Computational Aesthetics in Graphics, Visualization, and Imaging (2010)
O. Deussen and P. Jepp (Editors)

A robust and universal gradient domain imaging solver using gradient variables and locally varying metrics

László Neumann[1,2] and Ramón Hegedüs[1]

[1]Computer Vision and Robotics Group, University of Girona, Spain
[2]ICREA, Barcelona, Spain

Abstract

Gradient Domain Imaging (GDI) has gained a high importance and provoked numerous powerful applications over the last decade. It employs a workflow of creating an inconsistent gradient field (GF) from one or more images using different non-linear operations and finally it determines an image with a consistent, integrable GF that falls near to the prescribed inconsistent one. However, the result is not really predictable, often suffers from halo-effects and other local distortions at higher frequencies as well as from uncontrollable far-effects arising from local gradient-contradictions. The unfolding of these artifacts culminates in an undesired overall image appearance. None of the common GDI solvers can overcome these side-effects as they utilize the same local isotropic 'coefficient-pattern' in a sparse matrix description and they differ only in the numerical solution techniques.
We present a powerful GDI method solving the problem completely in the gradient domain with gradient-variables and using spatially varying metrics that depends only on the starting inconsistent gradient field. After obtaining the nearest consistent gradient field with the pre-defined metrics we return into the image space by double integration that yields the wanted pixel intensity values. Our method delivers a great aesthetic enhancement by eliminating halo effects and saving small details, furthermore providing a realistic and pleasant overall light distribution at lower frequencies. By significantly extending the range of allowed inconsistency in the prescribed gradient field, it also allows for solving a large class of problems that proved hopeless beforehand.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1. Introduction

Gradient Domain Imaging (GDI) has been an emerging field of digital imagery with growing importance and numerous practical applications in the last decade. It grants new possibilities for image manipulation that cannot be solved in frequency or spatial domain, nor with wavelet or multiresolution processing. Its development looks to be definitely interrelated with digital photography and the simultaneously emerging computational photography.

Let us recall some important and interesting applications of GDI without fulfilling an exhaustive survey. For high dynamics range imaging [FAT02] suggested a multi-resolution construction of a modified gradient field (GF). The image of reduced dynamic range was computed by solving a PDE with the Poisson solver resulting in an image, which has a GF near to the prescribed non-conservative one. We call a GF 'non-consistent' throughout this paper, which cannot be obtained as the GF of any image. [PGB03] and [SJTS04] proposed different versions of seamless editing and matting of images using the Poisson solver with appropriate boundary conditions. Another way for cloning and replacing image parts is suggested by [GEO04] through a modified gradient concept. Numerous interactive methods are available in GD for image editing, e.g. real-time painting [GWL*08], matting or creating photomontages [ADA*04]. Seamless image stitching or mosaicing [LZPW04], [EUS06], or compositing an image set into a unified or panoramic picture [AGA07] is another widely used field of GDI methods. The first gradient-based color-to-grayscale conversion method (Color2Gray) was suggested by [GOTG05], which often leads to strongly inconsistent gradient fields. Another gra-

DOI: 10.2312/COMPAESTH/COMPAESTH10/017-024

dient domain method to compute the plausible grayscale gradients can be found in [NCN07], which is based on the Coloroid system. From among the numerous GDI applications we can mention different kinds of image fusion, exposure fusion, or stylized, non-photorealistic image manipulation techniques. For example: context enhancement in [RIJ04], an interesting multi-flash based depth edge detection and its application for stylized rendering [RKF*04], photography artifacts removal in [ARN*05], edge suppression in [ARC06]. The list is incomplete and is quickly growing by time.

All the above methods employ basically the same mathematical approach to obtain the final image from the prescribed (manipulated or composed) gradient field: the overall quadratic error between the prescribed gradients and the unknown gradients of the resulting image is minimized. They apply everywhere the same weighting, which leads to a spatially invariant isotropic Laplacian kernel in the equation system. The common solvers are the PDE based Poisson multigrid solver, the FFT-based Poisson solver [HOC65] and different iterative techniques, like the conjugate gradient method with adaptive preconditioning [SZE06]. All of these techniques can also be realized on GPU.

Only a couple of papers mention varying weight factors associated with gradient components and therefore they cannot be solved using the Poisson solver. A framework of this method family is described in [ARC06] that was applied for surface reconstruction from gradient fields, based on calibrated photometric stereo. It works with spatial-domain variables, however, it also analyzes the main features of the gradient domain, by appyling graph-theoretical methods like minimal spanning-tree for selecting ignorable gradient elements in a binary weighting approach. Their so-called 'M-estimator' and 'Diffusion' methods assign continous weights to spatial-domain variables, similarly to our approach presented here except that we strictly confine to gradient variables. Beyond this significant difference, the weights in [ARC06] *depend on the residual error during iteration*, that is in an unpredictable way. A similar weighted technique is the 'gradient importance' method [NC06], where the motivation was HDRI whitout halo artifacts. Furthermore, [NC06] prescribes lower-upper constraints for the variables according to the allowed contrast of the display device. This permits the control of the otherwise not predictable *contrast range in the GDI*.

In this paper we will follow a fully different approach. Instead of seeking the solution in the image domain, an other straightforward mathematical possibility is working with gradient variables and eliminating the inconsistency of the GF which can be directly integrated to obtain the final image. This basic idea was first suggested by Attila Neumann in 2004 [NEU04], which was the basis of further developments which made us understand some features of the GF and resulted in a very simple iterative orthogonal projection and over-projection technique utilizing gradient variables. The variations of this method employing solely equal weights were presented in [NN05] and its application for the Color2Gray problem in [NCN07].

The remainder of this paper is organized as follows. In Section 2 we shortly describe the basics and interestingly contradictory nature of GDI and the qualitative character of the solution using gradient-variables and spatially varying metrics. Section 3 is the central part of our paper dealing with the relation of and switching between image and gradient domains. Here we present two different solvers using gradient-variables and locally varying metrics leading to the same solution. The first is a generalized orthogonal projection method, while the second one utilizes a closed-form sparse linear equation system with a new type of auxiliary variables introduced here, being neither luminance nor gradient ones. Section 4 presents the capabilities of the new method on problems, which involve the prescription of strongly inconsistent GFs. In Section 5 and 6 we present our conclusions and plans for future works, respectively.

2. The Nature of Gradient Domain and Gradient Variables

The problem of GDI begins with a prescribed ill-posed gradient field which cannot be obtained as the GF of any image. We need to find a consistent GF what is near to the prescribed one. The consistent GF has to preserve all salient features and intended image contents, while it should be free of artifacts, like the halo-effect. However, our efforts for solving this problem is inherently burden with *a theoretical contradiction*, due to the fact that the starting, non-integrable GF does not represent an image. For this reason it is difficult to establish what sort of salient features and image characteristics are conveyed in the prescribed GF and in what measure they are preserved in the final image with its corrected GF. The resulting consistent gradient field is only an approximation of the original inconsistent one, while it is impossible to determine a visual difference between the two as the corrected GF unambiguously corresponds to an image while the incosistent GF not. The difference between them is *only interpretable in the gradient domain*. Because of this contradictory nature of the problem, it is challenging to characterize the result of gradient domain imaging *a priori*, given a prescribed inconsistent gradient field. In fact, even the prediction of the contrast range and possible artifacts remains an issue.

In this paper we will show how *the nearest consistent GF* can be determined in gradient domain. We will see that the *distance or metrics definition is a key element of the GDI*. We should use spatially varying metrics in order to have more control over certain perceptual aspects and to eliminate the typical problems occurring when using the Poisson solver that is based on spatially uniform Euclidian metrics. The new metrics depend on the lengths of prescribed gradient elements and may also depend on the initial inconsistency values of the gradient-loops. In this way the proposed new method prevents the noticeable sign changes (often occurring in the case of Poisson solver) that would result otherwise in halo effects, and limits the spatial spreading out of errors by the correction of highly inconsistent loops. Furthermore, it approximately saves the relation among the lengths of gradients. Since this essential behavior of the new algorithm yields true also for any closed, non-elementary gradient loops, the realistic and feature preserving nature with the above mentioned positive effects of the new approach appears at every spatial resolution. One can define highly ill-posed local gradient-loops, of course, where none of the tricks help, but at least we can impose a limit to the spreading out of their local contradictions. In the worst case of extremely ill-posed GFs we may obtain at least a lower resolution image with an adequate appearance.

Our approach does not require the direct determination of image luminance values. Instead, it delivers the components of a consistent gradient field with an appropriately specified 'nearness' with respect to the prescribed one. The final image is then obtained by the double integral of this resulting gradient field. The new method works approximately with two-times more gradient-unknowns compared to the number of pixels, and in the same vector space where also the prescribed inconsistent GF is given. Due to the huge freedom factor of this space we cannot establish an equation system for the gradient variables directly as there are two times fewer gradient-loop equations than unknowns. However, we can determine *the nearest point in the subspace of consistent GFs to the prescribed starting one*. It can be mathematically proven that in case the concept 'nearest' means the Euclidean metrics, we obtain the same results what common solvers provide using luminance-variables directly in image domain, only in a different style.

3. Solution with Gradient Variables

3.1. Inconsistency in Gradient and Image Domain

Let the gradient field of an image **u** (of height Y and width X) be **g**, which is a consistent GF. The simplest finite difference definition of vector **g** with $Y{\cdot}(X\text{-}1)+(Y\text{-}1){\cdot}X$ dimension is:

$$g_x(j,i) = u(j,i+1) - u(j,i), j = 1,...,Y, i = 1,...,X-1$$
$$g_y(j,i) = u(j+1,i) - u(j,i), j = 1,...,Y-1, i = 1,...,X \quad (1)$$

Four neighboring pixels define an elementary gradient loop. In a real image the gradient-sums starting from the (j,i) until the $(j+1,i+1)$ pixel through any of the 2-step elementary ways (moving clockwise or counterclockwise) must be the same as the approprietaly signed sum of gradients along every closed curve in the image is zero. In a manipulated, inconsistent gradient loops these two ways yield different sums and the difference of the two sums - what we call *local inconsistency* - using the notation of (1) is:

$$E(j,i) = g_x(j,i) + g_y(j,i+1) - g_x(j+1,i) - g_y(j,i),$$
$$j = 1,...,Y-1; i = 1,...,X-1 \quad (2)$$

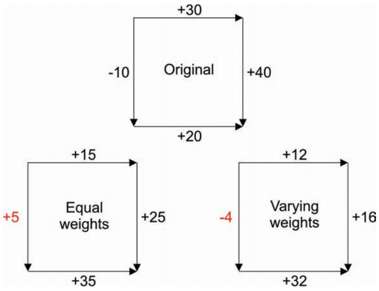

Figure 1: *(top) An originally strongly inconsistent elementary gradient loop with E=70-10=60 inconsistency (bottom left) Result after correction with equal weights (bottom right) Result after correction with varying weights. Notice the sign change caused by the equi-weighted solution (Poisson approach), while the proposed weighting avoids it and approximately preserves the relation among the lengths of gradients.*

The inconsistency vector **E** containing the $E(j,i)$ components row-wise has a dimension of $(Y\text{-}1){\cdot}(X\text{-}1)$. A GF computed from a real image retains zero local inconsistencies over the whole field. In possession of the **g** vector of a consistent GF we can simply return into the image space by a double integral (i.e. summation in our case):

$$u(j,i) = \sum_{m=1}^{j-1} g_y(m,1) + \sum_{n=1}^{i-1} g_x(i,n) + C \quad (3)$$

The double summation yields the same results regardless of it being organized row or column-wise. One operation is needed for each pixel. Similarly to direct image computing methods this problem is undefined up to a free additive constant C. Starting from an inconsistent GF the double integration gives a useless and disturbing image with typical artifacts.

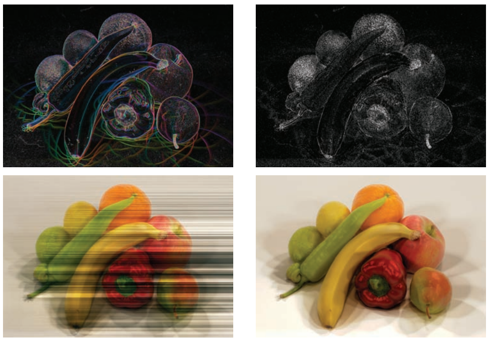

Figure 2: *(Top left) Illustrates the horizontal-vertical gradient components of an inconsistent GF (in CIELUV color space, where L = normalized gradient length, $u = g_x$, $v = g_y$) . (Top right) Pattern of local inconsistency values across the GF. Lightest parts show the highest inconsistencies, while dark parts are nearly consistent. (Bottom left) The u(j,i) image computed directly from the prescribed, inconsistent gradient field with the double integral according to Eq. (5) resulting in typical striped artifacts (Bottom right) Artifact-free image obtained from the corrected gradient field*

3.2. The orthogonal projection method

In this paper we present two methods. Both determine the nearest consistent GF to the starting inconsistent one. The first method is an iterative orthogonal projection or over-projection method, while the second one applies a closed form linear equation system with auxiliary variables. The orthogonal projection technique follows the one presented in [NCN07] but with anisotropic weighting.

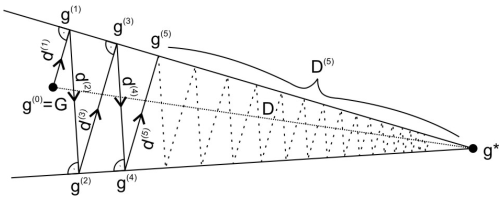

Figure 3: *2D illustration of the successive iteration steps taken with the orthogonal projection method in Eucledian metrics, converging from the inititial* $\mathbf{g}^{(\mathbf{0})}$ *point into* $\mathbf{g}^{(*)}$, *where the consistency requirement is fulfilled.*

In each step we eliminate the inconsistency of a selected elementary inconsistent gradient-loop. Four predefined weight factors multiplied by an appropriate constant d are added to each of the 4 selected gradient vector components, respectively. In this way the given elementary loop gets corrected, but at the same time the inconsistencies of its 4 neighbors also change. However, the iterative process will ensure that all elementary inconsistencies tend to zero, while the gradient vector approximates the consistent one being nearest to the starting inconsistent gradient vector (in the specified metrics).

For the sake of a unified mathematical description, in this paper the $Y{\cdot}(X\text{-}1)$ horizontal and $(Y\text{-}1){\cdot}X$ vertical gradient components of a given gradient field are put altogether into a corresponding vector with a dimension of $Y{\cdot}(X\text{-}1)+(Y\text{-}1){\cdot}X$, so that it encompasses all x-components in a row-wise sequence, followed by all y-components in a column-wise sequence:

$$\mathbf{G} = (\mathbf{G}_x, \mathbf{G}_y) = (G_x(1,1), G_x(1,2), \dots, G_x(Y, X-1), \\ G_y(1,1), G_y(2,1), \dots, G_y(Y-1, X)$$

Following this definition we call vector $\mathbf{G}$ the prescribed inconsistent gradient field. All other gradient fields denoted by vectors $\mathbf{g}^{(\mathbf{0})}$, $\mathbf{g}^{(\mathbf{1})}$, $\mathbf{g}^{(\mathbf{2})}$, ..., $\mathbf{g}^{(*)}$ throughout this paper are defined in the same way.

How the spatially varying weights can be assigned to the gradient components? In general $\mathbf{w} = (w_x, w_y) = f(G_x, G_y, E)$, where $\mathbf{w}$, $\mathbf{G}$, $\mathbf{E}$ are the weight vector, the prescribed gradient vector and the vector of *initial inconsistencies* depending on $\mathbf{G}$, respectively. As each gradient component is associated with a single weight, $\mathbf{w}$ has the same dimension and index-sequence as gradient vector $\mathbf{G}$. In our approach ***weights do not change during the iteration***, they only depend on the starting gradient field $\mathbf{G}$. We have a great freedom in how weights are exactly defined. By applying a weighting partly dependent on the initial inconsistencies we can preserve the consistent or nearly consistent areas of the prescribed gradient field. However, in this paper we confine to a simple weight definition directly based only on the $\mathbf{G}$ vector. According to our experiments the generally best selection is the simplest weighting with the absolute values of $\mathbf{G}$ vector components (see Fig.1. for demonstration). Also other monotonous functions of these weights, like simple power function of $|G(j,i)|^p$ gave acceptable results. In this way we can mostly avoid sign changes of gradient components that would lead to noticeable artifacts in the final image. Small gradients are kept small and the ratios or in very contradictory cases the signed relations of the gradient lengths are approximately preserved Furthermore, we applied a simple clipping rule that ensures positive w values:

$$w_x(j,i) = \max(|G_x(j,i), \varepsilon|), w_y(j,i) = \max(|G_y(j,i), \varepsilon|) \quad (4)$$

The selection of threshold $\varepsilon > 0$ is experimental. If ε is too small it can cause slow-down in the convergence speed. Conversely, if ε is too high, we approximate the equi-weighted case corresponding to Poisson solver. In the generalized orthogonal projection method a given elementary gradient-loop (see eq. 4) is corrected by adding constant times the generalized normal vector $\mathbf{N}(j,i)$ of the loop to the actual gradient vector $g^{(k)}$ at the kth iteration step. This normal vector contains the appropriate 4 weight factors $w_x(j,i)$,$w_y(j,i+1)$, $-w_x(j+1,i)$, $-w_y(j,i)$ according to eq.(4), and following row- and column-wise sequence as in $\mathbf{G}$:

$$\mathbf{N}(j,i) = (0, \dots, 0, w_x(j,i), 0, \dots, 0, -w_x(j+1,i), 0, \dots, 0, \\ -w_y(j,i), 0, \dots, 0, w_y(j,i+1), 0, \dots, 0 \quad (5)$$

Note that $\mathbf{N}(j,i)$ contains only 4 non-zero components. According to (1) we have $Y{\cdot}(X\text{–}1) + (Y\text{–}1){\cdot}X = 2YX - Y - X \approx 2YX$ gradient variables and $(Y\text{-}1){\cdot}(X\text{-}1)$ equations of the gradient-loops with their normal vectors. The normal vectors in (5) are evidently linearly independent for different (j,i) index-pairs. Let us see the background of an orthogonal projection with our generalized metrics. First we define a new scalar product: $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathbf{w}} = \sum_{k=1}^{M} \frac{a_k b_k}{w_k}$, where M is the dimension of $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{w}$ (weight) vectors. The vectors $\mathbf{a}$ and $\mathbf{b}$ are orthogonal through this definition, if $\langle \mathbf{a}, \mathbf{b} \rangle_w = 0$, which in the Euclidean space, however, represent an obliquely related vector-pair. The new non-Euclidean metric is: $\|\mathbf{a}\|_{\mathbf{w}} = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle_{\mathbf{w}}}$. Using this norm any vectors can be normalized. For the next algorithm let us define the unit vectors $\mathbf{N}^{\mathbf{0}}(j,i)$ that will simplify the following derivations:

$$\mathbf{N}^0(j,i) = \frac{\mathbf{N}(j,i)}{\|\mathbf{N}(j,i)\|_{\mathbf{w}}}, \text{ where}$$

$$\|\mathbf{N}(j,i)\|_{\mathbf{w}} = \sqrt{w_x(j,i) + w_y(j,i+1) + w_x(j+1,i) + w_y(j,i)}$$

according to the definition of norm in the new metrics. The $E(j,i)$ inconsistency of the (j,i) gradient-loop described in (2) with this notation is the generalized scalar-product of $\mathbf{N}(j,i)$ and $\mathbf{g}(j,i)$ vectors: $E(j,i) = \langle \mathbf{g}(j,i), \mathbf{N}(j,i) \rangle_{\mathbf{w}}$. We require this product to be zero for all (*j,i*) pairs in order to have a consistent gradient field. For this purpose we can subsitute vectors $\mathbf{N}$(*j,i*) for the normalized ones $\mathbf{N^0}(j,i)$, which makes the consistency requirement:

$$\left\langle \mathbf{g}(j,i), \mathbf{N}^0(j,i) \right\rangle_{\mathbf{w}} = 0 \tag{6}$$

In the iterative algorithm the elementary step is

$$\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + d^{(k)}(j,i)\,\mathbf{N}^0(j,i) \qquad (7),$$

where $\mathbf{g}^{(\mathbf{0})} = \mathbf{G}$. The factor (or distance) *d*(*j,i*) can be determined as:

$$d^{(k)}(j,i) = -\left\langle \mathbf{g}^{(k)}(j,i), \mathbf{N}^0(j,i) \right\rangle_{\mathbf{w}} = \\ -\frac{E^{(k)}(j,i)}{\sqrt{w_x(j,i)+w_y(j,i+1)+w_x(j+1,i)+w_y(j,i)}} \tag{8}$$

Convergence can be easily proven for the iteration process where in each step the gradient loop with the maximal $|E^{(k)}(j,i)|$ is selected, which requires the application of efficient data structures (e.g. Fibonacci-heap). Fortunately, it is easy to see that the convergence is guaranteed for any kind of iterative traversing through all elementary loops that do not necessarily require auxiliary data structure or extra storage. Furthermore, convergence is also ensured in case we apply $\omega \cdot d^{(k)}$ $(0 < \omega < 2)$ instead of *d(k)*, which corresponds to *under- or over-projections* in the gradient vector space. In different applications we found that selecting ω = 1.6...1.95 values speed up the iteration process. The equi-weighted case, i.e. $d = -1/2E$ provides an equivalent result with the Poisson solver, while the d(j,i) factors mean Euclidean distances. Let the consistent solution be $\mathbf{g}^*$, then the distance D between the starting $\mathbf{G}$ vector and $\mathbf{g}^*$ solution vector in the new norm is $D = \| \mathbf{g}^* - \mathbf{G} \|$. During the iteration we approach $\mathbf{g}^*$ successively (see Fig. 3 for a simplified illustration, where ω = 1). According to Pythagorean's theorem in the space of the new metrics the square of the actual distance from the solution decreases after each step:

$$D^2_{(k+1)} = D^2_{(k)} - \left(d^{(k)}\right)^2 \tag{9}$$

Vector $\mathbf{g}^* - \mathbf{G}$ is orthogonal to the linear subspace of consistent GFs. The length of vector $\mathbf{g}^* - \mathbf{G}$ is the shortest distance between $\mathbf{G}$ and the 'consistent subspace' by the generalized metrics. This metrics expresses our importance criteria and finally results in improved image quality.

3.3. The Closed Form Solution

In the description of the above method observe that in any iteration step when a correction of a given (j,i) gradient-loop is to be performed, it is done by spanning a distance always along the same fixed line direction $\pm\mathbf{N^0}(j,i)$ within the gradient vector space. From this it follows that for each (*j,i*) loop there exists a d(*j,i*) scalar, signed total distance factor that equals to the sum of the elementary $d^{(k)}(j,i)$distances covered through an infinite number of iterations:

$$d(j,i) = \sum_{k=1}^{\infty} d^{(k)}(j,i) \tag{10}$$

If all the distances d(*j,i*) were known, the solution vector $\mathbf{g}^*$ of the nearest consistent GF could be straightforwardly obtained by:

$$\mathbf{g}^* = \mathbf{G} + \sum_{j=1}^{Y-1} \sum_{i=1}^{X-1} d(j,i)\mathbf{N}^0(j,i) \tag{11}$$

It follows that by definition the consistency-requirement $\left\langle \mathbf{g}(j,i), \mathbf{N}^0(j,i) \right\rangle_{\mathbf{w}} = 0$ holds for each (*j,i*) gradient-loop. Now, substitute the expression (11) into these equations. According to the definition given in (5) observe that the normal vector $\mathbf{N^0}(j,i)$ of a given (*j,i*) gradient-loop is orthogonal to those of all the other loops except to the normal vectors $\mathbf{N^0}(j-1,i)$, $\mathbf{N^0}(j+1,i)$, $\mathbf{N^0}(j,i-1)$, $\mathbf{N^0}(j,i+1)$ of its direct neighbors. This allows the restriction of double summation given in (11) to the neighbors of each (*j,i*) gradient-loop when postulating the consistency-requirement, so that we obtain the equations:

$$\sum_{n=j-1}^{j+1} \sum_{m=i-1}^{i+1} d(n,m) \left\langle \mathbf{N}^0(n,m), \mathbf{N}^0(j,i) \right\rangle_{\mathbf{w}} = -\left\langle \mathbf{G}, \mathbf{N}^0(j,i) \right\rangle_{\mathbf{w}} \tag{12}$$

for each (*j,i*), *j*= *1*,...,*Y–1*; *i*=*1*,...,*X–1*, with the following constraints: $1 \le n \le Y-1$ and $1 \le m \le X-1$. In this way *d*(*j,i*) values can be directly determined as eq. (12) yields a linear equation system with unknowns *d*(*j,i*) for all (*j,i*) pairs. Each row of the equation system has 5 non-zero coefficients (except for those corresponding to border-loops that contain fewer), which build up an anisotropic Laplacian kernel depending on local weight factors. The matrix of the equation system is sparse, symmetric and positive definite. This allows that the equation system be solved using the preconditioned conjugate gradient method with low computational cost. The sought consistent gradient field $\mathbf{g}^*$ is then obtained by calculating (11) with the resulting *d*(*j,i*) values.

The vector $\mathbf{d}$ that encompasses all *d*(*j,i*) values has (*Y–1*)·(*X–1*) number of components, which is slightly less than the *Y*·*X* dimension of the luminance vector. This shows an interesting feature of the problem of finding the nearest consistent gradient field to a prescribed inconsistent one, as it can be formulated into an equation system of auxilairy variables *d*, the number of which is less than that of the pixels.

Finally we have to mention the possibility of a hybrid method that consists of a 'warming-up" phase with the orthogonal projection until a given *k* number of iteration steps and then continues with the closed-form solution. This hybrid technique delivers identical result to that produced by the above algorithms, while it may speed up the solution process.

4. Results and Applications

In this section we demonstrate how the proposed gradient solver with spatially varying weights performs in terms of image quality (elimination of artifacts, global light distribution, preservation of salient features, etc.) and how it compares to the solution using equal weights that provides identical results with the commonly used Poisson-solver. For this purpose we have chosen some simple gradient field manipulation techniques, which lead to prescribing highly inconsistent gradient fields. The applications shown in 4.2 and 4.3 are also of practical relevance, however, we strictly emphasize that these specific gradient manipulation algorithms serve only demonstration purposes in our paper and do not at all represent the only and best solution within the related subfield of image processing.

In all of the shown examples the gradient field of log Y luminance values of the image(s) were calculated and then manipulated, where Y is the luminance channel of XYZ color space. Original hue and saturation values considered in the CIELUV space were stored for all pixels and the new luminance values obtained from gradient manipulations (and after finding the nearest consistent gradient field to the initial one) were associated with these hue and saturation values to yield the final image. Spatially varying weights were set as defined in (4), while for equal weights $w_x(j,i) = w_y(j,i) = 1$, for all (j,i) pairs.

4.1. Point-error in the gradient field

The definitive difference in results produced by the gradient solvers with different weightings can be already seen by carrying out the most elementary gradient manipulation: changing only a single gradient element, i.e. introducing a point-error into the GF. Prescribing large gradients (relative to surrounding ones) are often the cause of strong local inconsistencies. We have set an arbitrarily selected g_x horizontal gradient component to +100 in the image and then sought the nearest consistent gradient field with spatially varying and equal weights, respectively. Fig. 4. shows the remarkable difference between the two solution methods.

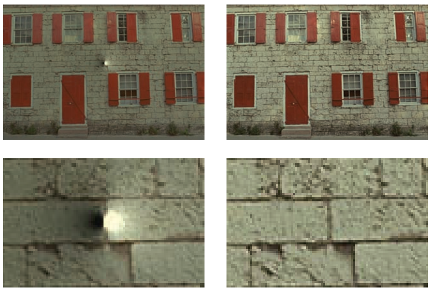

Figure 4: *Comparison of images obtained by correction of the GF containing a single point-error with (Left column) Equal weights (Poisson approach), (Right column) Spatially varying weights. Bottom row shows the magnified image parts around the location of the initial point error*

Since the commonly used equal weighting means that all elementary gradients are considered equal in contributing to inconsistency, the classical approach eliminates it by introducing substantial changes to untouched gradients, while not reducing the point-error enough and letting it visible in the final image. The result is an outspread, halo artifact around the manipulated gradient. In contrast, the solution where weights are determined by gradient lengths, i.e. large gradients are given more account for causing inconsistency, eliminates the introduced point-error completely and practically restores the original image without any artifacts. This elementary example already demonstrates well that there is a profound difference in the behavior of the two solution techniques.

4.2. Illumination Fusion: Max-Gradient Selection

There are numerous approaches and methods known for unifying differently illuminated images of the same scene from the same view-point. We can mention some powerful techniques which operate in the spatial domain, e.g. [ALK*03], [MKvR09]. Both of them applies a multi-scale workflow. As for gradient domain approaches, a simple algorithm was suggested by [PGB03]: compose a gradient field, in which the elementary gradients are set as the local maximum of all input gradients, $|G(j,i)| = \max_k |G_k(j,i)|$. In this way the fused image is intended to contain all the relevant local details, which are varyingly revealed by different illuminations throughout the image. However, these operations typically create a strongly inconsistent gradient field, which often leads to perceptually intolerable artifacts with common solvers.

Figure 5 shows the results of this illumination fusion technique applied to a series of 16 images shot of a golden bowl with variously oriented flash lights. Using the Poisson-solver (or equal weighting in our gradient-domain-based solver) clearly fails to produce a fused image with acceptable quality as it is compromised by the presence of extensive halo artifacts, especially pronounced around the upper part of the bowl. Our new gradient solver, however, proves successful by creating an artifact-free, perceptually pleasant image from the same gradient field, while preserving the maximum amount of local details as it is intended by the gradient manipulation algorithm. It is also immediately apparent that there is a significant difference between global light distribution across the images obtained by the two solvers: this is even more obviously depicted by the luminance difference image calculated from the results. The desirable global behavior of the new solver is granted by the fact that avoiding sign changes of and preserving the length relations of gradients are realized at all spatial frequencies. As a curiosity, in fig. 5. we also present the distribution of acquired **d** auxiliary variables as an image both for the equi-weighted method and for the new one. There is no space for deeper analysis here, but they nicely illustrate the difference between the two methods in style, both at high and low frequencies, accounting for the use of difference metrics.

This example clearly shows that the proposed new gradient solver is capable of rendering a gradient domain method usable, which previously unpredictably produced images that suffer from disturbing artifacts. However, we should note that prescribing the locally available maximal gradient across the image does not guarantee that the fused image

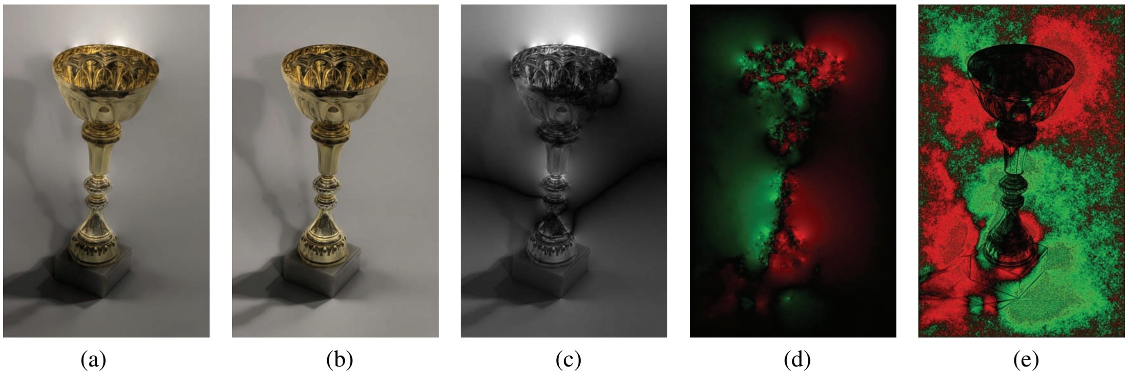

Figure 5: *GDI-based Illumination fusion: Result using (a) equal weights [Poission approach], (b) spatially varying weights; (c) Normalized absolute luminance difference of (a) and (b); Pattern of auxiliary variables d(j,i) in case of (d) Equal weights, (e) Spatially varying weights, where red/green color denotes the sign of d(j,i) and luminance encodes its magnitude*

will necessarily possess a realistic/pleasant global light distribution and also locally it may produce an unnatural look where in either of the input images there were peak lights. For this reason, a more sophisticated gradient-domain-based exposure fusion method is needed if one wishes to match or outperform state-of-the-art, non-gradient-based approaches.

4.3. Fast Solution of Gradient-Based Color2Gray

In this subsection we demonstrate the power of our new method on a known gradient domain technique for converting color images into grayscale [GOTG05]. Their basic idea to express the local changes even between equiluminant pixel-pairs with an appropriately signed color difference as a local grayscale gradient is very elegant. However, the algorithm has not been widely used due to its high demand on computation time that is related with the problem of stipulating a strongly violated gradient field. The reason for this is the highly discontinuous mapping of luminance and chrominance differences into signed gradient values in the proposed formula (there is an 'either luminance or chrominance' selection and another discontinuity by the particular way of hue-circle splitting). Partly because their algorithm produces a highly inconsistent gradient field, [GOTG05] could not apply the Poisson solver and had to suggest an other method of $O(N^4)$ complexity instead. Here we show that using our universal gradient solver good results can be achieved with a significantly lower computation time solving our sparse matrix equation system.

Figure 6 shows an example for Color2gray conversion with the classical Poisson-approach and the new one with spatially varying weights. The nearly equi-luminant geometrical figure reveals that Poisson-solver in this case is completely useless as it introduces apparent artifacts all over the image, while our new gradient solver provides a correct, visually pleasant grayscale image. In the case of a 1 Mpixel image the computation time with our gradient solver was 10 seconds on an Intel Core2Duo 2.66 Ghz CPU (using only single core). This demonstrates that using the proposed technique the Color2Gray algorithm becomes a practically usable method for images up from the Mpixel resolution range without GPU acceleration.

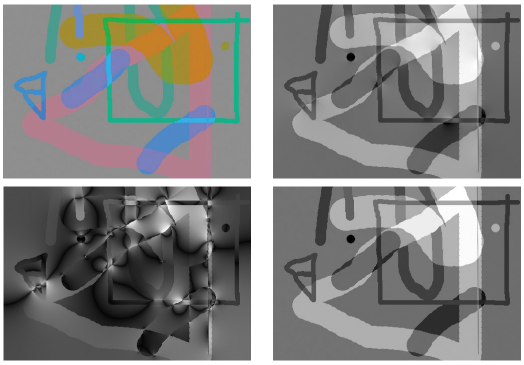

Figure 6: *Color2Gray example: (Upper left) Original color image, (Upper Right) Solution with equal weights, (Bottom Right), (Bottom Left) Normalized luminance difference between the two solutions*

Here we should note that any exclusively gradient-based Color2gray solution requires some kind of multi-scale approach to take far-effects into consideration in order to avoid particular grayscale conversion artifacts (e.g. disjointed patches of the same color may be transformed into different gray tones).

5. Conclusions

We proposed a new approach and solution technique for gradient domain imaging that solves the problem with gradient-variables. The presented iterative orthogonal projection method as well as the closed-form linear equation approach determines a consistent, i.e. directly integrable gradient field that is nearest to the starting inconsistent one in the introduced new metrics. The method is based on weighted gradients which determine the spatially varying metrics in the nearly $2 \cdot Width \cdot Height$ dimensional gradient space. The weights depend on the starting inconsistent gradient field and optionally also on the starting inconsistency values of the elementary gradient-loops. The metrics remains fixed during the iteration and does not depend on the actual residuum gradient field.

According to our experiments and mathematically based expectations we obtained better feature-preserving images with fine details at higher frequencies while eliminating disturbing halo effects as compared to the results provided by common GDI solvers. At the same time the global light distribution, that is, overall appearance at the low frequencies is also often more pleasant due to the implicit multi-scale nature of larger gradient-loops. The proposed new method opens the way for solving problems with a pre-scribed gradient field that is strongly inconsistent, which proved hopeless with the commonly used solvers. Our technique delivers a highly improved image quality for all existing gradient domain methods.

6. Future Works

In order to speed up our algorithm we plan to employ a multi-grid-style solver that includes equations of non-elementary gradient loops (e.g. the curves of quad-tree), since the sum along any closed curves must equal zero in the subspace of consistent gradient fields. Further investigations are proposed for finding the application-dependent optimal weighting strategy or other generalized metrics. It is an open question whether it is beneficial to allow changing of the metrics during the iteration process. We plan to submit these questions to a detailed study, in order to find out which techniques provide the perceptually most pleasant solution or convey the most salient local features from the inconsistent starting gradient field. Finally we need to find the fastest stop criterion which ensures that the result of double integration is an image containing under noticeable artifacts only.

Acknowledgements

László Neumann is thankful to ICREA, Barcelona, Spain. This work was partially funded through the Spanish Ministry of Education and Science(MCINN) under grant CTM2007-64751 and the EU project FP7-ICT-2009-248497.

References

[ADA*04] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker S., Colburn, A., Curless, B., Salesin, D., Cohen, M. 2004. Interactive digital photomontage. *ACM TOG* 23, 3, 294-302. 1

[AGA07] Agarwala., A. 2007. Efficient gradient-domain compositing using quadtrees. *ACM TOG* 26, 3, 94. 1

[ARN*05] Agrawal, A., Raskar, R., Nayar, S. K., Li, Y. 2005. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM TOG (SIGGRAPH'05)* 828-835. 2

[ARC06] Agrawal, A., Raskar, R., Chellappa, R. 2006. Edge suppression by gradient field transformation using cross-projection tensors. *Proc. Conf. Computer Vision and Pattern Recognition*, June 2006. 2

[ALK*03] Akers, D., Losasso, F., Klingner, J., Agrawala, M., Rick, J., Hanrahan, P. 2003 Conveying Shape and Features with Image-Based Relighting. *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* p.46, October 22-24, 2003 6

[EUS06] Eden, A., Uyttendaele, M., Szeliski, R. 2006. Seamless imagestitching of scenes with large motions and exposure differences. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* 2:2498-2505, 2006. 1

[FAT02] Fattal, R., Lischinski, D., Werman, M. 2002. Gradient domain high dynamic range compression. *ACM TOG* 21, 3, 249-256. 1

[GEO04] Georgiev, T. 2004. Photoshop healing brush: a tool for seamless cloning. *Workshop on Applications of Computer Vission (ECCV 2004)* 1-8. 1

[GOTG05] Gooch, A. A., Olsen, S.C., Tumblin, J., Gooch, B. 2005. Color2gray: Salient-preserving color removal. *ACM TOG* 24, 3 1, 7

[GWL*08] Goodnight, N., Woolley, C., Lewin, G., Luebke, D., McCann, J., Pollard, N. 2008. Real-time gradient domain painting. *ACM TOG (SIGGRAPH'08).* 1

[HOC65] Hockney, R. W. 1965. A Fast Direct Solution of Poisson's Equation Using Fourier Analysis. *Journal of the ACM (JACM), 12(1)* 95-113 2

[LZPW04] Levin, A., Zomet, A., Peleg, S., Weiss, Y. 2004. Seamless image stitching in the gradient domain. *European Conference on Computer Vision* 377-389. 1

[MKvR09] Mertens, T., Kautz, J, Van Reeth, F. 2009. Exposure Fusion: A Simple and Practical Alternative to High Dynamic Range Photography. *Computer Graphics Forum*, March 2009, 28(1), 161-171 6

[NEU04] Neumann, A. 2004. *Priv. Comm.* Nov. 2004. 2

[NN05] Neumann, L., Neumann, A. 2005. Gradient Domain Imaging *EG/ACM-SIGGRAPH workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, May 2005, Girona, Spain 2

[NC06] Neumann, L., Cadik, M. 2006. Gradient Importance based High Dynamic Range Imaging. *Seminar on Computational Aesthetics in Graphics, Visualization and Imaging*, Dagstuhl, Germany, Seminar 06221, 2006. 2

[NCN07] Neumann, L., Cadik, M., Nemcsics, A. 2007. An Efficient Perception-based Adaptive Color to Gray Transformation. *International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, June 2007, Banff, Alberta, Canada 2, 4

[PGB03] Perez, P., Gangnet, M., Blake, A. 2003. Poisson image editing. *ACM TOG* 22, 3, 313-318. 1, 6

[RIJ04] Raskar, R., Ilie, A., Yu., J. 2004. Image Fusion for Context Enhancement and Video Surrealism *Proceedings of NPAR 2004*, 85-152. 2

[RKF*04] Raskar, R., Tan, K-H., Feris R., Yu., J., Turk, M. 2004. Non-photorealistic Camera: Depth Edge Detection and Stylized Rendering using Multi-Flash Imaging. *ACM TOG (SIGGRAPH'04)*, 679-688. 2

[SJTS04] Sun J., Jia J., Tang C., Shum H. 2004. Poisson Matting. *ACM TOG* 22, 3 1

[SZE06] Szeliski, R. 2006. Locally adapted hierarchical basis preconditioning. *ACM TOG (SIGGRAPH'06)*, 1135-1143. 2