

Image Registration with Global and Local Luminance Alignment*

Jiaya Jia and Chi-Keung Tang

Vision and Graphics Group, Computer Science Department

Hong Kong University of Science and Technology

{leojia,cktang}@cs.ust.hk

Abstract

Inspired by tensor voting, we present **luminance voting**, a novel approach for image registration with global and local luminance alignment. The key to our modelless approach is the direct estimation of **replacement function**, by reducing the complex estimation problem to the robust 2D tensor voting in the corresponding voting spaces. No model for replacement function is assumed. Luminance data are first encoded into 2D ball tensors. Subject to the monotonic constraint only, we vote for an optimal replacement function by propagating the smoothness constraint using a dense tensor field. Our method effectively infers missing curve segments and rejects image outliers without assuming any simplifying or complex curve model. The voted replacement functions are used in our iterative registration algorithm for computing the best warping matrix. Unlike previous approaches, our robust method corrects exposure disparity even if the two overlapping images are initially misaligned. Luminance voting is effective in correcting exposure difference, eliminating vignettes, and thus improving image registration. We present results on a variety of images.

1 Introduction

Creating panoramas or mosaics is still an inexpensive and commonly adopted method to generate photographs of higher resolution and/or of wider angle of view. To successfully generate a visually acceptable or seamless mosaic from a few images, many registration methods [12, 13, 9, 16] 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. Seams in the image composite are quite noticeable. Worst, these complications may lead to image misregistration. Image mosaics with large exposure difference results in unnatural color transition and misregistration.

There are several factors that influence the luminance recorded in cameras, namely, exposure variance among images, white balance of a digital camera, Gamma correction,

vignetting¹ and digitizer parameters [3, 5, 15]. When two images are aligned, in order to eliminate seams and artifacts due to the above factors, feathering [13] can be applied to blend two overlapping images locally. Alternatively, a model for the response function [15] can be estimated to correct the luminance or exposure difference of the overlapping areas.

Estimating response function is an under-constrained problem. One common approach to tackle the problem is to perform radiometric calibration, by taking several images of a static scene under different lighting conditions. The consistency requirement of conditions other than exposures is crucial to the stability of the resulting model. This is difficult especially when image noise and ambiguities are very common. Moreover, even if such a model can be reliably estimated, the evaluation metric may not be available to judge the efficiency or optimality of the obtained model.

In this paper, we propose a *modelless* and effective approach to address the luminance disparity when two overlapping images are registered. Our approach is called *luminance voting*, which is inspired by 2D tensor voting, a modelless approach for structure inference in 2D. Luminance voting robustly removes image outliers and estimates an optimal mapping function for global and local luminance replacement. The optimality of our estimated response function is subject to the only constraint: the monotonic constraint, that is, higher exposures should always generate brighter pixel colors, and vice versa. Unlike the algorithm in [3], we only need the focal length of the first image to eliminate natural vignetting. No other curve or simplifying model is assumed.

For ease of reference, we summarize the notions and terminologies used in this paper below: $I(x, y)$ is the pixel color in position (x, y) of an image I . The corresponding scene illuminance of V , recorded by image I , satisfies

$$I = u(V) \quad (1)$$

where $u(\cdot)$ is the response function [15] and can be further decomposed as $(f \cdot k)(\cdot)$. $f(\cdot)$ is called *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

¹It indicates the unintended darkening at image corners and/or borders in a photographic image. Some different image formation mechanisms may account for vignetting [10]. Among them, natural and optical mechanisms inherent in many lens design are main causes.

*This research is supported by the Research Grant Council of Hong Kong Special Administration Region, China: HKUST6171/03E.

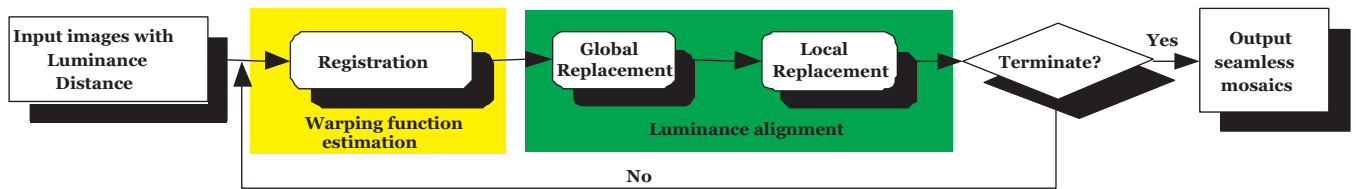


Figure 1: Overview of the luminance voting in image registration.

global color mapping process, i.e., 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 the pixels at the image center, even if incoming irradiance is similar. 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 Sec. 4.1.

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 color matching algorithms, which are based on luminance voting. In sections 5 and 6, we describe the details of estimating the global and local response function algorithms. Section 7 discusses and presents our results. Finally, we conclude our paper in section 8.

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.

2.1 Mosaic registration with exposure correction

Many advanced registration methods have been developed in recent years. To correct the exposure difference, one direction is to blend pixel colors in overlapping area. Video mosaic [11] is a registration algorithm to estimate homography by Levenberg-Marquardt method. To reduce visible artifacts, it blends overlapping area by a bilinear weighting function. However, some “mottling” spots still remain under different exposures. In [13], a feather-based algorithm can be used for better blending effect, but unnatural seam is still inevitable for different lighting conditions. Burt et al [1] made use of a multi-resolution spline to perform blending. Uyttendaele et al [14] used a block-based method to compute the response function for mapping illuminance. By using averaging and interpolation functions, 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 only performed at the overlapping area to transit images from one exposure environment to another more naturally.

2.2 Radiometric calibration

The other class of approach to eliminate the illuminance difference is radiometric/illuminance calibration [8, 2, 6, 15,

5]. By taking several static images with different exposures, response function $u(\cdot)$ or its inverse $u^{-1}(\cdot)$ is estimated to generate high range radiance maps. In [8], $u^{-1}(\cdot)$ 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 . Ramesh et al [15] adopted a noise contaminated pattern, and used maximum-likelihood method to estimate $u(\cdot)$. However, these methods are usually applied to static scene with some assumed imaging or noise models. Additional knowledge, such as exposure information [2], is required. Given only two unregistered images with one 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 mapping in order to align the luminance difference between images.

Hasler et al [3] combine image registration and camera internal parameters estimation into one optimization process, and globally eliminate luminance difference between images. However, they assume that the parametric models, e.g., gamma correction and vignetting, of the camera are known.

3 Review of 2D tensor voting

In this section, we give a concise review on 2D tensor voting [7]. Our approach is a feather curve extraction and noise elimination method that makes use of a *tensor* for token representation, and *voting* for communication among tokens. Tensor and voting are brought together by a voting field. Voting field is a dense tensor field for postulating smooth connections among tokens.

We are interested in answering the following geometric question in 2D. 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. 2(a) illustrates the situation. We claim that the osculating circle connecting O and P is the most likely connection [7], since it minimizes the total curvature along the hypothesized circular arc, and thus encodes the smoothness constraint. The most likely normal is given by the normal to the circular arc at P (thick arrow in Fig. 2(a)). The vote strength of this normal is decayed according to the following function: $\overline{DF}(r, \varphi, \sigma) = e^{-\frac{r^2 + c\varphi^2}{\sigma^2}}$ 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. If we consider all possible locations of P in the 2D space, the resulting set of normal

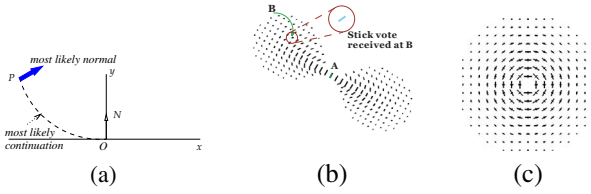


Figure 2: (a) Design of the 2D stick voting field, (b) how a voter A casts a stick vote to vote receiver (votee) B , and (c) the 2D ball voting field, obtained by rotating and integrating the 2D stick voting field.

directions thus produced constitutes the 2D stick voting field, Fig. 2(b). Note that σ is the only free parameter.

Given an input token A , how to cast vote to another token B for inferring a smooth connection between them, assuming that A 's normal is known, Fig. 2(b). First, we fix σ to determine the size of the voting field, or the *scale* of analysis. Then, we align the voting field with A 's normal, by translation and rotation. If B is within A 's voting field neighborhood, B receives a stick vote $[v_x \ v_y]^T$ from the aligned field. Hence, voting is similar to convolution, and the voting field is like a convolution mask, except that the voting result is not a scalar.

Tensor voting simply means rotation and translation to align the 2D stick voting field with A 's given normal, Fig. 2(b). 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{V}_{max} and \hat{V}_{min} . As a result, \mathbf{S} can be rewritten as $\mathbf{S} = \lambda_{max} V_{max} V_{max}^T + \lambda_{min} V_{min} V_{min}^T = (\lambda_{max} - \lambda_{min}) \hat{V}_{max} \hat{V}_{max}^T + \lambda_{min} (\hat{V}_{max} \hat{V}_{max}^T + \hat{V}_{min} \hat{V}_{min}^T)$. $\hat{V}_{max} \hat{V}_{max}^T$ is a stick tensor, with \hat{V}_{max} indicating curve normal direction. $\hat{V}_{max} \hat{V}_{max}^T + \hat{V}_{min} \hat{V}_{min}^T$ is a ball tensor.

If A 's normal is not known initially, we need to estimate it first. It is performed by voting with the 2D ball voting field, Fig. 2(c), which is obtained by rotating and integrating (by tensor sum) the vote contribution of the 2D stick voting field. The 2D ball voting field is then aligned with the A (along arbitrary direction since its normal is unknown), using exactly the same voting process described above. After decomposing the collected tensor into the corresponding eigensystem, the normal is given by \hat{V}_{max} .

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.

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 $I(x, y)$ and $I'(x'', y'')$ are corresponding pixels in the overlapping area of the two images.

The relationship between their corresponding incoming illuminance is given by

$$V(x, y) = t \tilde{V}'(x'', y'') \quad (2)$$

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

$$I(x, y) = u_I(V(x, y)) = u_I(tu_{\tilde{I}'}^{-1}(\tilde{I}'(x'', y''))) \quad (3)$$

$$= g(\tilde{I}'(x'', y'')) \quad (4)$$

$$= g(w(I'(x', y'))) \quad (5)$$

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

If I and I' are not correctly registered, we need to estimate not only the *replacement function* $g(\cdot)$ which measures the exposure and radiometry 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:

$$\min \sum (I(x, y) - g(w(I'(x', y'))))^2 \quad (6)$$

It is a non-linear and under-constrained optimization problem without model assumption to estimate two sets of parameters, i.e., $g(\cdot)$ and $w(\cdot)$, simultaneously. In our method, we estimate and refine them iteratively:

Given an estimation of *replacement function* $g(\cdot)$ with respect to a certain warping function w , in order to simplify Eqn. (6), we construct a new luminance adjusted image I'' which satisfies

$$w(I''(x', y')) = g(w(I'(x', y'))) \quad (7)$$

As a result, the luminance distance between I'' and I is shortened, and Eqn. (6) can be written as:

$$\min \sum (I(x, y) - I''(x', y'))^2 \quad (8)$$

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 color 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 is estimated (that is, I and I' are roughly registered), we can determine the *replacement function* $g(\cdot)$ by luminance voting (Sec. 4.3).

The above analysis of Eqn. (6) inspires us to decouple the estimation process of $w(\cdot)$ and $g(\cdot)$. Accordingly, we propose the refinement algorithm to alternatively solve those two sub-problems.

4.2 Refinement algorithm

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

1. Fix the estimated $g(\cdot)$ and the corresponding w , we construct a new input image I'' which adjusts color according to $g(\cdot)$, and satisfies Eqn. (7).
2. Solve the registration problem in Eqn. (8). Many methods can be used to tackle the normal homography estimation problem to refine the warping function w [12, 13, 16]. In our experiment, the robust 3-parameter rotational model together with local alignment [13] 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

$$\min \sum (I(x, y) - g(\tilde{I}'(x'', y'')))^2 \quad (9)$$

to determine the replacement function $g(\cdot)$ (section 4.3).

4. Repeat (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 illustrated in Fig. 1.

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.

Just like $f(\cdot)$ and $k(\cdot)$ in the response function $u(\cdot)$, which maps from illuminance to pixels 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. Global replacement function is to globally map colors between images, while the local replacement function is used to estimate 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 to 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, local replacement function is estimated in the input image. The final replacement function $g(\cdot)$ is constructed by compositing them, i.e.,

$$(\text{replacement function}) \quad g(\cdot) = global(local(\cdot)) \quad (10)$$

Fig. 1 shows the $g(\cdot)$ estimation process (in green). 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 model and fitting assumption, we propose a robust *luminance voting* approach, which is derived from the generic 2D tensor voting. Table 1 compares tensor voting and luminance voting.

	2D tensor voting	luminance voting
dimension	2D	2D
tensor	saliency tensor	saliency tensor
space	geometry space	joint image
constraint	no	monotonic
voters	all neighborhood	discard column votes
feather	curve extraction	saliency comparison
scale	one	multiscale

Table 1: Comparison of the 2D voting and luminance voting.

5 Estimating the global replacement function $global(\cdot)$ by luminance voting

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

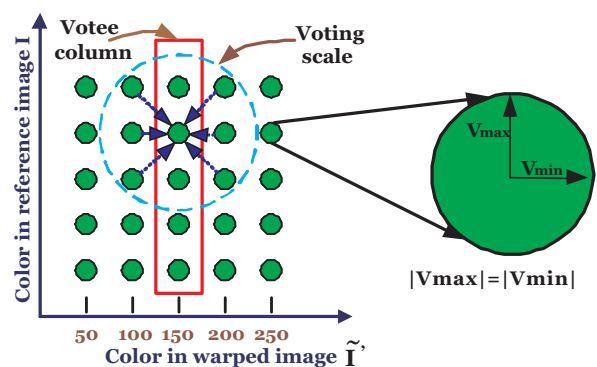


Figure 3: Voting in the joint image space (\tilde{I}, I). Left: the green dots represent tensors which communicate with each other in the voting field (cyan circle). The tensors in the red box are votees, which receive but do not cast votes. Blue arrows indicate the vote casting directions from voters to votees in the neighborhood. Right: an encoded tensor in the joint image space.

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 one joint image, where $n = 50$ in this illustration indicates the sampling density along the respective color axes.

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*. Like in the generic 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 $\hat{V}_{max}, \hat{V}_{min}$, that is,

$$\lambda(\hat{V}_{max}\hat{V}_{max}^T + \hat{V}_{min}\hat{V}_{min}^T) \quad (11)$$

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

Now we set the saliency λ of each saliency tensor $T(d_1, d_2)$ in the joint image to be proportional to the *instance number* v , where d_1 and d_2 are the coordinates. v is the total number of instances for corresponding pixel pairs $(I(x, y), \tilde{I}'(x'', y''))$ in overlapping area where $\tilde{I}'(x'', y'') = d_1$ and $I(x, y) = d_2$. Thus, we represent the equation as:

$$\lambda = b + v \cdot \Delta s \quad (12)$$

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

After saliency tensor encoding, each point in the joint image becomes a ball tensor with size (radius) equal to λ as defined by Eqn. (12). The following section describes our novel voting process for inferring the optimal global replacement function.

5.3 Luminance voting for color compensation

The key idea of luminance voting for optimal global replacement inference is twofold: (1) the robust and efficient inference by collecting tensor support in the neighborhood, and (2) the use of the *monotonic constraint*. In this section, we introduce the voting process and postpone the *monotonic constraint* description to Sec. 5.4.

Color mapping is performed from \tilde{I}' to I , where each \tilde{I}' value maps to exact one color in I . Accordingly, instead of performing curve extraction by generic tensor voting as reviewed in section 3, we apply the luminance voting process for each discrete color in \tilde{I}' , and the unique optimal correspondence in I is selected as the curve point for the global replacement function (e.g., Fig. 9(c) in the result section).

Since exactly one optimal replacement color in I is selected from a given color \tilde{I}' by the luminance voting process, other saliency tensors of the same color value in \tilde{I}' , or in the same votee column (red box 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 blue arrows.

After luminance voting, the eigensystem of the inferred saliency tensors is computed. The value at the site with maximum curve saliency $\lambda_{max} - \lambda_{min}$ in each votee column in is adopted as the optimal curve position (the curve in Fig. 9(c)). Note that the global replacement function is monotonic, thus the optimality of the curve is also constrained.

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. if $\tilde{I}'(x''_1, y''_1) > \tilde{I}'(x''_2, y''_2)$, then $I(x_1, y_1) > I(x_2, y_2)$ if $(x_1, y_1) \leftrightarrow (x''_1, y''_1)$ and $(x_2, y_2) \leftrightarrow (x''_2, y''_2)$ are corresponding pixel pairs in overlapping area.

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

We first sort samples for each vertical column in the voting space according to the resulting vote saliencies in descending

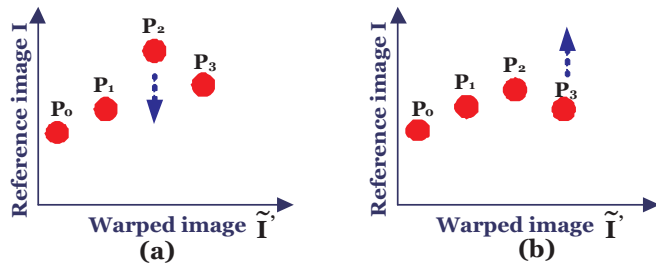


Figure 4: Starting from left to right along the \tilde{I}' axis, if one point P_3 in the curve is lower than some of the previous points, two situations may happen: (a) positions of the previous m points of P_3 are misestimated to be too high, (b) position of P_3 is a wrong estimation, which is too low.

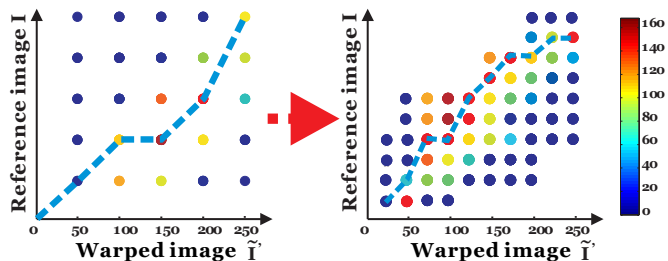


Figure 5: Global replacement estimation with multiscale luminance voting. The color of the points represents the resulting votee saliencies. The replacement curve inferred in the larger scale space (left) is propagated to the next level in the smaller scale space for inferring a more accurate curve (right). The cyan dashed curve is the inferred replacement curve. [color coded figure]

order, and put them into a queue associated with each column. A position index d for each sample is introduced, to indicate the position of the sample in the column. The local fitting algorithm proceeds as follows: If the position of one result point P_i is lower than P_{i-1} , iterate from highest saliency sample to the lowest saliency sample in the queue associated to the column, and for each position index d , iterate from P_i to P_{i-m} , where m is a user defined parameter.

1. Consider P_j in this range, we select the sub-optimal point P'_j in the same column, i.e, the point with position index $d-1$.
2. If $j < i$ and P'_j in the voting space is between P_{j-1} and P_j in vertical position, replace P_j with P'_j .
3. If $j = i$ and P'_j is higher than P_j , replace P_j with P'_j (Fig. 4(a)).

The iteration ends when $P_i \geq P_{i-1}$. This local fitting algorithm usually terminates in less than three iterations (Fig. 4(b)).

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 the overall global luminance voting algorithm as follows:

1. Fetch a new value d_1 along the \tilde{I} -axis, and set the temporary variable $j = 0$. Go to (5) if there is no more new value.
2. Fetch a new value d_2 in I -axis, and generate the initial saliency tensor at (d_1, d_2) in the joint image space.
3. Luminance voting with the ball voting field is used to infer and refine the saliency tensor at (d_1, d_2) . Let the inferred curve saliency be s . The larger s is, the more likely the point is on the replacement curve. If s is too small, the point is likely to be an outlier or noise point to be eliminated. Hence, we take s as the measurement: if $s > j$, set j to be s .
4. When all tensors at votee column d_1 have been voted for, set the point with maximum s in the column to be the curve point, and go to (1). Else go to (2).
5. Run local fitting algorithm.

5.6 Multiscale luminance voting

To speed up the voting process, we apply a multiscale voting scheme to reduce the effect of noise due to misregistration and vignettes.

The voting space can be considered as a uniformly sampled saliency image. Hence, we apply the Gaussian pyramid algorithm to construct image hierarchy such that the image of a given level has only one fourth of the points in the next lower level. Suppose we vote and generate the function curves in level $i - 1$. Only points on the curve are propagated to level i 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 results in images with similar luminance. 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, we apply another voting process to estimate the local replacement.

6 Estimating the local replacement function $local(\cdot)$ by luminance voting

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 lens and apertures are always known to be 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. Hence, we partition the image into uniform concentric annuli, as shown as T_i in Fig. 6(a). Each annulus has the same width w which is analogous to uniform sampling density n in the joint image space for global replacement estimation (Sec. 5).

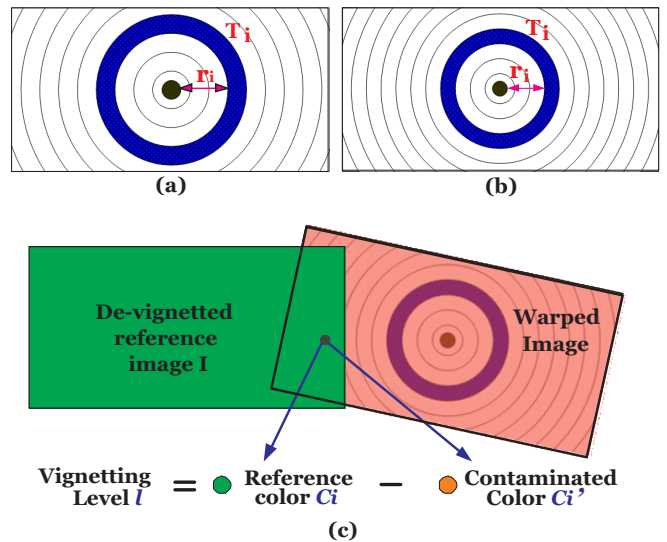


Figure 6: Multiscale voting space constructed by concentric annuli for estimating local color compensation. (a) Voting space in lower level with large scale. (b) Higher level with small scale. (c) Estimation of vignetting by measuring the color difference.

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 (de-vignetted) color in the input image be C_i , and the contaminated color after vignetting be C'_i . Then,

$$l = C_i - C'_i \quad (13)$$

To estimate the unknown C_i , we first roughly de-vignette the first reference image by substituting the input focal length into the natural vignetting equation [3]. Then, C_i can be regarded as the corresponding color in the reference image in the overlapping area. The process is illustrated in Fig. 6(c).

Consequently, our (r, l) space is analogous to the joint image space (\tilde{I}, I) for global replacement curve estimation, and the saliency tensors are also encoded in the same way. Moreover, similar monotonic constraint also applies here, since pixels on the image border (large r) are darker than those near the image center (small r). As a result, the same luminance voting process can be applied to estimate local replacement function.

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

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

7 Discussion and results

7.1 Fast versus incremental color replacement

In Sec. 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, it introduces noise and unnatural artifacts as in Fig. 7(a), when large exposure or white balance difference exist. To overcome this drawback, we change the replacement function from $g(\cdot)$ to *incremental color replacement* $\Delta g(\cdot)$, and modify step (1) in the algorithm of Sec. 4.2 as follows:

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

$$\Delta g(\tilde{I}') = \left(\frac{1}{\Delta t}\right) \cdot (g(\tilde{I}') - \tilde{I}') + \tilde{I}' \quad (14)$$

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

$$w(I''(x', y')) - \Delta g(w(I'(x', y'))) = 0 \quad (15)$$

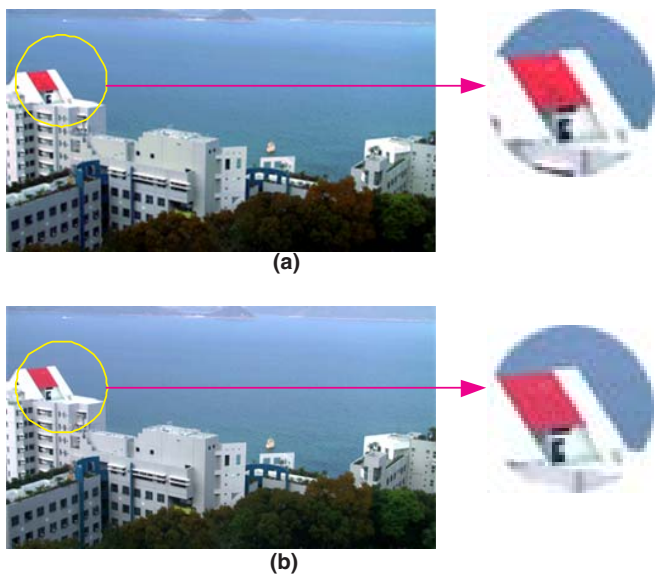


Figure 7: (a) Fast versus (b) incremental color replacement. Incremental replacement generates a more natural image composite.

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

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

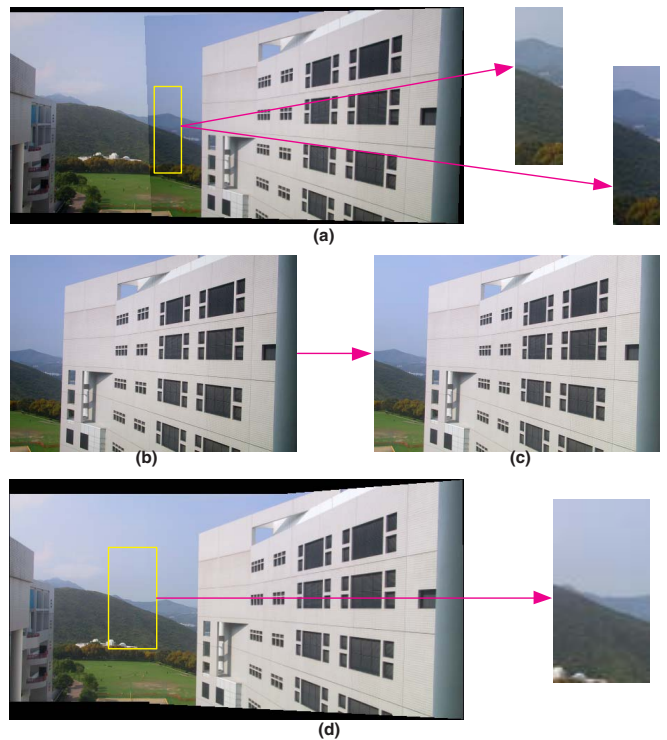


Figure 8: Local replacement function is estimated to generate seamless mosaic. The reference image is roughly de-vignetted by using input focal length. (a) Resulting mosaic with vignettes, intensity seam is noticeable. (b) Initial input image. (c) Resulting image after local replacement. (d) Resulting mosaic after local replacement.

7.2 Results

Fig. 8 shows the results of local replacement estimation. Initially, although the global luminance for two images is quite close, vignette is still inevitable. Therefore, we first roughly de-vignette the reference image by the user input focal length. Then the luminance in other input images is automatically compensated by voting process in local replacement [4].

To illustrate the global luminance alignment process, we use two images that are locally aligned and roughly registered, as shown in Fig. 9(a). The corresponding joint image space is plotted in Fig. 9(b) which contains a large number of noise and holes (e.g., no point instance in some vote 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 luminance voting, a monotonic curve is generated as shown in Fig. 9(c). All noise points are eliminated, and all holes are filled. The replacement function converges in 8 iterations. The in-between and final curves are also plotted in Fig. 9(c) for reference. The final seamless result is shown in Fig. 9(d).

8 Conclusion

We have described a novel and robust approach, luminance voting, to address the problem of global and local luminance alignment for image registration. Our method globally and locally adjusts luminance of two overlapping images, with-

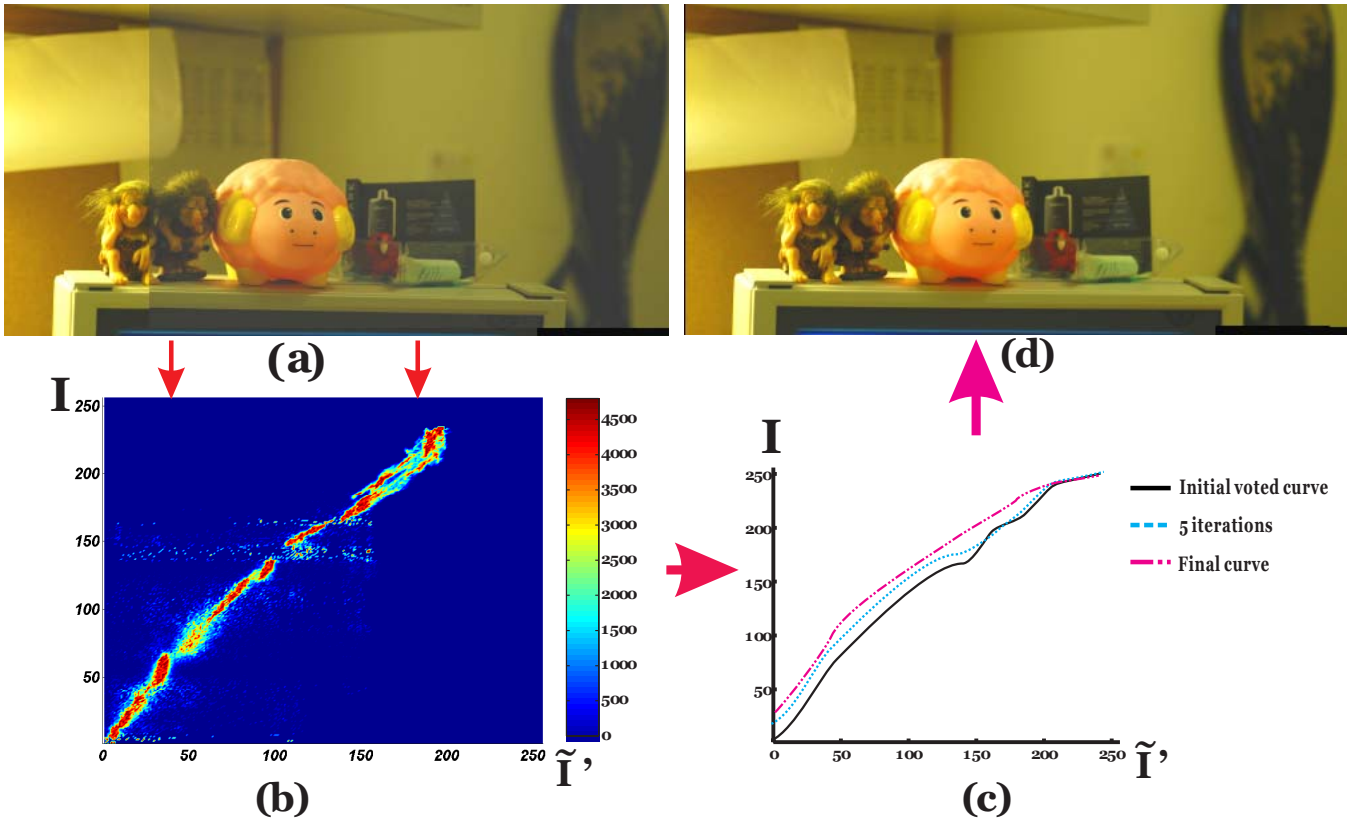


Figure 9: Luminance voting example. (a) Input mosaic with large exposure difference and some misregistration. (b) The luminance voting space (joint image space) is already very noisy and contains discontinuities. To demonstrate the robustness of our approach, we further add Gaussian noise of mean 0 and variance 5. (c) The global replacement curve refinement process. (d) The final seamless and globally aligned mosaic.

out assuming complex camera model or any simplifying 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, luminance voting is novel as it provides a fundamentally different approach to perform luminance 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. Our future work focuses on integrating the robust voting into the image registration process in the presence of moving objects, and on applying it to image-based applications.

References

- [1] P. J. Burt and E. H. Adelson. A multiresolution spline with applications to image mosaics. In *ACM Transaction on Graphics*, pages 2(4), 217–236, October 1983.
- [2] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. of SIGGRAPH 1997*, pages 369–378, 1997.
- [3] D. Hasler and S. Susstrunk. Colour handling in panoramic photography. In *Proc. of SPIE*, Jan 2001.
- [4] Jiaya Jia and C.-K. Tang. Luminance voting: image registration with global and local luminance alignment. Technical report, HKUST, 2003.
- [5] S. Mann and R. Mann. Quantigraphic imaging: Estimating the camera response and exposures from differently exposed images. In *CVPR 2001*, December 2001.
- [6] S. Mann and R. Piccard. Being ‘undigital’ with digital cameras: Extending the dynamic range by combining differently exposed pictures. In *Proc. Of ISTs 48th Annual Conference*, pages 422–428, May 1995.
- [7] G. Medioni, M.-S. Lee, and C.-K. Tang. A computational framework for feature extraction and segmentation. In *Elsevier*, 2000.
- [8] T. Mitsunaga and S. Nayar. Radiometric self calibration. In *CVPR 1999*, pages 374–380, 1999.
- [9] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE TPAMI*, 19(8):814–830, 1996.
- [10] Warren J. Smith. Modern optical engineering, 3rd ed. In *McGraw-Hill*, 2000.
- [11] R. Szeliski. Video mosaics for virtual environments. In *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.
- [12] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proc. of SIGGRAPH 1997*, pages 251–258, 1997.
- [13] Richard Szeliski and Heung-Yeung Shum. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.
- [14] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *CVPR 2001*, 2001.
- [15] V. Ramesh Y. Tsin and T. Kanade. Statistical calibration of ccd imaging process. In *ICCV 2001*, July 2001.
- [16] E. M. Riseman Z. Zhu and A. R. Hanson. Parallel-perspective stereo mosaics. In *ICCV 2001*, July 2001.