Tensor Voting for Image Correction by Global and Local Intensity Alignment

Jiaya Jia, *Member, IEEE Computer Society*, and Chi-Keung Tang, *Member, IEEE Computer Society*

Abstract—This paper presents a voting method to perform image correction by global and local intensity alignment. The key to our modeless approach is the estimation of global and local **replacement functions** by reducing the complex estimation problem to the robust 2D *tensor voting* in the corresponding voting spaces. No complicated model for replacement function (curve) is assumed. Subject to the monotonic constraint only, we vote for an optimal replacement function by propagating the curve smoothness constraint using a dense tensor field. Our method effectively infers missing curve segments and rejects image outliers. Applications using our tensor voting approach are proposed and described. The first application consists of *image mosaicking* of static scenes, where the voted replacement functions are used in our iterative registration algorithm for computing the best warping matrix. In the presence of occlusion, our replacement function can be employed to construct a visually acceptable mosaic by *detecting occlusion* which has large and piecewise constant color. Furthermore, by the simultaneous consideration of color matches and spatial constraints in the voting space, we perform *image intensity compensation* and *high contrast image correction* using our voting framework, when only two defective input images are given.

Index Terms—Image correction and recovery, color transfer, replacement functions, applications.

1 Introduction

To successfully generate a visually acceptable or seamless mosaic from a few images, many registration methods [26], [27], [22], [32] have been proposed which align images taken under a subclass of camera motions. However, in the registration process, the environment illuminance (or brightness) recorded by a moving/rotating camera is often inconsistent, even for a static scene. Exposure variation and other camera internal parameters further complicate the light recording process, causing abrupt color transition from one image to another. Image formation is related to several factors that influence the scene radiance recorded in cameras [11]:

- *Exposure variance*: In a complex environment, a moving/rotating camera can automatically adapt to different lighting conditions via controlling the shutter speed and aperture size. This automatic adaptation leads to intensity inconsistency of corresponding pixels among captured images.
- *White balance*: It is a function of digital camera which performs chromatic adjustment so that the object color remains invariant under different lighting conditions.
- *Gamma correction*: It is related to the mapping from the received analog illuminance signal to the digital pixel color produced by a photosensitive sensor. It is a nonlinear response function.
- *Vignetting*: It indicates the unintended darkening at image corners in a photographic image. Some different image formation mechanisms may account for vignetting [24]. Natural and optical mechanisms inherent in many lens designs are usually the main factors to cause vignettes. Mechanical vignetting is due to the use of improper lens attachment. The formula of natural vignetting can be written as [11]:

$$V_m = V_o \cdot \cos^4\left(\tan^{-1}\left(\frac{r}{f}\right)\right), \tag{1}$$

  where V_m is the illuminance measured by the sensor, V_o is the incoming illuminance, r is the distance of the pixel from the optical center of the image, and f is the focal length. In mosaics construction, images are overlapped. Any unnatural darkening around boundaries cannot go unnoticed.
- *Digitizer parameters*: Different digitizers may use different A/D conversion techniques which complicate the image formation process and generate nonlinear response.

When two overlapping images are aligned, in order to eliminate seams and artifacts due to the above factors, feathering [27] can be applied to blend the two images locally around the overlapping region. Alternatively, a model for the response function [28] can be estimated to correct the exposure differences only when several images are taken of a static scene.

In this paper, we propose an effective approach to address color or intensity disparity when two or more overlapping images are registered. Inspired by 2D tensor voting, we propose a modeless approach for *replacement function* estimation. Our intensity alignment robustly removes image outliers and estimates an optimal mapping function for global and local intensity replacement. The optimality is subject to the only monotonic constraint, i.e., higher exposures should always generate brighter pixel colors and vice versa. In image mosaicking, unlike the algorithm in [11], we only need the focal length of the first image to roughly eliminate natural vignetting. No other curve or simplified model is assumed. Our estimated replacement functions are integrated into an

- *The authors are with the Department of Computer Science, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: {leojia, cktang}@cs.ust.hk.*

Manuscript received 29 Aug. 2003; revised 16 Mar. 2004; accepted 10 May 2004.
Recommended for acceptance by A. Rangarajan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0254-0803.

0162-8828/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

TABLE 1
Notations Used in This Paper

	FUNCTIONS
$u(\cdot)$	response function
$f(\cdot)(global(\cdot))$	global response (replacement) function
$k(\cdot)(local(\cdot))$	local response (replacement) function
$g(\cdot)$	replacement function ($g(\cdot) = global(local(\cdot))$)
$w(\cdot)$	image warping function (with transformation matrix)
	IMAGES
I	reference image
I'	the other input image
I''	intensity/color adjusted image of I'
$\tilde{I}'$	the warped image of I', i.e., $\tilde{I}' = w(I'(x', y'))$
(x, y)	pixel location in I
(x', y')	pixel location in I'
(x'', y'')	pixel location in the warped image $\tilde{I}'$
n	sampling density in the joint image space
	IMAGE FORMATION
t	exposure ratio
V	scene illuminance
	TENSOR
$(\tilde{I}', I)$	joint image space where x and y axes are intensities in $\tilde{I}'$ and I respectively
$\mathbf{S}(d_1, d_2)$	tensor at joint image space (d_1, d_2)
λ_{min}	minimum eigenvalue (ball saliency)
λ_{max}	maximum eigenvalue
$sal(\cdot)$	tensor saliency
$\lambda_{max} - \lambda_{min}$	stick tensor saliency (curve saliency)
$\hat{e}_{max}$	eigenvector corresponding to λ_{max}

iterative scheme, where refinements in warping function and replacement function are alternately performed until convergence. Moreover, we also propose an occlusion detection method (Section 7.2) to address the inherent color inconsistency problem in the presence of occluding objects.

Another important contribution of our method is the application of image intensity compensation. Given only two defective photographic images taken continuously with different exposure time by hand-held cameras, we can successfully generate a corrected image with crisp edges and appropriate intensities (Section 7.3).

1.1 Notations in This Paper

For ease of reference, Table 1 tabulates some important notations and terminologies used in this paper. $I(x, y)$ is the pixel color in position (x, y) of an image I. The corresponding scene illuminance V, recorded by image I, satisfies

$$I = u(V), \tag{2}$$

where $u(\cdot)$ is the response function [28] which can be further decomposed as $(f \cdot k)(\cdot)$. $f(\cdot)$ is called the *global response function*, which collectively characterizes exposure, white balance, gamma correction, and digitizer parameters. Vignetting, being a position variant quantity, is addressed by the *local response function* $k(\cdot)$. Thus, *global response function* $f(\cdot)$ describes global color mapping process, that is, if two pixels receive the same amount of light in one image, they have the same color value irrespective of their positions in the image plane. On the other hand, *local response function* $k(\cdot)$ is related to pixel positions: The pixels around image borders are darker than those at the image center, even if incoming irradiance is of similar brightness. In this paper, we do not explicitly estimate $u(\cdot)$. Instead, the *replacement function* $g(\cdot)$ is our goal. $g(\cdot)$ directly measures the color difference between images. The relationship between $g(\cdot)$ and $u(\cdot)$ is analyzed in Section 4.1.

1.2 Organization of This Paper

The rest of this paper is organized as follows: Section 2 discusses and compares related work and overviews our contributions. In Section 3, we review the tensor voting algorithm. Section 4 provides an overview of our iterative scheme which alternates the estimation of registration and color replacement. In Sections 5 and 6, we describe the details of estimating the global and local response function algorithms. We present applications of our voting approach in Section 7. Finally, we discuss our method in Section 8 and conclude our paper in Section 9.

The preliminary version of this paper appears in [14], where the tensor voting approach for intensity (luminance) alignment is presented in the context of image alignment and mosaicking. In this paper, besides that more algorithmic details are explained, we extend our estimated replacement functions to the following useful applications and provide pertinent technical contributions:

- image mosaicking: for static scenes [14] and scenes with occluding objects.
- image correction: image intensity compensation and high contrast image correction.

2 RELATED WORK

There are two classes of studies related to our method: mosaic registration with exposure transition or correction across images and radiometric calibration. Related works in image mosaicking and image correction will be reviewed in their application context in Section 7.

2.1 Mosaic Registration with Exposure Correction

Many advanced registration methods have been developed in recent years. To correct the exposure difference, one approach is to blend pixel colors in overlapping area. Video mosaic [25] is a registration algorithm to estimate homography by the Levenberg-Marquardt method. To reduce visible artifacts, it blends overlapping areas by a bilinear weighting function. However, some "mottling" spots still remain under different exposures. In [27], a feather-based algorithm can be used for better blending effect, but an unnatural seam is still inevitable under different lighting conditions. Burt and Adelson [2] made use of a multiresolution spline to perform blending. Uyttendaele et al. [29] used a block-based method to compute the response function for mapping illuminances. By using averaging and interpolation functions, a more natural transition is achieved in their results. One disadvantage of these methods is that they assume the images are already correctly registered. Moreover, blending is performed by applying a transfer function at the overlapping area only. The frameworks themselves do not have global intensity correction or any optimization process for improving alignment accuracy. It is therefore difficult to adapt these methods into our proposed alternating registration and intensity correction scheme, which uses an optimization framework and offers significant advantages to be described.

In [5], Davis proposed a global consistent registration method which reduces blurry regions by extended phase correction with the assumption of orthogonal projection.

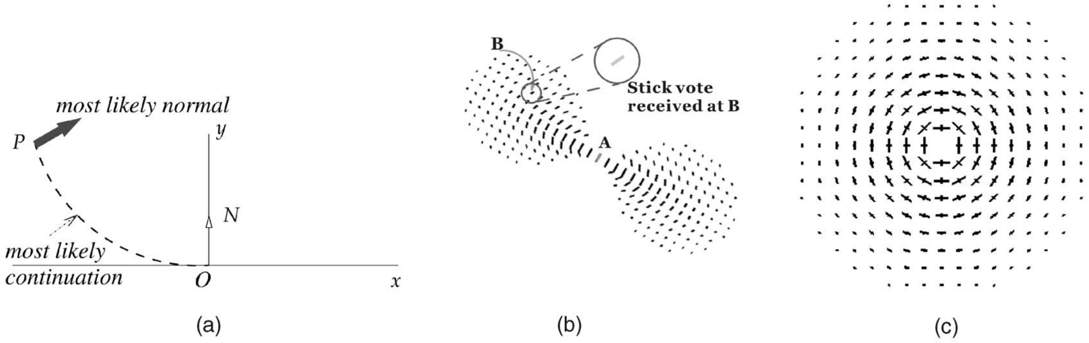


Fig. 1. (a) Design of the 2D stick voting field, (b) how a voter A casts a stick vote to vote receiver (votee) B, and (c) the 2D ball voting field, obtained by rotating and integrating the 2D stick voting field.

Our approach can also be applied to mosaicking in the presence of occluding objects while no such assumption is required. In fact, due to the variety of occlusions, phase correction does not work well in general situations.

2.2 Radiometric Calibration

The other class of approach to eliminate the intensity difference is radiometric calibration [20], [6], [17], [28], [16]. By taking several static images with different exposures, response function $u(\cdot)$ or its inverse $u^{-1}(\cdot)$ is estimated to generate high dynamic range images. In [20], $u^{-1}(\cdot)$ is modeled as a high-order polynomial $I = u^{-1}(V) = \sum_{n=0}^{N} c_n \; V^n$. Hence, calibration is viewed as determining the order N and coefficients c_n. Tsin et al. [28] adopted a noise contaminated pattern and used the maximum-likelihood method to estimate $u(\cdot)$. However, these methods are usually applied to static scenes with certain imaging or noise model assumption. Additional knowledge, such as exposure information [6], is required. Given only two unregistered images with a single overlapping area, these methods cannot be directly applied in image registration and estimation process. Our method does not explicitly infer the response function $u(\cdot)$. Instead, we formulate the problem as color/intensity mapping in order to improve the alignment between images.

In [12], radiometric calibration is performed to model the camera and eliminate senor noises. In [23], a 360 degree, high dynamic range mosaic has been produced by adding a spatially varying filter to the camera. Hasler and Susstrunk [11] combined image registration and camera internal parameters estimation into one optimization process and globally eliminate intensity difference between images. However, they assume that the parametric models, such as gamma correction and vignetting of the camera, are known. A parametric approach for the development and assessment of the accuracy of vignetting correction procedures is proposed in [7]. In [19], the generated color images are analyzed. After identifying specific features, the exposure level is adjusted according to a camera response function.

3 Review of 2D Tensor Voting

In this section, we give a concise review on 2D tensor voting [18]. Our approach consists of curve extraction and noise elimination, by associating a symmetric *tensor* representation for each input token. We employ a *voting* algorithm for "communication" among tokens: Discrete tokens lying on a *smooth* curve should mutually reinforce each other after "communication," while tokens not forming any smooth structure should not. Given two overlapping images I (reference image) and $\tilde{I}'$ (warped image), a token is a point in the *joint image space* $(\tilde{I}', I)$. Our *voting field* is a dense tensor field for postulating smooth connections among tokens, which imposes the necessary smoothness constraint for estimating a replacement function in the joint image space, to guarantee natural color transition (see Fig. 9c for tokens in the joint image space and Fig. 9d for a smooth curve example). Tensor and voting are brought together by a voting field.

In 2D structure inference, we seek to answer the following geometric question: Suppose there exists a smooth curve connecting the origin O and a point P. Suppose also that the normal $\vec{N}$ to the curve at O is known. What is the most likely normal direction at P? Fig. 1a illustrates the situation. Without a priori assumption, the osculating circle connecting O and P is chosen to be the most likely connection [18] since it minimizes the total curvature along the hypothesized circular arc. This constant curvature connection thus implicitly encodes the smoothness constraint. The most likely normal is given by the normal to the circular arc at P (thick arrow in Fig. 1a). The strength of this normal is made to decay according to the following function:

$$DF(r, \varphi, \sigma) = e^{-\left(\frac{r^2 + c\varphi^2}{\sigma^2}\right)}, \qquad (3)$$

where r is the arc length OP, φ is the curvature, c is a constant which controls the decay with high curvature, and σ controls smoothness, which also determines the neighborhood size within which tokens communicate. Thus, the proximity and smoothness constraints are encoded by the above decay function.

If we consider all locations of P in the 2D space, the resulting set of normal directions produced constitutes the 2D stick voting field, Fig. 1b. Note that σ is the only free parameter.

Let us describe the basic case: Given an input token A, how do we cast vote to another token B for inferring a smooth connection between them, assuming that A's normal is known. This is concerned with token communication and is implemented by the following voting algorithm. First, we fix σ to determine the size of the voting field. Then, we align the voting field, by translation and rotation, with A's normal, as shown in Fig. 1b. If B is within A's voting field neighborhood, B receives a stick vote $[v_x \; v_y]^T$ from the aligned field. Tensor voting is thus similar to convolution and the voting field is like a convolution mask, except that the voting result is not a scalar. Other input tokens cast votes to B as well. Second order *tensor sums* of all votes received at B are collected into a covariance matrix

$$\mathbf{S} = \begin{bmatrix} \sum v_x^2 & \sum v_x v_y \\ \sum v_y v_x & \sum v_y^2 \end{bmatrix}.$$

The corresponding eigensystem consists of two eigenvalues, $\lambda_{max} \geq \lambda_{min} \geq 0$, and two corresponding eigenvectors, $\hat{e}_{max}$ and $\hat{e}_{min}$. As a result, $\mathbf{S}$ can be rewritten as

$$\begin{aligned}\mathbf{S} &= \lambda_{max} e_{max} e_{max}^T + \lambda_{min} e_{min} e_{min}^T \\ &= (\lambda_{max} - \lambda_{min}) \hat{e}_{max} \hat{e}_{max}^T + \lambda_{min} (\hat{e}_{max} \hat{e}_{max}^T + \hat{e}_{min} \hat{e}_{min}^T).\end{aligned}$$

$\hat{e}_{max}\hat{e}_{max}^T$ is a stick tensor, with $\hat{e}_{max}$ indicating curve normal direction. $\hat{e}_{max}\hat{e}_{max}^T + \hat{e}_{min}\hat{e}_{min}^T$ is a ball tensor. $\lambda_{max} - \lambda_{min}$ is called the *curve saliency*. When $\lambda_{max} \gg \lambda_{min}$, it indicates a high likelihood that a smooth curve passes through the point.

Initially, we do not know the normal direction to A. In this case, we shall estimate it first. It is performed by voting with the 2D ball voting field, Fig. 1c. The ball voting field is obtained by rotating and integrating (by tensor sum) the vote contribution of a rotating 2D stick voting field. The 2D ball voting field is then aligned with A (along an arbitrary direction since its normal is unknown), using exactly the same voting process and collection described above. After decomposing the collected tensor into the corresponding eigensystems, the curve normal is given by $\hat{e}_{max}$ of the collected tensor $\mathbf{S}$.

In color replacement estimation, since curve normals are not known initially, we use the 2D ball voting field to estimate curve normals and curve saliencies at each discrete site in the joint image space. Afterward, the 2D stick voting field is adopted to propagate the smoothness constraint *along the tangent directions* for estimating a smooth curve. It is more effective than voting on scalar saliencies alone, where no direction information is utilized.

4 Our Approach

As reviewed in Section 1, the global and local response functions complicate the color recording process, which varies from image to image. The analysis here assumes intensity images. For color images, our algorithm operates on the three channels separately. Since we always have one image as reference, the color shifting effect is minimized in the optimization process.

4.1 Analysis

Consider two overlapping images, I and I', taken under different exposures. If I and I' are correctly registered, the warping function $w(\cdot)$ transforms the original input image to the warped one, i.e., $\tilde{I}'(x'', y'') = w(I'(x', y'))$, so that I and $\tilde{I}'$ are correctly aligned. Suppose (x, y) in I and (x'', y'') in $\tilde{I}'$ are corresponding pixels in the overlapping area of the two images. The relationship between their corresponding incoming illuminance is given by

$$V(x, y) = t\tilde{V'}(x'', y''), \tag{4}$$

where t is the exposure ratio, and it is determined by the exposure time of each image. Therefore, we have:

$$\begin{aligned} I(x, y) &= u_I(V(x, y))) \\ &= u_I(t u_{\tilde{I}'}^{-1}(\tilde{I}'(x'', y''))) \qquad (5) \\ &= g(\tilde{I}'(x'', y'')) \qquad (6) \\ &= g(w(I'(x', y'))), \qquad (7) \end{aligned}$$

where u_I and $u_{\tilde{I}'}$ may not be the same response function and $g(\cdot)$ is called the *replacement function*. Thus, the relationship between $g(\cdot)$ and $u(\cdot)$ can be represented as $g(\cdot) = u_I(t u_{\tilde{I}'}^{-1}(\cdot))$.

If I and I' are not correctly registered, we need to estimate not only the *replacement function* $g(\cdot)$ which measures the exposure and radiometric differences between images, but also the warping function $w(\cdot)$ to match corresponding pixels. Accordingly, the cost function to be minimized for image I and I' is:

$$\min \sum (I(x, y) - g(w(I'(x', y'))))^2. \tag{8}$$

The simultaneous estimation of $g(\cdot)$ and $w(\cdot)$ is a nonlinear and underconstrained optimization problem without a priori model assumption of the two sets of parameters. In our method, we estimate and refine them alternately.

Given an estimation of *replacement function* $g(\cdot)$ with respect to a certain warping function w, in order to simplify (8), we construct a new *intensity adjusted image* I'', which satisfies

$$w(I''(x', y')) = g(w(I'(x', y'))). \tag{9}$$

As a result, the intensity distance between I'' and I is shortened and (8) can be written as:

$$\min \sum (I(x, y) - w(I''(x', y')))^2, \tag{10}$$

which can be solved as a normal image registration process. We give the I'' construction process as follows:

- Since $g(w(I'(x', y'))) = g(\tilde{I}'(x'', y''))$ and $g(\cdot)$ models the intensity replacement quantity between images, we construct another warped image $\tilde{I}''(\cdot)$ which satisfies
$$\tilde{I}''(x'', y'') = g(\tilde{I}'(x'', y'')).$$
- Inversely warp $\tilde{I}''(x'', y'')$ by the same w to construct new input image:
$$I''(x', y') = w^{-1}(\tilde{I}''(x'', y'')).$$

Therefore, constructing image I'' simplifies the registration equation. Alternatively, if the warping function w has been estimated (that is, I and I' are roughly registered), we can determine the *replacement function* $g(\cdot)$ directly by tensor voting (Section 4.3).

The above analysis of (8) leads us to decouple the estimation processes of $w(\cdot)$ and $g(\cdot)$. Accordingly, we propose the refinement algorithm to alternately solve the two subproblems.

4.2 Refinement Algorithm

Initially, we assume $g(\cdot)$ is an identity mapping function to start the registration process. The algorithm then proceeds as follows:

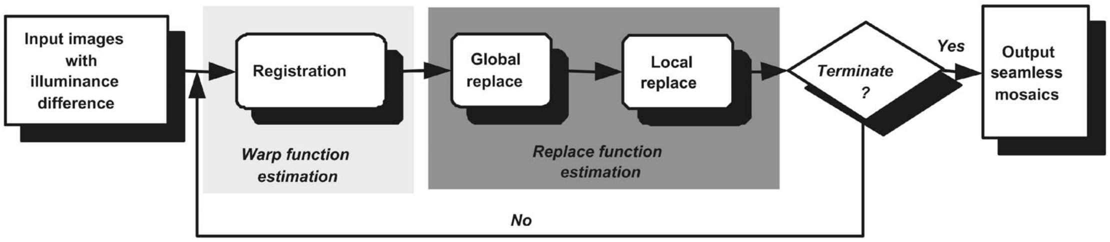


Fig. 2. Overview of our approach.

1. Fixing the estimated $g(\cdot)$ and the corresponding $w(\cdot)$, we construct a new input image I'' which adjusts color according to $g(\cdot)$, and satisfies (9).
2. Solve the registration problem in (10). Many methods can be used to tackle the normal homography estimation problem to refine the warping function $w(\cdot)$ [26], [27], [32]. In our experiment, the robust 3-parameter rotational model together with local alignment [27] is adopted, which rapidly converges to the correct solution even in the presence of some parallax.
3. Once the above registration is done, the warping matrix $w(\cdot)$ is obtained. Accordingly, we turn to solve

$$\min \sum (I(x,y) - g(\tilde{I}'(x'', y'')))^2 \tag{11}$$

   to determine the replacement function $g(\cdot)$ (Section 4.3).
4. Repeat Steps 1, 2, and 3 until the difference of $g(\cdot)$ in the last two iterations falls into the tolerance range.

The flowchart of the algorithm is shown in Fig. 2.

If the two images do not roughly align with each other after the first registration step, it will mislead the following intensity correction steps and the convergence will be seriously affected. There are several cases that make the registration step in Step 2 fail to produce a reasonable $w(\cdot)$: large intensity difference or occlusion, insufficient overlapping area, and the presence of large parallax. Among these factors, when occlusion occurs, we propose a detection algorithm to solve the convergence problem, which is depicted in Section 7.2.

4.3 Algorithm Outline for Estimating $g(\cdot)$

Since the estimation of $g(\cdot)$ is subjected to noise, uncertainties, and misalignment in overlapping area of images, Step 3 in the refinement algorithm is most crucial. We assume no model for the replacement function and make use of information of pixel colors between images only.

Just like $f(\cdot)$ and $k(\cdot)$ in the response function $u(\cdot)$, which maps from illuminance to pixel color, we decompose the replacement functions into the corresponding global and local components, say, $global(\cdot)$ and $local(\cdot)$, which map from image to image in the overlapping area. The global replacement function globally maps colors between images, while the local replacement function corrects the vignetting effect for each image.

Recall that vignettes are the gradual and slight darkening around image borders. It is less problematic to the registration accuracy than an erroneous global replacement function. For this reason, we perform global replacement function estimation first and regard any vignettes as noise or outliers in the estimation process. After that, thelocal replacement function is estimated in the input images. The final replacement function $g(\cdot)$ is constructed by compositing them, i.e.,

$$\text{(replacement function)} \;\; g(\cdot) = global(local(\cdot)). \tag{12}$$

Fig. 2 shows the $g(\cdot)$ estimation process (dark rectangle). Note that the only input value to the algorithm is the approximate focal length of the first reference image in order to roughly counteract initial vignettes in local replacement function estimation.

To apply the voting process in the image space without a model and fitting assumption, we propose a robust *tensor voting* approach to estimate $global(\cdot)$ and $local(\cdot)$, which is derived from the generic 2D tensor voting. Table 2 compares the original 2D tensor voting and intenisty voting.

5 Estimating the Global Replacement Function $global(\cdot)$ by Tensor Voting

Recall that the global replacement function maps intensity from image to image. The estimation process involves the voting space construction, tensor encoding, and the voting algorithm.

5.1 Voting Space Construction

In the overlapping area, each pixel in the warped image $\tilde{I}'(\cdot)$ has one corresponding pixel in the reference image $I(\cdot)$. Therefore, we construct the joint image, $(\tilde{I}', I)$, which maps colors between images I and $\tilde{I}'$ in the overlapping area. The left of Fig. 3 shows a typical joint image, where $n = 50$ in this illustration indicates the sampling density along the respective color axes. That is, only a subset of intensity levels $\{0, 50, 100, 150, 200, 250\} \subset [0, 255]$ is considered as voters in tensor voting to speed up processing (all other sites

TABLE 2
Comparison of the 2D Tensor Voting and This Paper

	2D tensor voting	this paper
dimension	2D	2D
token	2D point (x, y)	overlapping pixel intensity
tensor	saliency tensor	saliency tensor
space	geometry space	joint image
constraint	smooth	monotonic and smooth
voters	all neighborhood	discard column votees
feather	curve extraction	saliency comparison
scale	one (usually)	multiscale

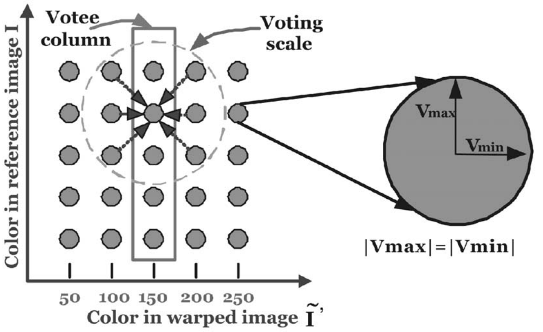


Fig. 3. Voting in the joint image space $(\tilde{I}', I)$. Left: The gray dots represent tensors which communicate with each other in the voting field (black circle). The tensors in the votee column (inside the dark box) receive votes but do not cast votes. The black arrows indicate the vote cast by voters in the neighborhood. Right: An encoded ball tensor in the joint image space.

are votees). We shall describe in Section 5.6 the typical choices of n.

5.2 Tensor Encoding

Once the joint image space is constructed as the voting space, we encode each point in the joint image into a *saliency tensor*. Similar to the original tensor voting, the saliency is defined by some λ. Initially, there is no preference of the tensor orientation in the joint image. So, we encode the point as a ball tensor in 2D space with eigenvalues $\lambda_{max} = \lambda_{min} = \lambda$ and arbitrary perpendicular eigenvectors $\hat{e}_{max}, \hat{e}_{min}$, that is,

$$\lambda\left(\hat{e}_{max}\hat{e}_{max}^T + \hat{e}_{min}\hat{e}_{min}^T\right). \tag{13}$$

This ball tensor turns out to be a circle as shown in Fig. 3.

Now, we set the saliency λ of each saliency tensor $\mathbf{S}(d_1, d_2)$ in the joint image to be proportional to the *instance number* v, where d_1 and d_2 are the joint image coordinates. v is the total number of instances for corresponding pixel pairs $(I(x,y), \tilde{I}'(x'', y''))$ in an overlapping area where $\tilde{I}'(x'', y'') = d_1$ and $I(x,y) = d_2$. Thus, our first proposal on the saliency definition is:

$$\lambda = b + v \cdot \Delta s, \tag{14}$$

where b is the base value for each tensor and Δs is the incremental scale to indicate the importance of v.

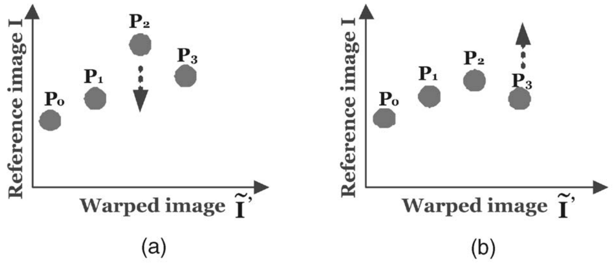
Since the local response function is estimated after the global one, we import some decay function here. Instead of using the simplified factor of $v \cdot \Delta s$ to roughly counteract the noise caused by natural vignette, we calculate the Taylor's expansion of natural vignetting given by (1) and get the vignetting factor for each pixel in image:

$$E_m(r) \approx E_o \cdot \left(1 + \frac{1}{2} \cdot \left(\frac{4}{f}\right)^2 \cdot r^2 + O(r^2)\right). \tag{15}$$

Combining the weight of Δs and the approximation of natural vignette $E_m(r)$ which replaces the fixed incremental scale in (14), we get:

$$\lambda = b + \sum_v \frac{\Delta s}{1 + \frac{1}{2} \cdot \left(\frac{4}{f}\right)^2 \cdot r^2 + O(r^2)}. \tag{16}$$

We can set f as 50 or the same as the given input reference image focal length. In our experiments, the setting of the focal length f is not critical to the robust tensor voting process.



(a)

(b)

Fig. 4. Starting from left to right along the $\tilde{I}'$ axis, if one point P_3 in the curve is lower than some of the previous points, two situations may happen: (a) Positions of the previous m points of P_3 are misestimated to be too high and (b) the position of P_3 is a wrong estimation, which is too low.

After saliency tensor encoding, each point in the joint image becomes a ball tensor with size (radius) equal to λ, as defined by (14). The following section describes our voting process for inferring the optimal global replacement function.

5.3 Tensor Voting for Color Compensation

Tensor voting infers an optimal global replacement function by collecting tensor support in the neighborhood and making use of the *monotonic constraint*. In this section, we describe the voting process and postpone the description of monotonic constraint to Section 5.4.

Color mapping is performed from $\tilde{I}'$ to I, where each value of $\tilde{I}'$ maps to exactly one color in I. Curve inference by tensor voting is performed, where the eigensystem of the inferred saliency tensors at each discrete point in the joint image is computed. The value at the site with maximum curve saliency $\lambda_{max} - \lambda_{min}$ in each votee column is chosen as the optimal curve position (the curve in Fig. 9c). Since exactly one optimal replacement color in I is selected for a given color $\tilde{I}'$ by the tensor voting process, other saliency tensors of the same color value in $\tilde{I}'$ (or in the same votee column in Fig. 3) do *not* cast vote. Instead, they are only vote receivers or votees. Referring to Fig. 3, we illustrate the vote casting directions from voters to votees in the neighborhood by black arrows.

5.4 Enforcing the Monotonic Constraint

Now, we define the monotonic constraint as follows:

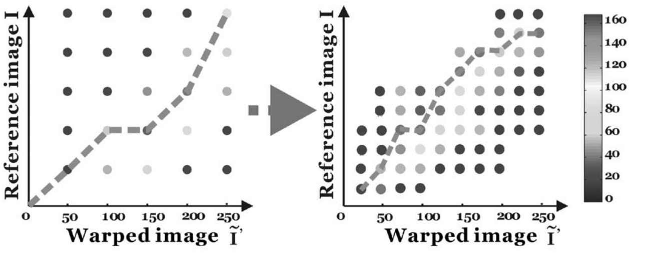
- *Monotonic constraint*: Let $(\tilde{I}', I)$ be the continuous joint image space. Given that $(x_1, y_1) \leftrightarrow (x_1'', y_1'')$ and $(x_2, y_2) \leftrightarrow (x_2'', y_2'')$ are corresponding pixel pairs in overlapping area. If $\tilde{I}'(x_1'', y_1'') > \tilde{I}'(x_2'', y_2'')$, then $I(x_1, y_1) > I(x_2, y_2)$.

In other words, the replacement curve has nonnegative gradient at each point.

To enforce the monotonic constraint in tensor voting, we propose a *local fitting algorithm* to refine the curve obtained in the previous section. Two situations to be rectified are shown in Fig. 4.

Let P_i be a point in the joint image voting space with coordinate (x_i, y_i). Recall that, for each column x_i, we select a unique optimal point P_i^* with coordinate (x_i, y_i^*), where the corresponding tensor has the maximum vote saliency among all tensors in the same column. Given two neighboring curve points P_i^* and P_{i-1}^*, if $y_i^* < y_{i-1}^*$, monotonicity is violated.

To rectify it, for each column x_i, we first sort all tensors in nonincreasing order of vote saliency and keep the sorted list as L_i. Afterward, the local fitting algorithm proceeds as



(a) (b)

Fig. 5. Global replacement estimation with multiscale voting. The gray scale of the points represents the resulting votee saliencies. The replacement curve inferred in a coarse scale space (a) is propagated to the next level in a fine scale space for inferring a more accurate curve (b). The dashed curve is the inferred replacement curve.

follows: Suppose P_i^* is lower than P_{i-1}^*, our process iterates column x_v from x_i to x_{i-m}, where m is a user-defined parameter. In each iteration:

1. Consider P_v^* in the current pass. We select the suboptimal point P_v' in the same column, i.e., the point following P_v^* in list L_v.
2. If $x_v = x_i$ and P_v' is higher than P_v^*, replace P_v^* with P_v'. If $y_i^* \geq y_{i-1}^*$, the iteration ends.
3. If $x_v < x_i$ and P_v' is between P_{v-1}^* and P_v^* in vertical position in the voting space, replace P_v^* with P_v'. If $y_i^* \geq y_{i-1}^*$, the iteration ends.

To ensure that the local fitting algorithm globally satisfies the monotonic constraint, we start from left to right along the $\tilde{I}'$ axis and, for each P_i^*, lower than P_{i-1}^*, we perform the local fitting described above until all curve points are examined.

5.5 The Algorithm

Finally, we summarize our global intensity voting algorithm as follows:

1. Tensor construction: For each (x_i, y_i), where $1 \leq i \leq \lfloor D/n \rfloor$, D is the color depth, and n is the sampling density or step size, we construct a ball tensor (with $\lambda_{max} = \lambda_{min} = \lambda$) according to (14).
2. First pass tensor voting: For each column x_i, tensor voting is performed in the whole space. All tensors in column x_i only receive votes, while others vote to them. Curve normal (denoted by $\hat{e}_{max}$) and vote saliency ($\lambda_{max} - \lambda_{min}$) are inferred at each point.
3. Second pass tensor voting: For each column x_i, tensor voting is run with the stick voting field by aligning the field with each curve normal to reinforce curve points and suppress others. Tensors in column x_i only receive votes, while all others cast votes to them. Result saliency $sal(\cdot)$ is obtained at each site. The larger sal is, the more likely the point is on the replacement curve to be inferred. On the contrary, if sal is too small, the point is likely to be an outlier or noise point to be eliminated. Hence, we regard the position where the corresponding tensor has largest saliency in column x_i as the optimal curve point P_i^*.
4. Run the local fitting algorithm.

5.6 Multiscale Voting

The voting space can be considered as a uniformly sampled saliency image. In a large overlapping area, the intensity data are dense (Fig. 9b). The size of voting fields is not required to be very large. Normally, a 9×9 field is used. Using this voting field, a typical sampling density n is 1. For highly contaminated data, we can increase the voting field size to enforce more smoothing.

In our method, in order to speed up the voting process, the multiscale voting scheme is adopted, which also helps to reduce the effect of noise due to misregistration and vignettes.

We apply the Gaussian pyramid algorithm to construct an image hierarchy. The number of pixels at level i is one quarter of the number at level $i-1$ (or, equivalently, n is two times larger). The size of the voting field differs by a factor of two in each successive level.

Suppose we vote and generate the function curves in level $i-1$ (coarse scale). Only points on the curve are propagated to level i (fine scale) for inferring a finer one. Each point in level $i-1$ has five corresponding points in next level. Therefore, in level i, we only need to vote for at most $5n_i$ points, where n_i is the sampling density in level i image. Fig. 5 shows two joint images in two consecutive scales or levels in the Gaussian pyramid.

The global replacement function corrects input images with compatible intensities. However, to better align them, we need to eliminate vignettes as well. Based only on the knowledge of focal length of the first reference image only, we apply another voting process to estimate the local replacement function.

6 Estimating the Local Replacement Function $local(\cdot)$ by Tensor Voting

The local replacement function is a position dependent mapping. A new voting space is needed to account for position changes in the saliency tensor encoding.

Since lenses and apertures are known to be mostly circular and centrosymmetric to the optical center, vignetting is a function of the distance r from the optical center, which is normally projected onto the image center. Accordingly, we partition the image into uniform concentric annuli, shown as T_i in Figs. 6a and 6b. Each annulus has the same width w, which is analogous to uniform sampling density n in the joint image space for global replacement estimation (Section 5).

Our new voting space is parameterized by (r, l), where r is the mean of the inner and outer radii for each annulus and l is the vignetting level for different annuli in the input image: Let the original (devignetted) color in the input image be C_i and the contaminated color after vignetting be C_i'. Then,

$$l = C_i - C_i'. \tag{17}$$

To estimate the unknown C_i, we first roughly devignette the first reference image by substituting the input focal length into the natural vignetting equation ((1) in [11]). Then, C_i can be regarded as the corresponding color in the reference image in the overlapping area. The process is illustrated in Fig. 6c.

Note that the two dimensions of the (r, l) space are quite different in their measurement and unit. In order to unify the metric so that points in the space can be encoded as homogeneous tensors (along the two dimensions), we should perform normalization in the space. The procedure is as follows:

1. Measure the longest distance r_{max} from a corner to the center of the image, i.e., half of the diagonal length of the image. Therefore, r_{max} indicates the resolution of the annuli.

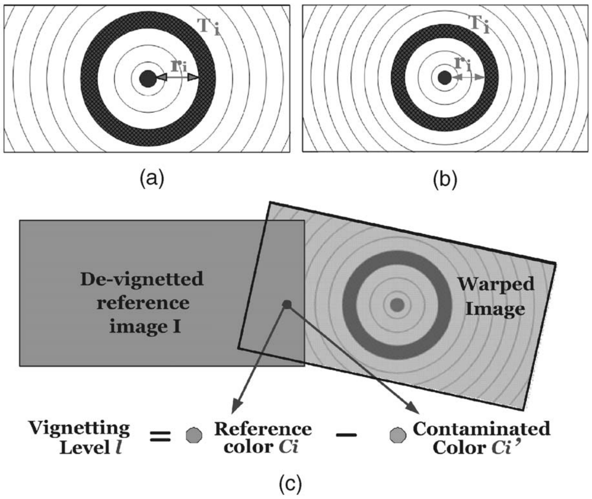

T_i
r_i
(a)
T_i
r_i
(b)
De-vignetted reference image I
Warped Image
Vignetting Level l = Reference color Ci − Contaminated Color Ci'
(c)

Fig. 6. Multiscale voting space constructed by concentric annuli for estimating local color compensation. (a) Voting space in a coarse scale. (b) Voting space in a fine scale. (c) Estimation of vignetting by measuring the color difference.

2. Compare all values of l in the overlapping area and select the largest one as l_{max}. The unit is pixel.
3. Normalize l to l': $l' = l\frac{r_{max}}{l_{max}}$ in quantity measurement.

Therefore, we normalize the 2D voting space such that the maximum values and quantization intervals for both axes are uniform. The (r, l') space is therefore analogous to the joint image space $(\tilde{I}', I)$ for global replacement curve estimation and the saliency tensors are thus encoded in the same way.

Moreover, similar monotonic constraint can also be applied here since pixels on the image border (large r) are darker than those near the image center (small r). As a result, the *same* tensor voting process can be applied to estimate local replacement function.

The same multiscale processing can be performed to speed up the inference process and enhance the outlier elimination ability, as illustrated in Figs. 6a and 6b, where the width w of annuli for different levels is scaled, analogous to the sampling density n in the global case. Given that the width w of annuli in the finest voting space is small enough, we densely sample the vignetting functions to give a good approximation for the local replacement.

Once we have obtained both global and local replacement functions, $global(\cdot)$ and $local(\cdot)$ (Section 5 and this section), we concatenate them to estimate $g(\cdot)$.

7 Applications

Our global and local replacement approach by tensor voting is applied to the following practical applications: image mosaicking and color correction. The key idea is to adapt our replacement function estimation into each application in order to vote for the functions accordingly. For RGB images, tensor voting is performed on the three channels separately. In our final result, the three channels are scaled according to the proportion of mean values between the final corrected image $g(\tilde{I}'(x'', y''))$ and reference image I to minimize the color shifting effect.

For each application, the technicalities involved will be described. Note that image mosaicking requires an iterative scheme since the warping function $w(\cdot)$ needs to be estimated alternately with $g(\cdot)$. Image correction only requires the estimation of $g(\cdot)$. Therefore, noniterative intensity voting for function estimation is sufficient.

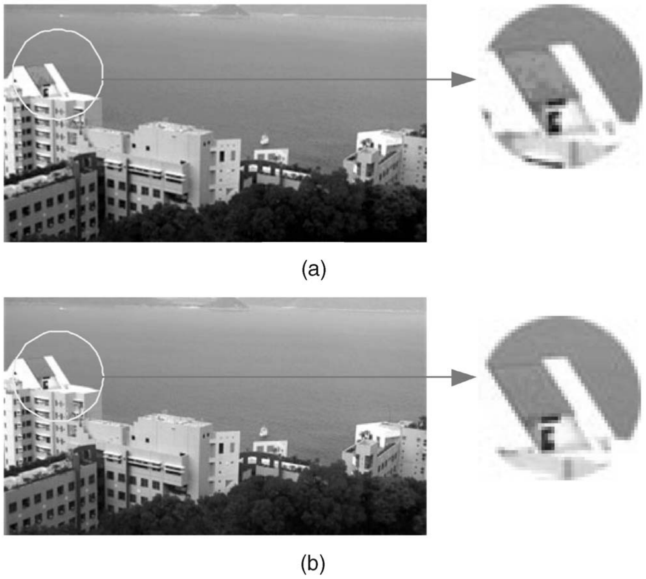


Fig. 7. (a) Fast versus (b) incremental color replacement. Incremental replacement generates a more natural image composite.

7.1 Image Mosaicking of Static Scenes

Fast versus incremental color replacement. In Section 4.2, the replacement function $g(\cdot)$ in Step 1 of the refinement algorithm directly maps color from image $\tilde{I}'$ to I, which makes the iteration rapidly terminate. We call this *fast color replacement*. Unfortunately, for image mosaicking, fast replacement introduces unnatural artifacts, as in Fig. 7a, when large exposure or a white balance difference exists. To overcome this problem, we change the replacement function from $g(\cdot)$ to *incremental color replacement* $\Delta g(\cdot)$ and modify Step 1 in the algorithm of Section 4.2 as follows.

Once we have estimated the mapping of $g(\tilde{I}')$ by tensor voting (Section 5), we calculate the incremental replacement function as

$$\Delta g(\tilde{I}') = \left(\frac{1}{\Delta j}\right) \cdot \left(g(\tilde{I}') - \tilde{I}'\right) + \tilde{I}', \tag{18}$$

where $\Delta j > 1$ is the incremental step size to control the intensity adjustment in each iteration. Then, we construct a new input image with rectified color satisfying

$$w(I''(x', y')) - \Delta g(w(I'(x', y'))) = 0. \tag{19}$$

By controlling the color increment in each iteration, noises are more effectively eliminated. Fig. 7b shows a better replacement with intervention of $\Delta j = 10$.

In our experiments, given moderate misregistration and large intensity disparity, our algorithm converges in less than 10 iterations by adopting Δg in the refinement algorithm and generates seamless mosaics. For a Pentium III 1GHz PC, the running time of tensor voting in the joint image space with the construction of Gaussian pyramid is less than 4 minutes on images with 400×400 overlapping pixels.

Fig. 8 shows the results of local replacement estimation. Initially, although the global intensity for the two images is quite close, vignette is still noticeable. Therefore, we first

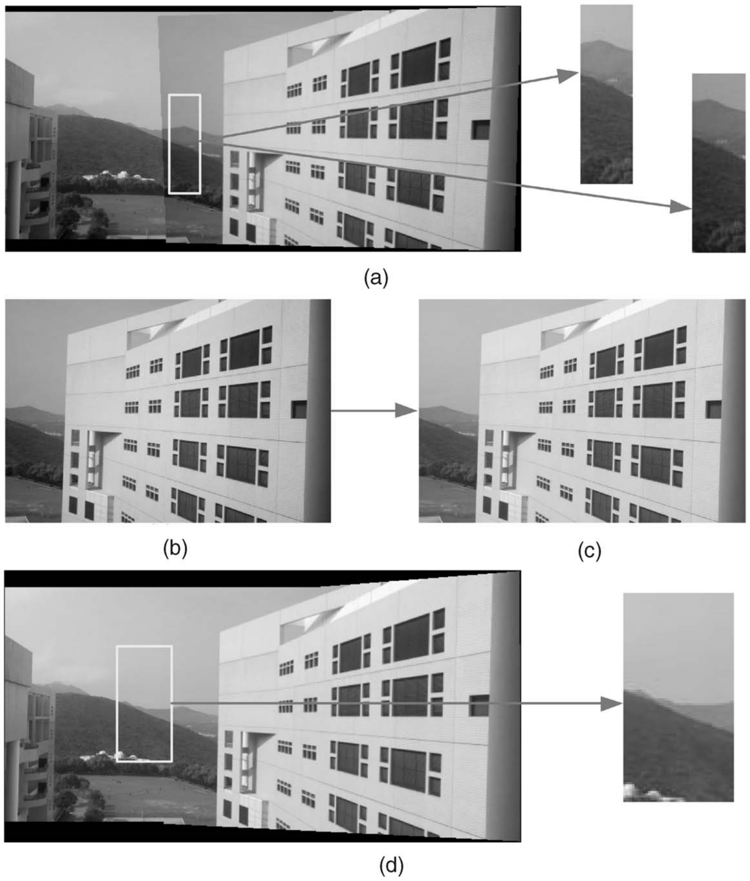


Fig. 8. The local replacement function is estimated to generate a seamless mosaic. The reference image is roughly devignetted by using input focal length. (a) Result mosaic with vignettes: the intensity seam is noticeable. (b) Initial input image. (c) Result image after local replacement. (d) Result mosaic after local replacement.

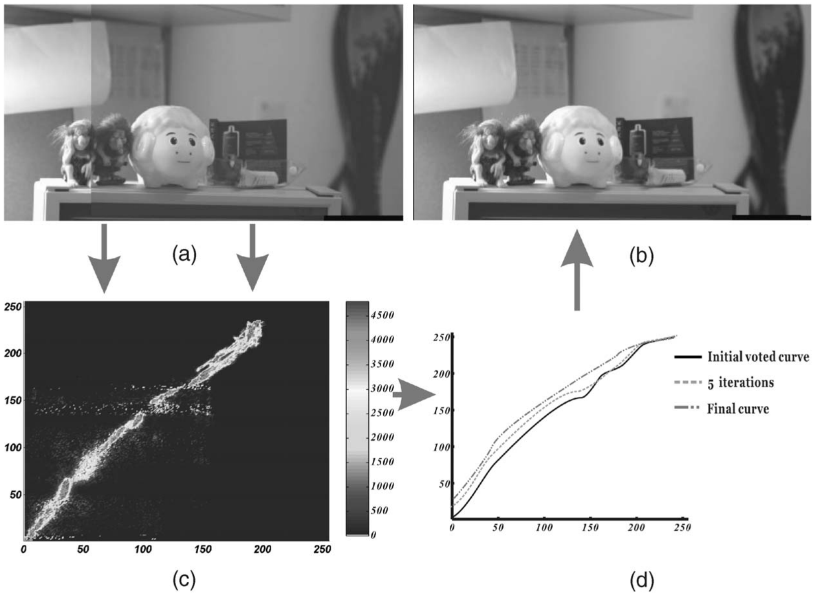


Fig. 9. Intensity voting example. (a) Input mosaic with large exposure difference and misregistration. (b) The final seamless and globally aligned mosaic. (c) The points in the tensor voting space (joint image space) are very noisy. The gray scale indicates the curve saliency $\lambda_{max} - \lambda_{min}$ at all sites after applying tensor voting. (d) The global replacement curve refinement process in the green channel.

roughly devignette the reference image by the user input focal length. Then, the intensity in other input images is automatically compensated by the voting process in local replacement.

To illustrate the global intensity alignment process, we use two images that are locally aligned and roughly registered, as shown in Fig. 9a. The corresponding joint image space is plotted in Fig. 9b, which contains a large amount of noise and

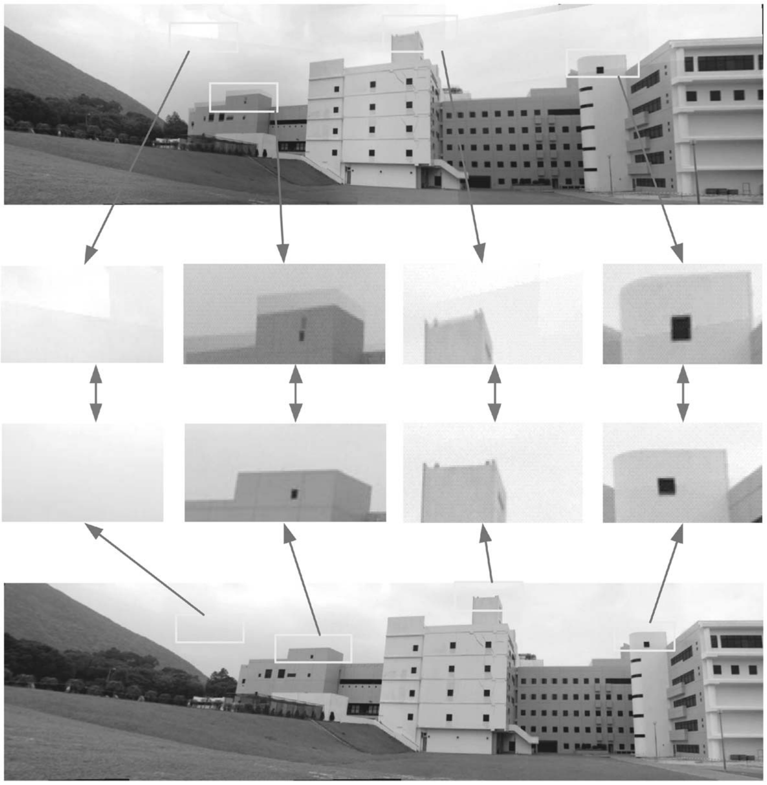


Fig. 10. The top and bottom show the result of constructing large mosaics without and with color alignment, respectively. The same image registration methods are applied without color blending. With our global and local color compensation, the final mosaic has significant improvement in quality.

holes (e.g., no point instance in some votee columns), mainly caused by misregistration. Moreover, to demonstrate the robustness of our method, we add Gaussian noise of mean 0 and variance 5 to the space. After tensor voting, a monotonic curve is generated, as shown in Fig. 9c. All noise points are eliminated and all holes are filled. The intermediate and final curves are also plotted in Fig. 9c for reference. The final seamless result is shown in Fig. 9d.

The panorama in Fig. 10 shows the ability of our algorithm to register a large number of images and to simultaneously compensate intensity or color difference globally and locally. They are aligned image-by-image.

The robustness of feature enhancement is the major advantage in our voting process. We observe that color/intensity inconsistency (Fig. 11) is more detrimental than image noise due to the measurement process in mosaics construction or image registration. To illustrate this phenomenon, we use several frames captured from a low quality Web camera, where the noise level is quite high. Significant flickering is observed in the image sequence. Fig. 11a shows that, without any color/intensity correction, the resulting mosaic cannot be correctly registered in the presence of a large amount of image noise. We perform the global color correction and present result in Fig. 11b. Without large color/intensity differences, the images are successfully aligned. Note that several image filters can be employed here to eliminate noises. However, they are not performed in this example to show the raw output of our method.

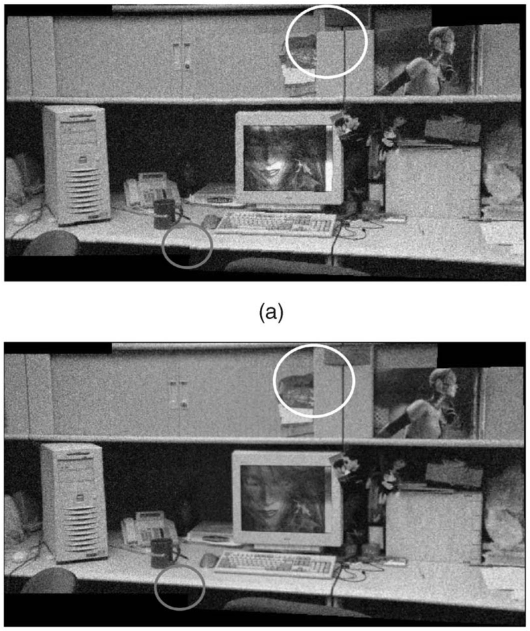
7.2 Image Mosaicking in the Presence of Occluding Objects

In this section, we extend our intensity correction approach to perform image mosaicking in the presence of large occluding objects. For instance, as shown in Fig. 12, given an image pair containing global intensity disparity, we want to remove occlusion so as to construct a visually seamless mosaic.

In practice, the occlusion cannot be simply distinguished directly by taking the intensity difference. The initial misalignment further complicates this problem. Thus, in most cases, to detect and eliminate occlusion, it is often assumed that the images are already correctly aligned and no intensity discrepancy exists [29].

As mentioned in Section 7.1, in most cases, our algorithm converges in less than 10 iterations, resulting in a roughly diagonal curve in the voting space. Nevertheless, when our algorithm does not converge, severe misalignment exists which may be caused by large occlusion. This effect is manifested into a partial intensity change that cannot be corrected in the global voting process. One example is shown as the oscillatory curve in Fig. 12d.

According to this observation, we can detect occlusions between images in our framework. To further deocclude them, we make an assumption that the occluding object is nearly piecewise constant in color (such as the human figure shown in Fig. 12), which guarantees that the occlusion does affect registration accuracy and is distinguishable from small misalignment of images or noises. Specifically, if our method cannot converge after some iterations, the following algorithm is triggered:



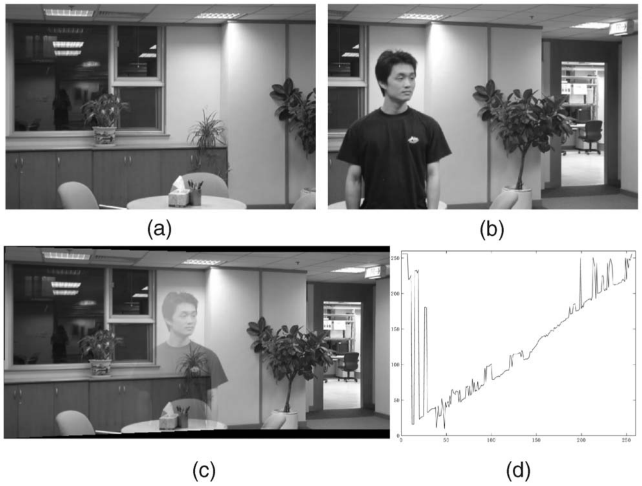
(a)

(b)

Fig. 11. Intensity correction for noisy images. These images are obtained from an inexpensive Web camera where a large amount of random noise is present. The top image is the mosaic constructed with the original image sequence, while the bottom one is generated after our color correction. Better alignment, as well as the intensity transition, is achieved.

1. Calculate the color difference between two corresponding pixels in the overlapping area (at positions returned after several iterations) and generate a new image I_0. That is, $I_0(x,y) = |I(x,y) - w(I'(x',y'))|$. I_0 is shown in the top of Fig. 13. Note that the large values in I_0 are either caused by occlusion (the human) or misregistration (the shrub).
2. Now, I_0 contains color differences for all pixels which need to be grouped into different regions according to their values. Accordingly, we adopt the mean shift method [4] to perform the grouping and segmentation on these images for the following steps.
3. We associate each region r_i in I_0 with a weight, defined as $w_{r_i} = \sum_{N_{r_i}} s_{r_i}$, where N_{r_i} is the pixel number of regions r_i and s_{r_i} is the pixel value. The regions are then sorted according to w in descending order.
4. The following iterations are performed until converegence:

    a. Denote the region in I_0 with the largest w by R_0, and map the region into original I and I'. Let R and R' be the mapped regions in I and I', respectively. For example, in Fig. 13, the white region in the top image is mapped to two different regions on the bottom images, which are indicated by the arrows as shown.
    b. Remove R and R' from I and I', respectively, which are classified as occlusion instead of intensity disparity.
    c. Align the two images again. If convergence tolerance has not been reached, go to Step 4a.
5. Finally, the user can choose which image (I or I') is to be outputted in the converged overlapping area. Alternatively, it can be determined by certain user-defined weighting function proportional to the region sizes R and R'. We show our result in Fig. 14.



(a)

(b)

(c)

(d)

Fig. 12. Mosaicking with a large occluding object. (a) and (b) are two input images, where (b) contains a large partial occlusion. (c) Image registration with occlusion. (d) After several iterations, the replacement function curve in the green channel is still fluctuating.

Since our method only removes homogeneous regions one at a time in each iteration in Step 4, the algorithm continues on improving the image alignment while large color-inconsistent regions are removed gradually. Our multiscale voting scheme can handle large holes resulting from region removal and the convergence condition is similar to that of the static scenes. Furthermore, we still use the local fitting algorithm to enforce monotonicity requirement. Fig. 14 shows the converged replacement function curve in the green channel.

Note that this algorithm does not cause any damage to the overlaid result. In addition, we do not distinguish the occluded or occluding objects. The output mosaics simply represent the final result with maximum color/intensity consistency. We use a single occluding object in our examples to facilitate the depiction. However, the method can be generalized to more of them.

In some cases, due to camera shake or scene movement, the occluding objects may not be projected into a sharp image. In the following section, we describe another application related to *motion deblurring*, which can be applied before employing the above image mosaicking technique.

Again, our proposed method on image intensity compensation also utilizes the tensor voting framework.

7.3 Image Intensity Compensation

Acquiring crisp images in dim light with a hand-held camera is difficult. In this section, we propose a novel method using tensor voting to perform intensity correction. Given two defective images I' and I: I' is taken with almost instantaneous exposure time. Thus, I' is an underexposed image with unacceptable colors (too dark), but is largely free of motion blur. I is a normal image acquired under an extended period of exposure time. The color and brightness of I is acceptable, while it is blurry due to camera or scene movement. The two images are taken successively so that their image centers are almost the same.

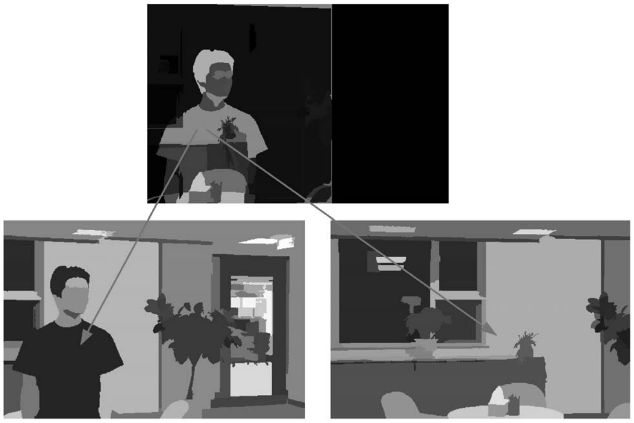


Fig. 13. The matching process. A region in the top image can be mapped to different regions in the original images. We regard regions in the bottom images as mapped ones if they cover more than 90 percent of all pixels in the region in the top image.

Our goal is to infer a color mapping function g such that the resulting image I_C is constructed by applying g to the underexposed image I', that is, $I_C(x, y) = g(I'(x, y))$.

We propose encoding *color matches* and *spatial structures* of I and I' as tensor saliencies and show that our tensor voting approach for replacement function estimation can be readily adapted to image intensity correction. Most previous works achieve the goal by blind deconvolution, under different assumptions on point spread functions (PSFs) [9], [31], [15], [30]. However, these point spread functions are sometimes very difficult to estimate: In a single image, different parts of the scene may require different PSFs. Alternatively, by making use of additional hardware, Ezra and Nayar [1] proposed a hybrid system consisting of a primary camera (of high spatial resolution) and a secondary camera (of high temporal resolution) to estimate motion directions. Our proposed approach does not directly deblur images. Instead, by using replacement functions, we robustly estimate intensity compensation from I' to I with the following constraints.

Color matches by histogram equalization. Since a scene with higher brightness always generates brighter pixels [10], the colors in I' and I can be matched in descending order of pixel intensity values. We use histogram equalization to estimate color mappings. The converted histogram is then encoded into a set of $\{(c_L^m, c_H^m)\}_{m=1}^{D}$, where D is the color depth.

Spatial constraints by oversegmentation and region matching. Since the two images are taken successively, there exists a strong spatial constraint between them. Given a homogeneous image region, we observe that the color toward the center of the region is less affected by blurring than the region boundary.

To perform region matching, we first oversegment I so that each region in I contains similar colors. Afterward, we perform morphological eroding with the same set of parameters for each region in I. The total number of iterations to completely erode the regions is recorded. Then, we sort all iteration numbers in descending order and select the first M regions as the most possible candidates. The positions of these region centers are selected as matching positions. Finally, we pick pixel pairs $\{(s_L^m, s_H^m)\}_{m=1}^{M}$ in I and I' in the matching positions.

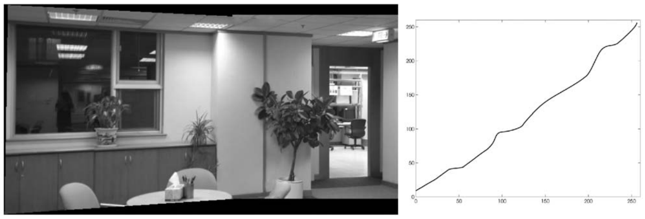


Fig. 14. The deocclusion result. Occlusion is detected and removed in our tensor voting framework. The converged replacement function curve in the green channel is also shown.

Inference of replacement function. The set of joint images $\{(c_L^m, c_H^m)\} \cup \{(s_L^m, s_H^m)\}$ obtained above corresponds to color matching and spatial constraints, indicating higher confidence for the underlying replacement function curve to pass through. We encode ball tensors with larger saliencies at these positions, while smaller saliencies are for all others. Finally, the similar noniterative tensor voting in the joint image space (I', I) is performed to infer the replacement function g. Note here that the voting space is less complicated than $(\tilde{I}', I)$ for image mosaicking because no warping is necessary in this application.

In our previous work [13], a Bayesian approach was proposed for two-image deblurring, also using two defective images. In algorithmic terms, our current approach uniformly transforms the color statistics and spatial constraints into soft constraints for voting in the joint image space, by increasing the saliency of the encoded tensors. In [13], the likelihood and prior density functions are separately modeled, taking proximity and correspondence uncertainty into consideration, whereas here tensor voting elegantly encodes them uniformly into its voting fields. Comparable results are produced in both methods.

Results. We show two results on image intensity compensation given two imperfect images due to camera shaking and scene movement.

The bridge example in Fig. 15 shows the ability of our method to optimally combine the color information from two input images. Our resulting image I_C is very crisp and clear, as shown in Fig. 15c. The two input images in Figs. 15a and 15b are taken with shutter speeds $\frac{1}{30}$ s and $\frac{1}{2}$ s, respectively. The replacement function curves for the three channels inferred by tensor voting are shown in Fig. 15d. Note that the shape of the replacement function curves is very different from that in Fig. 9 because the two input images in this example have very large intensity disparity.

Our method also effectively addresses the problem of object movement or deformation when the object movement is too fast for normal exposure interval. Fig. 16 illustrates one experiment. The input image taken with normal exposure is locally blurred, that is, the underlying PSF has no uniform representation in the whole image, making deconvolving methods vulnerable to failure. In our method, by slowing down the camera shutter speed by three stops, we produce an improved I_C with a largely reduced blurring effect.

7.4 High Contrast Image Correction

The above idea of image intensity correction given two defective images can be applied to correct high contrast

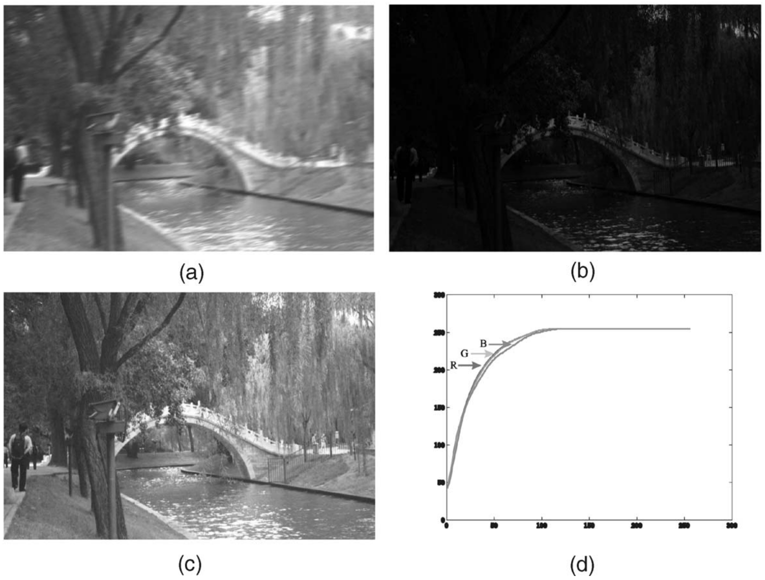


Fig. 15. Bridge example of image correction. (a) and (b) are the two input images. (c) Shows our corrected result. (d) Shows the inferred replacement function curves for the three channels.

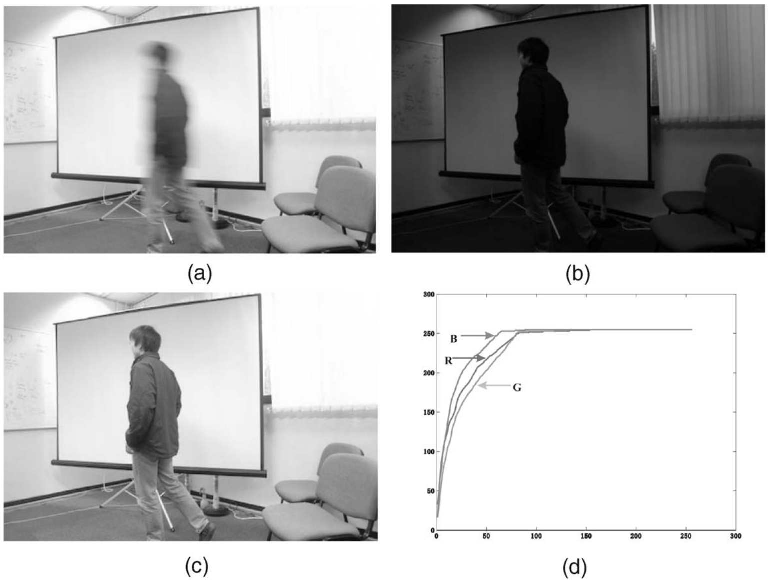


Fig. 16. Walking example. (a) and (b) are input images. Our result is shown in (c), which demonstrates that local blurring in images can be naturally handled. (d) Shows the voted replacement function curves for the three channels using our approach.

scene as well. Again, two images I' and I are successively taken, where I is a saturated image. The joint image space is still (I', I).

The only modification to deal with high contrast scenes is the use of the color transfer function in [21] instead of histogram equalization, where no limit is imposed on the maximum value of the transferred color. In our method, all *non*saturated pixels in I are used for color transfer to I'. After applying [21], the mapping result of I' exceeds the color depth, which extends the saturated pixels to larger color values. Thus, we construct a higher range image to reveal details in both bright and dark regions. The same tensor voting is used to infer the resulting replacement function. In the whole process, the saturated pixels which encode improper values are not taken into consideration.

We present our results in Fig. 17: Figs. 17a and 17b are I and I', respectively, and Fig. 17c is reconstructed by setting $g(\cdot)$ as the original histogram equalization function where most details are buried in saturated white papers. Fig. 17d is our final result with enhanced colors and details by modifying $g(\cdot)$ to use the color transfer method in [21]. Obviously, characters are preserved in white papers. Tone mapping [8] is performed to display the image we constructed in Fig. 17d.

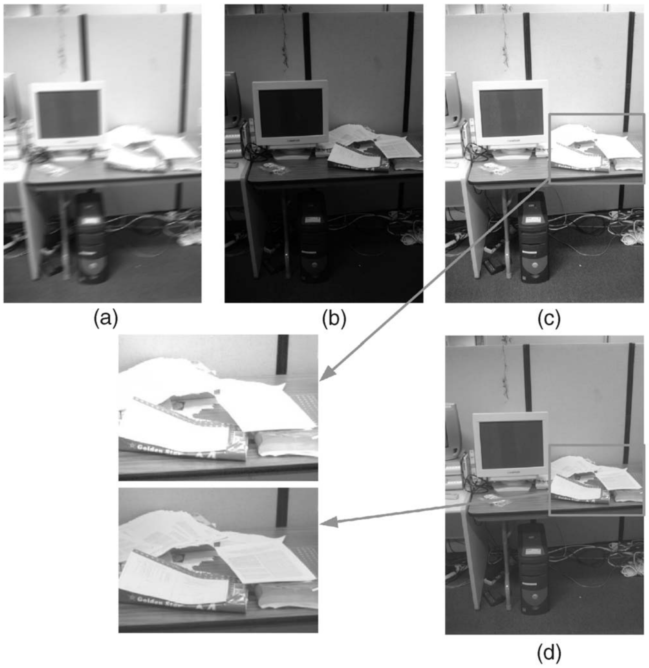


Fig. 17. Image correction for a high contrast scene using our voting approach. (a) I has a saturated area. Part of the color is acceptable, but the image is too blurry. (b) I' has sharp structure information, but it is too dark. (c) The result produced by applying a conventional histogram equalization as preprocessing where most details are buried in saturated white papers. (d) Our final result I_C, where pixel intensity values are enhanced. As a result, fine details are preserved, especially in the originally saturated area. The bottom left images are selected enlarged portions from (c) and (d).

8 Discussion

8.1 Direct Estimation versus Tensor Voting

Tensor voting plays an important role in our method to estimate a correct intensity replacement curve. This method is analogous to the Markov Random Field (MRF) assumption [3], where neighbors can give some potential support for each site, either positive or negative.

There are two main steps in the tensor voting method to facilitate curve extraction in the voting space. The first is the encoding process to enrich an input site which does not have any orientation information initially. In the joint image or voting space, each site is associated with a 2D ellipse (initially, a circle, represented by arbitrary orthonormal eigenvectors). The other main step is the voting process, which is analogous to applying the MRF rule: By gathering neighborhood information for each site, the resulting space ultimately amplifies the support of true data points, while noises are suppressed at the same time.

Furthermore, our voting method infers *new* intensity values subject to the smoothness (voting field) and the monotonic constraints (local fitting algorithm) which are imposed on the result replacement curve. As shown in the previous application sections, tensor voting is robust to a large amount of noise.

We give an illustration to compare between our tensor voting method and direct estimation, where the estimated replacement function only makes use of existing intensities.

The direct estimation below is a curve extraction process without tensor voting. We perform the following steps for all points in the same joint image space:

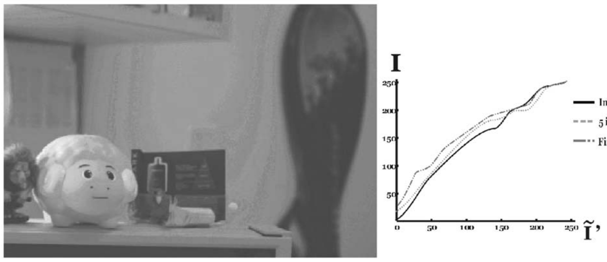


Fig. 18. Comparison between tensor voting and direct estimation by using the same example as in Fig. 9. In both figures, we compensate for the intensity disparity and generate recolored images. Here, intensity artifact is obvious. Fig. 9 shows the intensity with natural transition, where the final curve is also smoother.

1. For each site in the space, we attach an instance number h, which is the total number of instances of $(\tilde{I}'(\cdot), I(\cdot))$ in the overlapping area (as described in Section 5.2).
2. In each votee column x_k (Fig. 3), we simply compare h for all sites and select the site with the largest value h. That is,

$$P_k^* = \{(x_k, y_k^*) | h(x_k, y_k^*) > h(x_k, y_k) \\ \text{for all } 1 \leq y_k \leq \lfloor D/n \rfloor\}.$$

3. Other procedures, such as local fitting and iteration, remain the same, as stated in the previous sections. The replacement function thus obtained is used in global intensity correction.

Therefore, the only difference between direct estimation and tensor voting is the voting process. We show our comparison in Fig. 18. By simply considering the maximum h in each column in the 2D space, we cannot distinguish noise from true data points. This noise may be generated by misregistration, image noise, and occlusion. Consequently, the "recolored" image contains unnatural transition and error replacement (Fig. 18). Through our tensor voting, the noisy data are refined by collecting neighborhood support, which reinforces a site if it is very likely to be correct (that is, lying on a smooth and monotonic curve). The tensor voting result was already shown in Fig. 9.

9 Conclusion

We have described a unified and robust approach that uses tensor voting to address the problem of global and local intensity alignment for image registration. Our method globally and locally adjusts intensities of two overlapping images, without assuming a complex camera model or any simplified assumptions other than the monotonic constraint. Our iterative scheme converges quickly, thanks to the robust estimation of the replacement functions by tensor voting. Compared with other techniques, tensor voting is novel as it provides a fundamentally different approach to performing intensity alignment and effective since an optimal function under the monotonic constraint is obtained. In the whole process, only a rough focal length for the first reference image is required. We have applied our voting methodology to a variety of applications: image mosaicking of static scenes, image mosaicking in the presence of occluding objects, intensity compensation, and correcting saturated images.

Acknowledgments

The authors would like to thank all the reviewers for their constructive comments to improve the final manuscript. This research is supported by the Research Grant Council of Hong Kong Special Administrative Region, China: HKUST6175/04E.

References

[1] M. Ben-Ezra and S.K. Nayar, "Motion Deblurring Using Hybrid Imaging," *Computer Vision and Pattern Recognition,* 2003.

[2] P.J. Burt and E.H. Adelson, "A Multiresolution Spline with Applications to Image Mosaics," *ACM Trans. Graphics,* vol. 2, no. 4, pp. 217-236, Oct. 1983.

[3] R. Chellappa and A.K. Jain, *Markov Random Fields: Theory and Applications.* Academic Press, 1993.

[4] D. Comanicu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[5] J. Davis, "Mosaics of Scenes with Moving Objects," *Computer Vision and Pattern Recognition,* 1998.

[6] P. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *Proc. SIGGRAPH,* pp. 369-378, 1997.

[7] A. Edirisinghe, G.E. Chapman, and J.P. Louis, "Radiometric Corrections for Multispectral Airborne Video Imagery," *Photogrammetric Eng. & Remote Sensing,* vol. 67, no. 8, pp. 915-924, Aug. 2001.

[8] P. Shirley, E. Reinhard, M. Stark, and J. Ferwerda, " Photographic Tone Reproduction for Digital Images," *Proc. SIGGRAPH,* pp. 267-276, 2002.

[9] R. Fabian and D. Malah, "Robust Identification of Motion and Out-of-Focus Blur Parameters from Blurred and Noisy Images," *CVGIP: Graphical Models and Image Processing,* 1991.

[10] M.D. Grossberg and S.K. Nayar, "What Can Be Known about the Radiometric Response Function from Images?" *Proc. European Conf. Computer Vision,* May 2002.

[11] D. Hasler and S. Susstrunk, "Colour Handling in Panoramic Photography," *Proc. SPIE,* Jan. 2001.

[12] G.E. Healey and R. Kondepudy, "Radiometric CCD Camera Calibration and Noise Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 3, pp. 267-276, Mar. 1994.

[13] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum, "Bayesian Correction of Image Luminance with Spatial consideration," *Proc. European Conf. Computer Vision,* 2004.

[14] J. Jia and C.-K. Tang, "Luminance Voting: Image Registration with Global and Local Luminance Alignment," *Proc. Int'l Conf. Computer Vision,* 2003.

[15] D. Kundur and D. Hatzinakos, "A Novel Blind Deconvolution Scheme for Image Restoration Using Recursive Filtering," *IEEE Trans. Signal Processing,* vol. 46, no. 2, pp. 375-390, Feb. 1998.

[16] S. Mann and R. Mann, "Quantigraphic Imaging: Estimating the Camera Response and Exposures from Differently Exposed Images," *Computer Vision and Pattern Recognition,* Dec. 2001.

[17] S. Mann and R. Piccard, "Being "Undigital" with Digital Cameras: Extending the Dynamic Range by Combining Differently Exposed Pictures," *Proc. IST 48th Ann. Conf.,* pp. 422-428, May 1995.

[18] G. Medioni, M.S. Lee, and C.K. Tang, *A Computational Framework for Segmentation and Grouping.* Elsevier Science B.V., 2000.

[19] G. Messina, A. Castorina, S. Battiato, and A. Bosco, "Image Quality Improvement by Adaptive Exposure Correction Techniques," *Proc. IEEE Int'l Conf. Multimedia and Expo,* 2003.

[20] T. Mitsunaga and S. Nayar, "Radiometric Self Calibration," *Computer Vision and Pattern Recognition,* pp. 374-380, 1999.

[21] M. Gooch, B. Reinhard, E. Ashikhmin, and P. Shirley, "Color Transfer between Images," *IEEE Computer Graphics and Applications,* pp. 34-40, 2001.

[22] H. Sawhney and S. Ayer, "Compact Representations of Videos through Dominant and Multiple Motion Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 8, pp. 814-830, Aug. 1996.

[23] Y.Y. Schechner and S.K. Nayar, "Generalized Mosaicing: Wide Field of View Multispectral Imaging," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 10, pp. 1334-1348, Oct. 2002.

[24] W. J. Smith, *Modern Optical Engineering,* third ed. McGraw-Hill, 2000.

[25] R. Szeliski, "Video Mosaics for Virtual Environments," *IEEE Computer Graphics and Applications,* pp. 22-30, Mar. 1996.

[26] R. Szeliski and H.-Y. Shum, "Creating Full View Panoramic Image Mosaics and Environment Maps," *Proc. SIGGRAPH,* pp. 251-258, 1997.

[27] R. Szeliski and H.-Y. Shum, "Construction of Panoramic Image Mosaics with Global and Local Alignment," *Int'l J. Computer Vision,* vol. 36, no. 2, pp. 101-130, 2000.

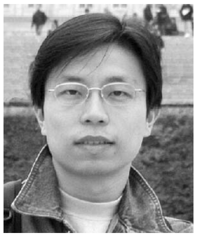
[28] Y. Tsin, V. Ramesh, and T. Kanade, "Statistical Calibration of CCD Imaging Process," *Proc. Int'l Conf. Computer Vision,* July 2001.

[29] M. Uyttendaele, A. Eden, and R. Szeliski, "Eliminating Ghosting and Exposure Artifacts in Image Mosaics," *Computer Vision and Pattern Recognition,* 2001.

[30] A. Lantzman, Y. Yitzhaky, I. Mor, and N.S. Kopeika, "Direct Method for Restoration of Motion-Blurred Images," *J. Optical Soc. Am., A,* vol. 15, no. 6, pp. 1512-1519, June 1998.

[31] Y. Levy, Y. Yitzhaky, G. Boshusha, and N.S. Kopeika, "Restoration of an Image Degraded by Vibrations Using only a Single Frame," *Optical Eng.,* 2002.

[32] Z. Zhu, E.M. Riseman, and A.R. Hanson, "Parallel-Perspective Stereo Mosaics," *Proc. Int'l Conf. Computer Vision,* July 2001.



Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2004. He is currently an assistant professor in the Computer Science and Engineering Department, the Chinese University of Hong Kong. He is also a visiting scholar at the Visual Computing Group of Microsoft Research Asia. His research interests include image and video processing, image-based rendering. and pattern recognition. He is a member of the IEEE Computer Society.

Chi-Keung Tang received the MS and PhD degrees in computer science from the University of Southern California, Los Angeles, in 1999 and 2000, respectively. He has been with the Computer Science Department at the Hong Kong University of Science and Technology since 2000, where he is currently an assistant professor. He is also an adjunct researcher in the Visual Computing Group of Microsoft Research, Asia, working on various exciting research topics in computer vision and graphics. His research interests include low to mid-level vision such as segmentation, correspondence, shape analysis, and vision and graphics topics such as image-based rendering and medical image analysis. He is a member of the IEEE Computer Society.