,\sqrt{\widetilde{\boldsymbol{Z}}}\,\boldsymbol{R}^{\,T} \;=\; \sqrt{\boldsymbol{Z}_{\,g}} \\ &\text{or, solving,}\quad \boldsymbol{A} \;=\; \sqrt{\boldsymbol{Z}_{\,g}}\,\boldsymbol{R}\,\left(\sqrt{\widetilde{\boldsymbol{Z}}}\right)^{-1} \end{aligned} \tag{19}$$

Now, we also wish to fulfill constraint (b), that the adjusted gradient $\boldsymbol{\nabla\rho}$ approximates as closely as possible the colorized input $\widetilde{\boldsymbol{\nabla\rho}}$. This implies a constraint on rotation $\boldsymbol{R}$ as follows:

$$\begin{aligned} &\widetilde{\boldsymbol{\nabla\rho}} \;\simeq\; \boldsymbol{\nabla\rho} \\ &\Rightarrow \boldsymbol{A} \;\simeq\; \boldsymbol{I}_{\,2} \\ &\Rightarrow \sqrt{\boldsymbol{Z}_{\,g}}\,\boldsymbol{R} \simeq \sqrt{\widetilde{\boldsymbol{Z}}} \end{aligned} \tag{20}$$

with $\boldsymbol{I}_{\,2}$ the 2×2 identity matrix.

Now let us decompose the product of square roots of colorized structure tensor and greyscale tensor,

$$\left(\sqrt{\widetilde{\boldsymbol{Z}}}\right)^{T}\,\sqrt{\boldsymbol{Z}_{\,g}} \;=\; \boldsymbol{D}\,\boldsymbol{\Gamma}\,\boldsymbol{E} \tag{21}$$

with $\boldsymbol{\Gamma}$ diagonal (the transpose above is actually unnecessary since $\widetilde{\boldsymbol{Z}}$ is symmetric). Then the solution $\boldsymbol{R}$ to the Orthogonal Procrustes problem in the last line of (20) is given [22] in terms of Least Squares by

$$\boldsymbol{R} \;=\; \boldsymbol{E}\,\boldsymbol{D}^{\,T} \tag{22}$$

The best least squares result for matrix $\boldsymbol{A}$ in eq. (19) thus results from using the above rotation $\boldsymbol{R}$, and we determine a modified colour gradient $\boldsymbol{\nabla\rho}$ from (16). To re-integrate the colour gradient thus determined, we go to a colour-opponent space and separately re-integrate each chroma channel, setting the constant of integration as in the Heuristic method. Again, for the luminance channel we use the original, grey input.

In terms of idempotency, we have the same situation as for the Heuristic method in that, although iterating the algorithm gives back the same colour gradient, because we re-integrate the gradient into an output colour image the method is not idempotent. And again, iteration of the method progressively desaturates the resulting output.

Fig. 1(f) illustrates that the output for this method based on a Least Squares solution is quite close to (but not the same as) the output for the Heuristic method in Fig. 1(d). Indeed, comparing to the user-supplied hints in Fig. 1(b), the output of the Least Squares based method is seen to adhere more closely to the desired colours, while yet maintaining a much more realistic look than the initial colorized image Fig. 1(c) generated from the method in [1]. In general, while the heuristic-driven approach served very well to examine the main

Fig. 1. (a): Input grey image; (b): Hints; (c): Colorized by method of Levin et al.; (d): Colorization adjusted by proposed Heuristic method; (e): Output after three iterations of Heuristic method; (f): Output of Least Squares method; (g): Fig. (b) colorized by method of Sapiro (straightforward Jacobi iteration implementation with Dirichelet boundary conditions); (h): Fig. (g) adjusted by Heuristic method; (i): Fig. (g) adjusted by Least Squares method; (j): Fig. (b) colorized[1] by method of Sapiro and Yatziv (k): Fig. (j) adjusted by Heuristic method; (l): Fig. (j) adjusted by Least Squares method.

ideas set out here, we found that the optimization-based approach usually produces output very similar to the heuristic-driven method, but is a simpler algorithm and generates equal or better colorizations.

[a]Since supplied Java applet [15] is interactive, colour hints used to generate Fig. 1(j) are the same colours as in Fig. 1(b) but only approximately in the same locations.

Fig. 2. Simply desaturating works quite haphazardly and also typically we can see the colour-hints too well.

5.2. *Results for Heuristic and Least Squares Methods*

Fig. 3 compares results using the method of [1] with both the Heuristic method of §4.1 and the Least Squares based method on §5.1.

Note that results shown here are meant to *illustrate the beneficial effect of the projection algorithm*, not to generate the best colorization that could possibly be created. Again, we apply minimal colour hints in order to examine the effect of the realistic-gradient methods.

Fig. 3(c,d) shows that the isolated orange hints in (b) appear almost *neon* in the initial colorization (c) given by [1], but acceptable in the more realistic projected outputs (d,e) for the two proposed methods. The same effect is seen in (f-j). In the third set, the colorized photo in (m) has an unacceptable neon effect on the red beads and a highly visible colour artifact on the tablecloth, and these problems are greatly mitigated in (n,o), with (o), the result of the Least Squares approach, yielding closer colours to the user-supplied hint colours. The output in (r) appears lurid (if impressive), whereas the versions in (s,t) are indeed more realistic and delicate. Differences do exist between the Heuristic based method and the Least Squares based one, although they are subtle. Zooming in on (i) and (j) reveals that, especially around the eyes, the Least Squares based colorization is more realistic. Fig. 4 shows the difference image for the two methods. In that the second approach is also faster, we consider that it forms the best solution. For the Heuristic method we found a mean execution time of 67.60 sec for 480×640 images using unoptimized Matlab, versus 3.81 sec for the Least Squares method, on a dual-core 64-bit 2.53Ghz machine. The timing difference is due to the fact that the Heuristic method must take the time to examine signs and assign the correct eigenvector, while the Least Squares method can simply proceed in a vectorized form which is much faster in a vector-based environment. Certainly the latter timing indicates that our proposed projection schema would be more time-saving for a user than a round of adding new colour hints.

6. Simplified Colorization

We have been concerned in this paper with what can be accomplished using only a small set of user-supplied colour hints. Since the proposed realistic colorization algorithm is useful for producing usable colorizations out of unacceptable ones, we might rely further on the projection algorithm by investigating what can be accomplished using a very simplistic and fast colorization in the first place. The objective here is not to generate a fast colorization

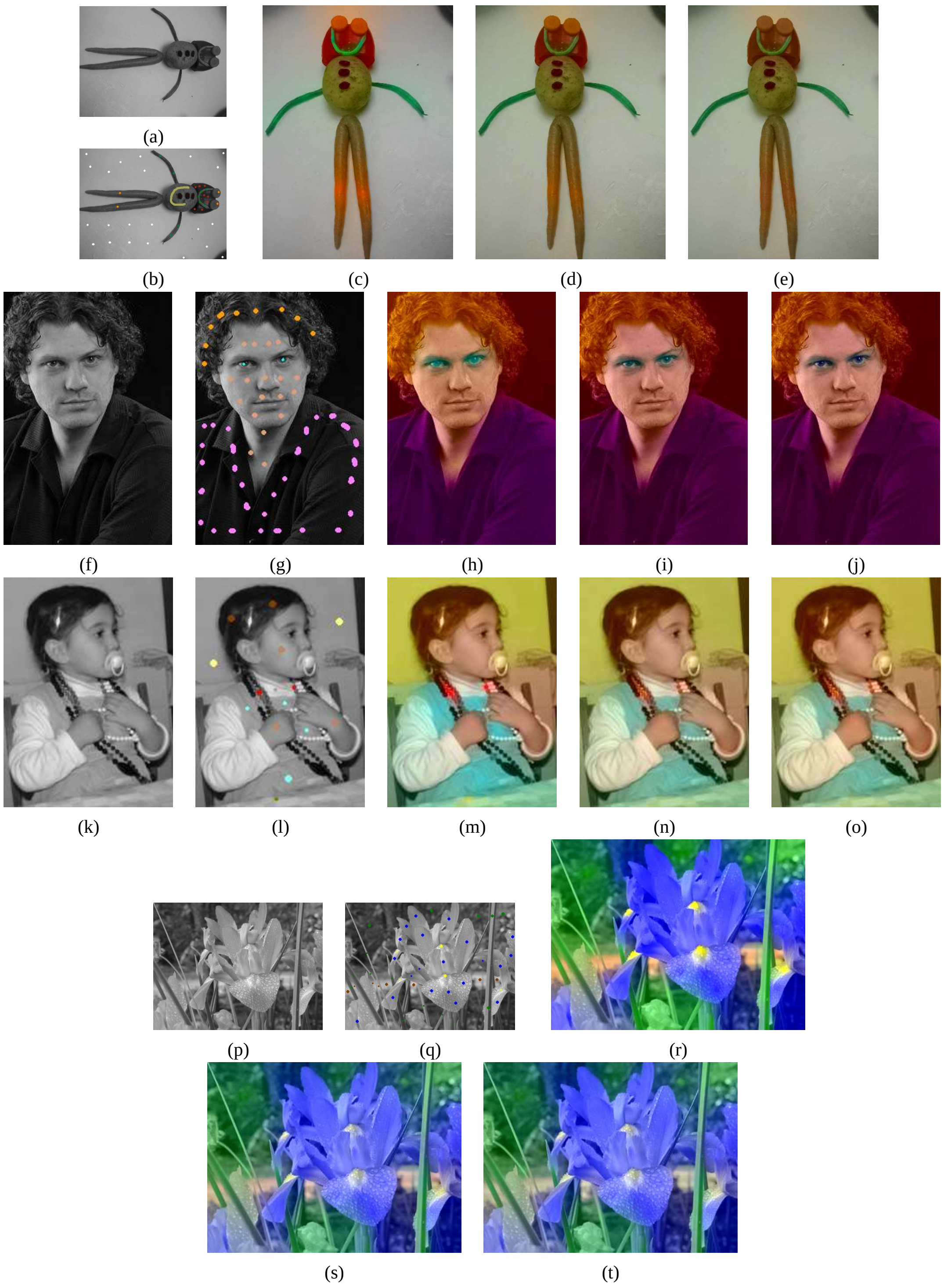

Fig. 3. Colorization using structure tensor gives more realistic output colour images: (a,f,k,p): input grey image; (b,g,l,q): user-supplied hints; (c,h,m,r): colorized image; (d,i,n,s): realistic colorized based on Heuristic approach; (e,j,o,t): realistic colorized based on Least Squares.

Fig. 4. Difference image between realistic colorizations using Heuristic method and Least Squares method.

(although this does take place, albeit crudely), but simply to further investigate the effect of the realistic projection proposed in this paper on colorized images.

Suppose we again ask the user to introduce a few colour hints into the grey image. But instead of utilizing a full colorization algorithm, we simply spread the chrominance of the colour hints to neighbouring pixels, as in [4]. But to make a faster algorithm (and, incidentally, a more challenging task for the realistic colour projection), we do not even apply information about the greyscale gradient, as in [4], and certainly not the relatively complex graph-based greyscale distance used there. Instead, we simply consider Euclidean (x, y)-distance of a pixel from the colour hint. This way, we do expect a very fast but quite unacceptable colorization — which the projection corrects, to a degree. The idea is to examine the beneficial effect of the proposed algorithm, even on a simplistic colorization scheme.

That is, we simply apply a labelling scheme to a binary image of locations of colour hint blobs, with label 0 for the background, and labels $1..k$ for the k user-supplied hint areas. Here we find connected components using 8-connected pixels. Thus for each colour blob we have a distance d from each pixel to the closest pixel on the colour blob. We normalize via $d \to (1 - d/max(d))^p$, localizing the effect of each colour blob to its close surround by raising to the power p (typically $p = 15$). Then for each chrominance channel, we add the chrominance for the current colour blob to all pixels, modulated by the distance d. I.e., we simply spread the colour by weighting with a function of distance from that colour hint. For the grey channel, we use the original grey.

Clearly, such a simple algorithm, while fast, will not result in very good results, but it is nonetheless interesting to see what effect the realistic-projection method set out here has on the colorization.

Fig. 5 shows that applying the suggested simplified colorization algorithm produces quite unacceptable results (leftmost image). However, projection, using the proposed Least

Fig. 5. Simplified colorization using gives unacceptable results. However using proposed realistic colorization projection produces passable results upon iteration of the projection algorithm.

Squares based realistic colorization algorithm, produces results that are fairly reasonable, and improve if we iterate the projection. Since the simplified algorithm is so fast, such a strategy relying on iterated realistic projection might indeed be acceptable enough for a simple colorization.

7. Comparison of Results for Three Methods

Since colour hints are simply made up by the user, it is not straightforward to evaluate colorization results except by visual inspection. However, if we start with an actual colour image, we can take the original colours in the image as our hints, in the colour-hint locations (cf. [1]).

As an example, we again make use of the same image, but now use the original colour image that was in fact the source of a greyscale image: Fig. 6(a) shows the original image, and (b) shows the colour hints as derived from the original colour image — here, our blobs consist of constant colour given by the mean colour in each hint area. Image (c) shows the result for an initial colorization using the method in [1], whereas images (d,e) show the results of the two realistic projection algorithms presented here. All output images (c,d,e) are are quite close to each other and yield comparable correlation coefficient values compared to the input colour image: *viz.* 0.9925, 0.9924, and 0.9925 respectively for Figs. 6(c,d,e). For a colorized version of Fig. 3(k), here comparing against a fully colorized version using a good many user hints (given in [1], not shown), correlation values are 0.9612, 0.9639, and 0.9641 for a colorization using on the one hand [1] and just the few hints in Fig. 3(l) and

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

Fig. 6. (a): Input colour image for devising a grey image; (b): Hints – each is the mean *actual colour* in the hint areas defined by Fig. 3(g); (c): Colorized by method in Levin et al.; (d): Colorized by proposed Heuristic method; (e): Colorized by proposed Least Squares method.

on the other hand the two projection algorithms. Note that the number of colour hints used here is substantially smaller than for the full colorization indicated in [1] — the authors discuss adding "a few red pixels on the beads", a laborious process which is the opposite of the type of minimalist colorization task we examine here.

From results such as these we conclude that, given *correct* colour hints that agree with the input colour image, the methods all give good agreement with the original and it is difficult to see the effect of the projection. However, when we start with the very simplistic colorization suggested in §6, the effect of the projection can easily be seen. Fig. 7(a) shows that an initial colorization using the simplistic method produces a quite unacceptable result. However, if we then project, the result in Fig. 7(b) is changed substantially. Comparing to

(a) (b)

(c) (d) (e)

Fig. 7. (a,d): Simplistic method for colorizing – unacceptable; (c): Colour hints; (b,e): Projected simple output is reasonably acceptable.

the original colour image, the initial correlation is 0.9420, whereas for the projected image it is 0.9885, a substantial improvement (but not as close as for a non-simplistic colorization, of course). Fig. 7(d,e) show the result of the simplistic colorization using mean actual colours for locations shown in Fig. 7(c) (boosted to maximum brightness by dividing by the maximum over all RGB values.). Again, the effect of projection on the colorization is substantial, leading to an increase of correlation from 0.8747 to 0.9631. Other images produced very similar results. Whereas the initial simplistic colorization is quite blurred, of course, albeit very quick to produce, the projected version is considerably closer to a target colorized image.

8. Conclusions

We have presented an algorithm for adjusting the colour gradients of a colorized image so as to generate the same maximum-contrast direction as in the original greyscale input. The resulting re-integrated colour image therefore has colour-contrast that appears the same as the luminance-contrast of the original, and hence appears more realistic.

Two embodiments of the main idea were presented. The first, which serves to motivate the discussion, is a reasonable heuristic approach that drives the Di Zenzo matrix for the colorized image into agreement with that for the input, greyscale image. The second, an optimization approach, casts the equivalency into an Orthogonal Procrustes problem, which has a unique solution. The optimization-based algorithm is simpler, and can generate better results.

We show the effect of the gradient-adjustment approach presented by looking at a very simplistic colorization scheme, which clearly produces poor results. Nevertheless, when projected into colour gradients which better capture the contrast in the input greyscale image, even a simplistic colorization algorithm can produce an adequate first pass for colorization, which could at the minimum form the basis for a subsequent re-colorization step.

References

1. A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.
2. G. Sapiro, "Inpainting the colors," in *ICIP 2005*, 2005, pp. II 698–7011.
3. R. Irony, D. Cohen-Or, , and D. Lischinski, "Colorization by example," in *Eurographics Symp. on Rendering*, 2005, pp. 201–210.
4. L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Im. Proc.*, vol. 15, pp. 1120–1129, 2006.
5. R. Szeliski, "Locally adapted hierarchical basis preconditioning," *ACM Trans. Graph.*, vol. 25, no. 5, pp. 1135–1143, 2006.
6. O. D. Cedeño, M. Rivera, and P. Mayorga, "Computing the alpha-channel with probabilistic segmentation for image colorization," in *ICCV07: Int. Conf. on Computer Vision*, 2007.
7. X. Liu, J. Liu, and Z. Feng, "Colorization using segmentation with random walk," in *CAIP09: Int. Conf. on Computer Analysis of Images and Patterns*. Springer-Verlag, 2009, pp. 468–475.
8. M. Fornasier and R. March, "Restoration of color images by vector valued bv functions and variational calculus," *SIAM J. Appl. Math.*, vol. 68, no. 2, pp. 437–460, 2007.
9. S. Kang and R. March, "Variational models for image colorization via chromaticity and brightness decomposition," *IEEE Trans. on Image Processing*, vol. 16, no. 9, pp. 2251–2261, 2007.
10. O. Lezoray, V. Ta, and A. El Moataz, "Nonlocal graph regularization for image colorization," in *Int. Conf. on Patt. Rec.*, 2008, pp. 1–4.
11. A. Buades, B. Coll, J. Lisani, and C. Sbert, "Conditional image diffusion," *Inverse Problems and Imaging*, vol. 1, no. 4, pp. 593–608, 2007.
12. M. Quang and S. Le, "Image and video colorization using vector-values reproducing kernel Hilbert spaces," *J. Mat. Imaging Vis.*, vol. 37, pp. 49–65, 2010.
13. A. Levin. [Online]. Available: *http://www.cs.huji.ac.il/ yweiss/Colorization/colorization.zip*
14. *J.-F. Lalonde and A. Efros, "Using color compatibility for assessing image realism," in* ICCV 2007*, 2007.*
15. *L. Yatziv. [Online]. Available: http://mountains.ece.umn.edu/ liron/colorization/applet.htm*
16. *S. D. Zenzo, "A note on the gradient of a multi-image,"* Comp. Vision, Graphics, and Image Proc.*, vol. 33, pp. 116–125, 1986.*
17. *G. Sapiro and D. Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering,"* IEEE Trans. Im. Proc.*, vol. 15, pp. 1582–1586, 1996.*
18. *D. Socolinsky and L. Wolff, "Multispectral image visualization through first-order fusion,"* IEEE Trans. Im. Proc.*, vol. 11, pp. 923–931, 2002.*

19. *R. T. Frankot and R. Chellappa, "A method for enforcing integrability in shape from shading algorithms,"* IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 10, pp. 439–451, 1988.*
20. *M. Drew and G. Finlayson, "Realistic colorization via the structure tensor," in* Int. Conf. on Image Proc. *IEEE, 2008.*
21. *B. Wandell,* Foundations of Vision: Behavior, Neuroscience and Computation. *Sinauer Assoc., 1995.*
22. *G. Golub and C. van Loan,* Matrix Computations. *John Hopkins U. Press, 1983.*

Photos and Biographies

Mark S. Drew is a Professor in the School of Computing Science at Simon Fraser University in Vancouver, Canada. His background education is in Engineering Science, Mathematics, and Physics. His interests lie in the fields of image processing, color, computer vision, computer graphics, multimedia, and visualization. He has published over 150 refereed papers in journals and conference proceedings. Dr. Drew is the holder of a U.S. patent in digital color processing, a U.S. patent application in color computer vision, two U.K. and international patent applications in digital image processing, and two U.S. and U.K. patent applications in image reconstruction and image accenting. He is the co-author of the book "Fundamentals of Multimedia" published by Prentice-Hall in 2004.

Graham D. Finalyson gained his PhD from Simon Fraser University in 1995 and was appointed to a Professorship in Computing Sciences at the University of East Anglia in 1999. He has to date published over 200 articles and conference papers. In 2003 Finlayson was awarded the Philip Leverhulme prize for science. Finlayson was granted a Royal Society-Wolfson Merit Award in recognition of his research contribution to the fields of colour in computer vision and colour imaging in October 2008. And in November 2009, he received the Davies medal from the Royal Photographic Society for his contribution to digital imaging.