Robust Image-to-Image Color Transfer Using Optimal Inlier Maximization

Magnus Oskarsson
Centre for Mathematical Sciences, Lund University, Sweden
magnus.oskarsson@math.lth.se *

Abstract

In this paper we target the color transfer estimation problem, when we have pixel-to-pixel correspondences. We present a feature-based method, that robustly fits color transforms to data containing gross outliers. Our solution is based on an optimal inlier maximization algorithm that maximizes the number of inliers in polynomial time. We introduce a simple feature detector and descriptor based on the structure tensor that gives the means for reliable matching of the color distributions in two images. Using combinatorial methods from optimization theory and a number of new minimal solvers, we can enumerate all possible stationary points to the inlier maximization problem. In order for our method to be tractable we use a decoupling of the intensity and color direction for a given RGB-vector. This enables the intensity transformation and the color direction transformation to be handled separately. Our method gives results comparable to state-of-the-art methods in the presence of little outliers, and large improvement for moderate or large amounts of outliers in the data. The proposed method has been tested in a number of imaging applications.

1. Introduction and motivation

We will in this paper target the problem of robust color transfer between images. Color transfer, also known as color mapping, color correction or color balancing, is the problem of transferring the colors between two or more images so that they in some sense are the same. A nice introduction and overview to the problem is given in [9].

There are various scenarios and use cases for color transfer, but in this paper we are addressing the specific case when we have two images of the same scene, and where we have known pixel-to-pixel correspondences. We will describe an overall system for robust estimation of the color transformation, given two input images in the same geometric coordinate system. However, our robust fitting could also be directly used in any other feature based [32] or patch based [20] color transfer algorithm. For our feature based method, we will show how to optimally maximize the number of inliers, *i.e.* the number of correspondences that follow the fitted color transformation within some error bound. This gives a very robust approach that can handle large amounts of outliers. By outliers we mean data points that do not follow any underlying sought color transform, and that arise due to *e.g.* mismatches and occlusions. Previously, methods that optimally find models that maximize the number of inliers, in polynomial time, have been developed, [8]. The authors used it to produce algorithms for optimal image stitching and 2D-registration. In [44] similar methods were used to perform large-scale image-based localization. We will in this paper apply these ideas to color matching.

Our main contribution in this paper is *an optimal inlier maximization scheme that robustly fits color transformations to data (Section 3.2 and 3.3)*. In order use this method in a system, we also describe a feature detector and feature descriptor for color matching (Section 2), and how to choose the working color space in order to get separable color transformations (Section 3.1)[1].

1.1. Related work

There exists a number of approaches for transferring the colors of one image to another. Most of these approaches work on the overall color distribution of the images [37, 33, 11, 23, 10, 30, 45, 16, 18] . One main reason is that these methods also work on images that depict different scenes entirely, *i.e.* where there is no obvious spatial relation between the images. However, this can also be an impediment, especially when such spatial correspondences exist. To this end, methods based on segmentation [42, 46], and matching patches in the corresponding images have been developed, [20]. This, and many such feature based methods, also estimate a geometric transformation between the images. In this paper we restrict ourselves to the case when this transformation is known, or when the images are naturally in the same geometric coordinate system (in our

* Supported by the strategic research project eSSENCE and the Horizon 2020 project 876019 (ADACORSA) funded by EU and Vinnova.

[1] Code at: *github.com/hamburgerlady/antifeature-color-transform*

experiments we show a number of natural use cases where this is true). Other feature based methods [32, 48, 46, 31] typically match SIFT features, and then fit the transformation to colors extracted at these points. Most of these methods are not robust to outliers to any large extent, however, the method of Park *et al*. [32] is based on a robust low rank matrix formulation. Recently also methods based on learning, for so-called style transfer, have shown impressive results [50, 28], but these methods typically do not consider outliers.

2. Feature representation

We will in this section describe our feature representation. Most feature based methods for color transfer use SIFT features, [32, 31, 46, 48], mostly since these methods also do geometric matching. However, such feature points are by design points with high frequency structures. This is not ideal for modelling color, since color by nature is a low frequency property. For this reason, many color transfer algorithms extract the color by some form of averaging over a larger window around the SIFT points, [31, 46, 48]. We will base our representation on the structure tensor [5, 25, 24], but it could be based on other detectors, *e.g.* MSER [29]. The structure tensor, as its name suggests, captures much of the local intensity variations in a compact and robust way. These properties make it very suitable as a tool for feature extraction, and the classic Foerstner-Harris corner detectors [15, 21] were based on it. It is well known that the magnitudes of the eigenvalues of the structure tensor $J(x, y)$ capture the local structure at (x, y), so that typically corners have two large eigenvalues, linear structures and edges have one large and one small eigenvalue and slowly varying areas have two small eigenvalues. These properties have been used to find corners in images, by thresholding *e.g.* the determinant (which equals the product of the eigenvalues) of $J(x, y)$. In our case we are interested in feature point that have as little corner or edge structures as possible, and as such they would correspond to *antifeatures* to what we normally extract as features. We could do this by looking at the product of the eigenvalues, and when this is small it constitutes a good candidate for a feature. It turns out that we get better behavior by looking at the logarithm of the product (this will among other things suppress lines and edge structures), so we will use this as a basic function from which we extract our features,

$$\varphi(x, y) = \log(1 + \det J(x, y)). \tag{1}$$

We would like φ to be small at our feature points, so we will threshold it at some level. We choose this level as a factor of the mean $\bar{\varphi}$ of φ, over the whole image. We then choose all points (x, y) that are also local minimizers of φ, *i.e.* we

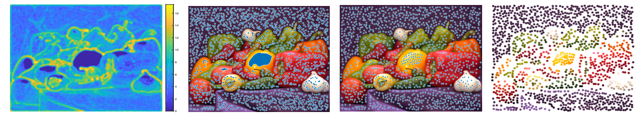

Figure 1. Left shows the response function $\varphi(x, y)$ as in (1). Then follows the initial feature set A, the final feature set A' after lateral inhibition and the descriptors for the extracted features.

extract our feature set A as $A = A_1 \cap A_2$, with

$$\begin{aligned} A_1 =& \{(x, y) \,|\, \varphi(x, y) \leq C\bar{\varphi}\}, \\ A_2 =& \{(x, y) \,|\, \varphi(x, y) \leq \varphi(x', y') \,\forall\, |(x', y') - (x, y)| < \epsilon\}. \end{aligned} \tag{2}$$

In most cases we aim at features that cover the whole of the image, in order to capture the color variations, but at the same time we do not want too dense sampling. To this end we apply a simple lateral inhibition mechanism [36], by ordering the features randomly and cancelling out a neighborhood around each feature taken in the order. This gives us our final feature set A'. In Figure 1 the result of our feature extraction scheme is shown on an example image. We choose the locally averaged color as our descriptor $\mathbf{d}$ for a feature position (x_i, y_i), by

$$\mathbf{d}(x_i, y_i) = I(x, y) * G(x, y)|_{(x_i, y_i)}, \tag{3}$$

where $G(x, y)$ is a smoothing Gaussian function. To the right in Figure 1 the resulting descriptor colors are shown for the example image.

3. Color transformation inlier maximization

In the presence of outliers, finding accurate correspondences is difficult, and robust methods are highly desirable. Typically, we have a model that depends on a number of parameters Θ, and we would like to fit this model to our data in a robust way, such as

$$\min_{\Theta} \sum_i \ell(r_i), \tag{4}$$

where r_i is the residual for data point i, and $\ell(x)$ is a robust loss function. Here we choose to minimize the number of outliers (or equivalently to maximize the number of inliers), and hence define

$$\ell(r) = \begin{cases} 0 & \text{if } r \leq \epsilon, \\ 1 & \text{otherwise}, \end{cases} \tag{5}$$

for some inlier bound ϵ. This formulation leads to a challenging optimization problem, but we will now show how we can find a tractable solution. In [8] it was shown how the number of inliers can be maximized in polynomial time, for a fixed-dimensional model, where the computational complexity follows directly as a consequence of the theory of optimization. This is done by rewriting (4) using a dummy function f[2],

$$\min_{\Theta} f(\Theta) \tag{6}$$

$$s_i g_i(\Theta) \leq 0 \quad i = 1, \ldots, n, \tag{7}$$

$$h_j(\Theta) = 0 \quad j = 1, \ldots, m - k. \tag{8}$$

Here $g_i(\Theta) \leq 0$ is a polynomial expression in Θ equivalent to $r_i \leq \epsilon$, $h_j(\Theta)$ are (polynomial) embedding constraints for the parameters Θ and

$$s_i = \begin{cases} 1 & \text{if data point } i \text{ is an inlier,} \\ -1 & \text{otherwise.} \end{cases} \tag{9}$$

The $k-$dimensional parameter space should be a differentiable manifold embedded in $\mathbb{R}^m$ with a set of $(m - k)$ equality constraints h_j and $f(\Theta)$ should be a polynomial. The main theorem from [8] shows that one can find the optimal solution with respect to the number of inliers by enumerating a finite set of so called *critical points*, essentially being the Karush-Kuhn-Tucker (KKT) points. These critical points divide the solution space into regions that contain different combinations of inliers and outliers, and the optimal solution with respect to the number of inliers will be found in one of the critical points. The critical points are found by enumerating all possible subsets of data points of sizes less or equal to the dimension (k) of the parameter space. For all subsets of data points B of size $|B| = k$ the critical points Θ^* are given as solutions to

$$g_i(\Theta^*) = 0, \quad i \in B. \tag{10}$$

For subsets B of size $|B| < k$ the critical points fulfill (10), and in addition that

$$\{\nabla g_i(\Theta^*), i \in B\} \cup \{\nabla f(\Theta*), \nabla h_1 \Theta^*, \ldots\} \tag{11}$$

should be a linearly dependent set. For all possible critical points, we check how many inliers we get to our problem, and the optimal solution will be the among these solutions. For a $k-$dimensional parameter space this leads to an $\mathcal{O}(n^{k+1})$ algorithm.

3.1. Problem formulation

We are now ready to formulate and solve the main optimization problem addressed in this paper. Our assumption is that we have two sets of (geometrically aligned) corresponding feature points ($\{\mathbf{f}_i\}$ and $\{\mathbf{f}'_i\}$) whose descriptors ($\{\mathbf{d}_i\}$ and $\{\mathbf{d}'_i\}$) each characterize the color distribution of their respective images. We assume that there are outliers in these correspondences, due to *e.g.* misalignment, saturation or occlusions. We would now like to find a transformation T on the descriptors so that $T(\mathbf{d}_i) \approx \mathbf{d}'_i$ for all or most i. In robust estimation one key to a tractable system is for the model to be low-dimensional, in order to avoid combinatorial explosion. So, we would like to define our transformation T with as few parameters as possible, but so that it still captures most of the typical color transformations that occur naturally. There are a number of different color spaces that have been described in the literature, *e.g.* the HSI family or the CIE L*a*b* [34], with different properties and uses. Many of these convert the RGB channels, and separate them into one channel that captures the intensity values (the intensity or luminance) and two channels that capture the color or chromaticity distribution in some way (*e.g.* into hue and saturation). For color transfer based on histograms, one usually wants to do this separately on each channel. In this case it is important to use a color space that separates the color channels into decorrelated (and if possible independent) color channels [43, 35]. For an investigation into how different color spaces influence the color transfer problem see [39]. At first glance it would seem sensible to work in a perceptually modeled space such as L$\alpha\beta$ [41] or CIE L*a*b*. However, since we in any case need to model non-linearities in our method, moving back and forth to such a space typically only adds computational overhead. For this reason, we will define our separation in a simpler way, and base it on separating intensity and color in a specific way. Similar separations have also been used previously [27]. For a given descriptor vector $\mathbf{d}$ (which is a 3-vector defining an RGB-color), we define the intensity v as

$$v(\mathbf{d}) = |\mathbf{d}|. \tag{12}$$

Note that this is a non-linear function of $\mathbf{d}$ as opposed to many other approaches where the luminance channel is defined as a linear combination of the RGB-vector. The motivation for defining our intensity as the length of $\mathbf{d}$ is that we can then define the color $\mathbf{c}$ as

$$\mathbf{c}(\mathbf{d}) = \frac{\mathbf{d}}{|\mathbf{d}|} = \frac{\mathbf{d}}{v(\mathbf{d})} \Leftrightarrow \mathbf{d} = v(\mathbf{d})\mathbf{c}(\mathbf{d}). \tag{13}$$

This means that all color vectors $\mathbf{c}$ have length one. We will now work with transformations on $\mathbf{d}$ that consist of a *non-linear* scaling, S (defined on a vector $\mathbf{x}$ as $S(\mathbf{x}) = S(|\mathbf{x}|)\mathbf{x}/|\mathbf{x}|$), followed by a rotation R. So, for a given descriptor $\mathbf{d}$ we have

$$T(\mathbf{d}) = R(S(\mathbf{d})) = R(S(v \cdot \mathbf{c})) = R(S(v)\mathbf{c}) = S(v)R(\mathbf{c}), \tag{14}$$

since the rotation only acts on the direction of the vector,

$$R(\mathbf{d}) = R(v \cdot \mathbf{c}) = vR(\mathbf{c}), \tag{15}$$

and the length is preserved after rotation,

$$|R(\mathbf{d})| = |\mathbf{d}| = |v \cdot \mathbf{c}| = |v||\mathbf{c}| = |v| = v. \tag{16}$$

[2]The function f is used to move the loss function to the constraints, and can be chosen arbitrarily (simple) as long as we get a finite number of critical points

One can also easily verify that the order isn't important, *i.e.* $T(\mathbf{d}) = R(S(\mathbf{d})) = S(R(\mathbf{d})) = S(v)R(\mathbf{c})$. Viewing the color of a pixel as a direction in RGB-space has (albeit without the unit-length constraint) been used in color constancy and illumination estimation applications, [12, 2, 13], with homographies as transformation class.

3.2. Robust intensity estimation

We will now show how inlier maximization can be applied to intensity transformations. For our optimization to be tractable we need our model to be relatively low-dimensional, but at the same time be able to model real intensity transformations, including non-linear exposure and gamma correction. We assume that our color spaces have been affinely mapped to lie in the interval $[0, 1]$, so for this reason we let our non-linear scaling map the origin to the origin. Transformations based on polynomials have previously been used for color calibration and illumination estimation [22, 14, 1]. We have experimented with a number of different polynomial scaling functions, and have chosen to work with third-degree polynomials (passing through the origin),

$$S(v) = p_{abc}(v) = av^3 + bv^2 + cv. \tag{17}$$

These are not necessarily monotone in v, but we will enforce this constraint by discarding solutions that do not fulfill monotony. Given two sets of corresponding feature points with intensities ($\{v_i\}$ and $\{v'_i\}$) and an inlier bound ϵ_v we would like to solve

$$\min_{a,b,c} \sum_i \ell(D(p_{abc}, (v_i, v'_i)), \tag{18}$$

$$\ell(r) = \begin{cases} 0 & \text{if } r \leq \epsilon_v, \\ 1 & \text{otherwise}, \end{cases} \tag{19}$$

where D is the distance from the polynomial to the point. There is no closed form solution for the smallest distance from a point to a polynomial curve. Here we will use an approximate distance, based on the linearization of the polynomial curve at the given point. We then define the distance from the polynomial to the point as the distance from the point to the tangent of the polynomial curve at the $x-$coordinate of the point, *i.e.* given a point (x_0, y_0) and a polynomial p_{abc} the distance is given by

$$D(p_{abc}, (x_0, y_0)) = \frac{|p_{abc}(x_0) - y_0|}{\sqrt{1 + (\frac{dp_{abc}}{dx}(x_0))^2}}. \tag{20}$$

We now solve (18) by reformulating it in the form of (6) and go through all critical points. In order to do this, we define our dummy function as $f(a, b, c) = a$. The dummy function can be chosen arbitrarily, as long as it gives a finite number of critical points, *i.e.* that there are a discrete number of critical points as opposed to an infinite number. In this case h_j is empty since we have no additional constraints on our parameters (a, b, c). As building blocks for finding the critical points we need polynomial solvers for the cases when three, two and one inlier constraints are fulfilled exactly respectively. Here, only the first two cases turn out to have non-empty solution sets. We will start with the three-point solver. Given three corresponding intensity measurements $\{(v_1, v'_1), (v_2, v'_2), (v_3, v'_3)\}$ we would like to find a third-degree polynomial p_{abc} going through the origin so that

$$D(p_{abc}, (v_i, v'_i)) = \epsilon_v, \quad i = 1, 2, 3. \tag{21}$$

This can be reformulated as three polynomial constraints by multiplying with the denominator in (20) and then squaring each side of the expression. This gives three polynomials g_i, in three variables a, b and c of total degree two (see supplemental material for the derivation details). This system of polynomial equations can be solved in a number of different ways. We choose to use the automatic generator from [26] to produce a Matlab solver based on the action matrix method. This gives us a solver with in general eight solutions with an elimination template of size 26×34. The Matlab implementation runs in $20\mu s$ on a 2.5 GHz Intel Core i7 MacBook Pro.

The second sub-problem that we need to solve is when two constraints are active and when the gradient of the goal function is linearly dependent with the gradients of the constraints, so that

$$g_i = 0, \quad i = 1, 2, \tag{22}$$

$$\det([\nabla f \ \nabla g_1 \ \nabla g_2]) = 0. \tag{23}$$

Since the gradient of the goal function is constant, and the gradients of the constraints g_i are linear in the variables, the determinant constraint will be a second-degree polynomial in the parameters (a, b, c). This means that we end up with the same type of equations as for the three-point case, and we can use the same minimal solver for the two-point case. We now find the best model by going through all combinations of three and two points. This leads in this case to a total complexity of $\mathcal{O}(n^4)$ for n correspondences.

3.3. Robust rotation estimation

For the rotation estimation we assume that we have n corresponding unit length color directions ($\{\mathbf{c}_i\}$ and $\{\mathbf{c}'_i\}$) and an inlier bound ϵ_c. We would then like to find a rotation matrix R so that

$$\min_R \sum_i \ell(|R\mathbf{c}_i - \mathbf{c}'_i|), \tag{24}$$

$$\ell(x) = \begin{cases} 0 & \text{if } x \leq \epsilon_c, \\ 1 & \text{otherwise}. \end{cases} \tag{25}$$

We will represent our rotations using quaternions $\mathbf{q} = (q_0,\, q_1,\, q_2,\, q_3)$. If $|\mathbf{q}| = 1$ we can transform our quaternion to a rotation matrix $R = Q(\mathbf{q})$. However since we would like to use as few model parameters as possible we will use the Cayley-Gibbs-Rodrigues formulation [3] to scale our quaternion so that $q_0 = 1$, and hence we now only need to use three parameters (q_1, q_2, q_3) to model our space[3]. The problem is that now $Q(\mathbf{q})$ is no longer a valid rotation matrix. We can solve this by scaling it with the inverse of the squared norm of $\mathbf{q}$ so that

$$R(\mathbf{q}) = \frac{Q(\mathbf{q})}{\mathbf{q}^T\mathbf{q}}. \tag{26}$$

Similarly, as for the intensity transformation estimation, we need basic solvers for a number of cases, in order to find all critical points. We have now embedded our parameter space in $\mathbb{R}^3$, so the dimension is three, *i.e.* we have $h = q_0 - 1$ in (6). We thus need to investigate cases when up to three constraints are active. We will start by expressing the residual constraint g_i for one correspondence. For each i we have

$$|R\mathbf{c}_i - \mathbf{c}'_i|^2 = |\mathbf{c}_i|^2 + |\mathbf{c}'_i|^2 - 2(\mathbf{c}'_i)^T R\mathbf{c}_i. \tag{27}$$

Now using our Cayley representation for the rotation, and using (27) we get for an active constraint

$$\begin{aligned} |\mathbf{c}_i|^2 + |\mathbf{c}'_i|^2 - 2(\mathbf{c}'_i)^T \frac{Q(\mathbf{q})}{\mathbf{q}^T\mathbf{q}}\mathbf{c}_i &= \epsilon_c^2 \Leftrightarrow \qquad (28)\\ 2(\mathbf{c}'_i)^T Q(\mathbf{q})\mathbf{c}_i + (\epsilon_c^2 - |\mathbf{c}_i|^2 - |\mathbf{c}'_i|^2)\mathbf{q}^T\mathbf{q} &= 0 \equiv g_i(\mathbf{q}). \end{aligned}$$

Here $Q(\mathbf{q})$ is quadratic in $\mathbf{q}$ so each $g_i(\mathbf{q})$ is a quadratic expression in $\mathbf{q}$. For three active constraint we get three equations in the three parameters. These are three full second-degree polynomials so we get exactly the same problem as for the three-points intensity solver, so we can use the same solver but with different structure in the input coefficients.

When only two constraints are active, we need to use our defined dummy function, which in this case we choose as $f(\mathbf{q}) = q_1$. We again have a linear goal function which means that we also in this case get the same structure as for the two-points intensity solver, and again we get three full second-degree polynomials in the three variables. We can again use the same minimal solver. For fewer active constraints than two, the solution sets are empty.

We now find the best rotation by going through all combinations of three and two points. This leads in this case to a total complexity of $\mathcal{O}(n^4)$ for n correspondences.

3.4. System overview

In order for the methods in the previous sections to be tractable we needed low dimensional parametric spaces. However, from these methods we do not only get the actual transformations, we also get robust estimates of the inlier outlier partition of the correspondences. This means that we can use the estimated inliers to further refine our model. We propose a simple extension of our model by relaxing our rotation registration to an affine registration. We assume that we now have transformed corresponding feature pairs $(\mathbf{d}_i, \mathbf{d}'_i)$ for all i that are deemed inliers from the previous estimation. We then affinely map our descriptors in a least squares manner, *i.e.* we find A and $\mathbf{b}$ that solve

$$\min_{A,\mathbf{b}} \sum_i |A\mathbf{d}_i + \mathbf{b} - \mathbf{d}'_i|_2^2. \tag{29}$$

This is a linear least squares problem which can be solved in closed form using standard methods. In addition to this we use a sub-sampling scheme of our initial feature set. In all experiments and evaluation, we use maximum 50 correspondences in our robust method, for speed-up. Based on the initial transformation we use the inlier set of all feature points for the final affine refinement. See supplemental material for model comparison using different number of correspondences and different transformation classes (affine, projective or higher order polynomial models).

4. Performance evaluation

In this section we will investigate how well our methods work in a controlled setting, in order to evaluate the performance in terms of accuracy and robustness. We will also compare our method with a number of other methods, on several benchmark datasets. Concerning the total running time of our algorithm, it depends heavily on the number of input correspondences. As an example if we have 50 correspondences, the time of going through all triplets of points for the intensity estimation would be #triplets $\cdot$ solver time $= 19600 \cdot 20\mu s \approx 0.4s$. For larger images, time will also be spent in the actual transforming of the image (this is true for any algorithm). See Table 2 for comparison of complete execution times on the experiments. We have not explicitly compared to RANSAC. Our proposed solvers can be directly used as minimal solvers in a RANSAC framework (by setting $\epsilon = 0$). However, there is no gain here since the complexity will be the same (we still need to sample three points, and RANSAC will never guarantee an optimal solution, even if we do exhaustive sampling [8]). A better option if runtime is critical is to use our framework with an early bailout option. Higher order models would require many more point correspondences in RANSAC, such as *e.g.* 12 for an affine model [31], and is not tractable in our setting.

4.1. Numerical stability of solvers

We will start by investigating the numerical stability of our minimal solvers. We will focus on the two three-point

[3] Note that we cannot represent rotations that correspond to $q_0 = 0$, and we use the standard way of handling this by applying a random (known) rotation before processing.

Table 1. Benchmark result on 40 synthetically color transformed images with perfect geometric alignment from [47]. The results are given as means of PSNR, SSIM, and the trained metric LPIPS[49] over all images, where each row represents different fractions of outliers.

	CS (dB)						SSIM						LPIPS					
Inl.	[33]	[17]	[11]	[30]	[32]	*Our*	[33]	[17]	[11]	[30]	[32]	*Our*	[33]	[17]	[11]	[30]	[32]	*Our*
1.0	**42.4**	38.9	36.5	25.3	33.7	36.5	**98**	97	97	86	96	97	**0.04**	**0.04**	0.05	0.13	0.05	0.05
0.8	20.4	25.2	21.2	21.8	33.2	**35.7**	80	93	83	84	96	**97**	0.32	0.06	0.25	0.16	**0.05**	**0.05**
0.6	16.7	19.8	17.7	19.3	27.8	**35.1**	63	83	70	80	91	**96**	0.45	0.12	0.33	0.18	0.12	**0.05**
0.4	14.5	16.4	15.5	17.4	20.7	**34.5**	48	66	58	77	81	**96**	0.53	0.23	0.39	0.21	0.24	**0.06**
0.2	12.9	14.0	13.8	15.9	17.1	**30.5**	37	42	48	73	71	**92**	0.60	0.43	0.44	0.24	0.35	**0.08**

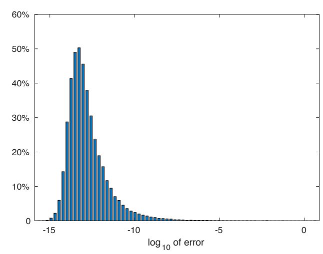

Figure 2. Histogram over the logarithm of the errors for a large number (10,000) of problem instances, given synthetic data with no noise. It shows the performance of our three-point rotation solver.

solvers used in the intensity and rotation estimation respectively. At the core of these minimal solvers we have the same polynomial problem, three full multivariate polynomials in three variable of total degree two. The used solvers have the same template size so here we could have used the exact same solvers. However, since we have different structure in how we set up the problem in terms of what we input and what we estimate, we opted to generate two different solvers. In Fig. 2 the histogram over the logarithms of the absolute errors from 10,000 random instance problems is shown, for our three-point rotation solver. We get essentially the same numerical characteristics for our other solver.

4.2. Benchmark on image color matching

In order to compare our method against other color transformation methods we tested our method on a standard benchmark set. The dataset is described in [47] and contains pairs of images, a target and a source image. The source image has been synthetically transformed from the target image, and the objective is to transform the source image so that it fits the target image. Examples of input and output can be seen in Fig. 3. The images are perfectly geometrically aligned, but no information on the color transformation is provided. To run our method, we detect features in the source image, and then extract descriptors at the same feature point positions in both images, using the method described in section 2. We then robustly fit an intensity transformation and a rotation using our optimal inlier estimation. The inlier set is then used to find an affine transformation. The total transformation (intensity, rotation and affine) is then applied to the whole source image to produce our candidate target image. To test the robustness of our method with respect to outliers we simulated gross outliers by randomly changing the color of a certain percentage of the target image. We then estimated the transform, and evaluated the result using the original uncorrupted target image. The evaluation is based on the PSNR-values, the

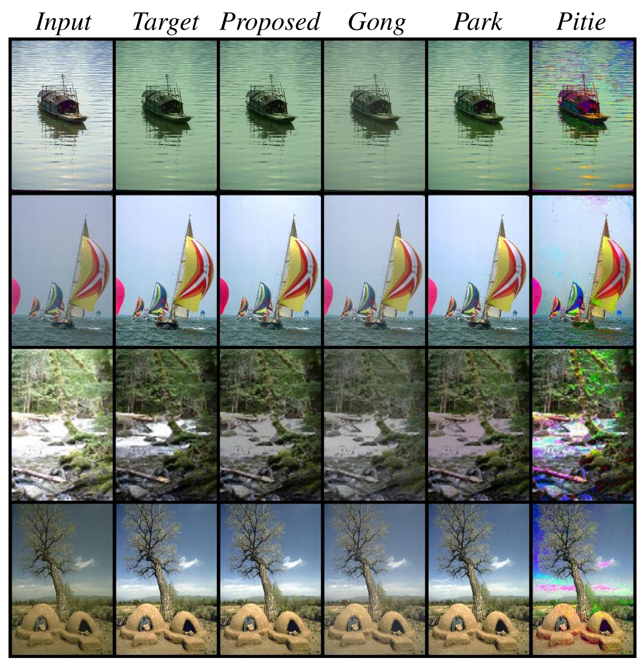

Figure 3. Example results on the synthetic color matching dataset [47] with 20% added outliers. Left column shows the input images, second column shows the target images, and then follows the transformed images using the proposed and compared methods. See supplemental material for more results.

Table 2. Mean execution time per image for benchmark tests.

	[33]	[17]	[11]	[30]	[32]	*Our*
Synthetic (s)	1.4	3.3	0.62	0.19	1.6^*	1.1
Middlebury (s)	5.8	7.0	2.1	0.49	3.6^*	1.9
TUT-INTEL (s)	11.5	5.3	3.8	0.97	6.4^*	3.1

*The timings for [32] do *not* include feature extraction and matching.

Figure 4. Example results on the INTEL-TUT dataset [4], with 20% added outliers. Left column shows the RAW input images, second column shows the target JPG images, and then follows the transformed images using the proposed and the compared methods [33] and [32]. See supplemental material for more results.

structure similarity index and the trained metric LPIPS[49], between the transformed images and the target images. We compare our method to a number of other methods for color transformation estimation; the methods of Pitie *et al.* [33], Gong *et al.* [17], Fecker *et al.* [11], Park *et al.* [32] and Nikolova and Steidl [30]. The results can be seen in Table 1 and Fig. 3 (20% outliers), where we have also included the results from the other methods. We test going up to having 80% outliers. Although this would seem unlikely in a real application, we wanted to show the breaking point of our algorithm as well as show that the results degrade gracefully up to this point. A number of compared methods will perform slightly better when there are no outliers, but the differences are very small. Higher order methods will have a tendency to give artifacts, for even small outlier rates. See also supplemental material for additional tests on model size. In order to further test our method on a larger set of real images, we have used two additional datasets for illumination estimation and color constancy, namely the Middlebury color dataset [6] and the INTEL-TUT dataset [4]. The first dataset contains 85 registered RAW/JPG pairs of natural scenes taken with 12 different camera models. The second dataset contains around 2000 raw images taken with three different cameras. It also contains color corrected JPG images intended for illustration of how a color corrected image should look like. However, we can use these images to benchmark our color transfer method against previous methods. We follow the procedure from the previous section, and the results can be seen in Table 3 for the Middlebury dataset. The results on the far more challenging INTEL-TUT dataset are shown in Table 3 and Fig. 4. The corresponding distributions of LPIPS[49] over all images can be seen in Fig. 5. The resulting distributions for SSIM and PSNR are given in the supplemental material, and show similar statistics. The conclusion is that, even for small outlier levels, previous methods degrade unfavorably.

5. Example applications

Our emphasis in this paper is the presentation of an optimal inlier maximization algorithm for color transfer estimation. We will in this section showcase how our method can be incorporated as a building block in different applications. We want to show that the handling of outliers is practical in many systems, but we do not claim that the following simple implementations are necessarily beyond the current state-of-the-art within the specific applications. Additional results can be found in the supplemental material.

Image stitching The process described in the previous section can be directly applied also to image stitching. Misalignments and moving objects make robust methods desirable. We use standard methods for the geometric alignment. We then run our feature detector on the overlapping region, and extract our color descriptor from the two images. We run our robust color matching, and use the estimated transform on the whole image. This gives us our two images now also hopefully in the same color coordinate system. Often some form of blending function is used to avoid borders between images in the stitching, but here we don't use any blending. For overlapping pixels, we simply choose the pixel-wise max. The result on an example can be seen in Fig. 6. To the left is the stitched images without the estimated color transform, and to the right using the estimated color transform. In this case get a very seamless stitching result. See the supplemental material for additional results.

HDR estimation from multiple exposure LDR brackets Next example application is how to estimate a high dynamic range (HDR) image from several low dynamic range (LDR) images [40, 7, 19, 38]. We have applied our method to blind bracketing, when the exposure settings for the input images are unknown. We will need to handle problems with misalignment, moving objects and severely over and under exposed parts of the images, so this motivates the need for robust methods. Please see the supplemental material for results and details on how we applied our method.

Color transfer for non rigid motion In order to see if our method could also be used for geometrically unaligned images, we compared our method with the feature based robust method of Park *et al.* [32]. We use their sift based method for extracting matching patches. We then replaced their color matching method with our proposed method. On each tentative patch pair we ran our feature detector and extractor (typically this gives two to four feature points for a 32×32 patch). We then ran our robust color estimation and applied the transform on the whole input image. The results for two example images are shown in Fig.7. Left shows the input images, and the second column images show the

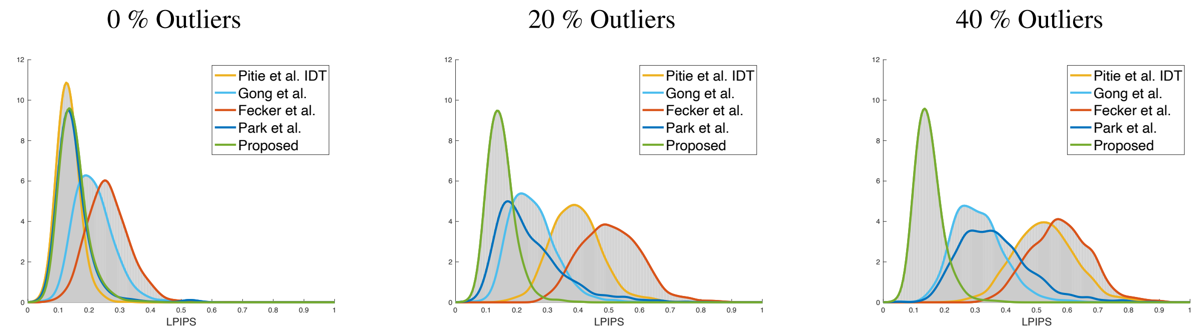

Figure 5. Distribution of LPIPS[49] on the INTEL-TUT dataset [4], for varying outlier levels. See suppl. mtrl for SSIM and PSNR values.

Table 3. Benchmark result on the Middlebury dataset [6] (top) and the INTEL-TUT dataset [4] (bottom). The results are given as means of PSNR, SSIM, and the trained metric LPIPS[49] over all images, where each row represents different fractions of outliers.

	CS (dB)						SSIM						LPIPS					
Inl.	[33]	[17]	[11]	[30]	[32]	*Our*	[33]	[17]	[11]	[30]	[32]	*Our*	[33]	[17]	[11]	[30]	[32]	*Our*
1.0	**35.7**	28.0	29.7	23.0	27.4	32.2	**98**	89	92	82	92	96	**0.03**	0.13	0.10	0.15	0.08	0.05
0.8	20.9	23.2	19.9	21.3	25.6	**32.2**	79	82	72	81	89	**96**	0.28	0.17	0.27	0.16	0.10	**0.05**
0.6	17.1	18.9	16.7	19.6	18.7	**31.8**	60	73	57	79	78	**96**	0.39	0.24	0.35	0.18	0.21	**0.05**
1.0	**29.4**	27.8	24.7	17.0	28.0	29.0	**88**	84	69	34	85	86	**0.13**	0.22	0.26	0.43	0.15	0.15
0.8	19.7	24.2	16.0	16.4	24.1	**28.9**	63	80	47	34	74	**86**	0.39	0.25	0.51	0.43	0.24	**0.15**
0.6	16.4	20.2	13.9	15.6	19.0	**28.9**	44	71	36	34	65	**86**	0.53	0.31	0.58	0.44	0.36	**0.15**

Figure 6. The result of stitching two images, with or without the proposed color transformation. Left: Result after pointwise max at each pixel. Middle: The same result but with the proposed color transformation on the second image. No additional blending was used. See supplemental material for more details and examples.

target color space. The third column images are the result from Park *et al.* and the rightmost images are the results from our proposed method.

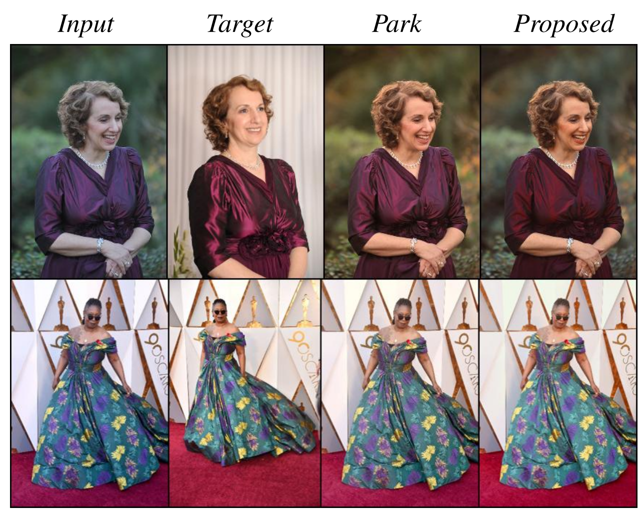

Figure 7. Non rigid color transformation. Left shows input images, second column shows target color space. Then follows the method of Park *et al.* [32] and the proposed method run on the tentative patch matches from Park *et al.* The results are similar, but the proposed method arguably matches the colors slightly better.

6. Conclusion

We have in this paper introduced an optimal inlier maximization algorithm for robust feature-based color transfer estimation. Using methods from optimization theory and a number of new minimal solvers we can enumerate all possible stationary points to the inlier maximization problem. In order for our method to be tractable we use decoupling of the intensity and color direction for a given RGB-vector, enabling separation of the intensity transformation and the color direction transformation. We have shown that our method gives results comparable to state-of-the-art methods in the presence of no outliers in the data, and that it outperforms other methods for moderate or large amounts of outliers in the data. An additional benefit is that the robust sampling is an efficient way to represent the color distribution, without needing to optimize over all pixels.

References

[1] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1535–1544, 2019. 4

[2] Mahmoud Afifi, Abhijith Punnappurath, Graham Finlayson, and Michael S Brown. As-projective-as-possible bias correction for illumination estimation algorithms. *JOSA A*, 36(1):71–78, 2019. 4

[3] Simon L. Altmann. *Rotations, quaternions and double groups*. Clarendon Press, Oxford, 1986. 5

[4] Caglar Aytekin, Jarno Nikkanen, and Moncef Gabbouj. Intel-tut dataset for camera invariant color constancy research. *arXiv preprint arXiv:1703.09778*, 2017. 7, 8

[5] Josef Bigun and Gösta Granlund. Optimal orientation detection of linear symmetry. In *IEEE International Conference on Computer Vision (ICCV)*, 1987. 2

[6] Ayan Chakrabarti, Daniel Scharstein, and Todd E Zickler. An empirical camera model for internet color vision. In *BMVC*, volume 1, page 4. Citeseer, 2009. 7, 8

[7] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, page 31. ACM, 2008. 7

[8] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Tractable algorithms for robust model estimation. *Int. Journal of Computer Vision*, 112(1):115–129, 2015. 1, 2, 3, 5

[9] Hasan Sheikh Faridul, Tania Pouli, Christel Chamaret, Jürgen Stauder, Alain Trémeau, Erik Reinhard, et al. A survey of color mapping and its applications. *Eurographics (State of the Art Reports)*, 3:2, 2014. 1

[10] Hasan Sheikh Faridul, Jürgen Stauder, and Alain Trémeau. Illumination and device invariant image stitching. In *IEEE International Conference on Image Processing (ICIP)*, pages 56–60, 2014. 1

[11] Ulrich Fecker, Marcus Barkowsky, and André Kaup. Histogram-based prefiltering for luminance and chrominance compensation of multiview video. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1258–1267, 2008. 1, 6, 7, 8

[12] Graham Finlayson, Han Gong, and Robert B Fisher. Color homography: theory and applications. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):20–33, 2017. 4

[13] Graham D Finlayson. Colour and illumination in computer vision. *Interface focus*, 8(4):20180008, 2018. 4

[14] Graham D Finlayson, Michal Mackiewicz, and Anya Hurlbert. Color correction using root-polynomial regression. *IEEE Transactions on Image Processing*, 24(5):1460–1470, 2015. 4

[15] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986. 2

[16] Oriel Frigo, Neus Sabater, Vincent Demoulin, and Pierre Hellier. Optimal transportation for example-guided color transfer. In *Asian Conference on Computer Vision*, pages 655–670. Springer, 2014. 1

[17] Han Gong, Graham D Finlayson, Robert B Fisher, and Fufu Fang. 3d color homography model for photo-realistic color transfer re-coding. *The Visual Computer*, 35(3):323–333, 2019. 6, 7, 8

[18] Mairead Grogan and Rozenn Dahyot. L2 divergence for robust colour transfer. *Computer Vision and Image Understanding*, 181:39–49, 2019. 1

[19] Michael D Grossberg and Shree K Nayar. High dynamic range from multiple images: Which exposures to combine. In *Proc. ICCV Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*, 2003. 7

[20] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *ACM transactions on graphics (TOG)*, volume 30, page 70. ACM, 2011. 1

[21] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988. 2

[22] GW Hong, MR Luo, and PA Rhodes. A study of digital camera colorimetric characterization based on polynomial modeling. *Color Research and Application*, 26(1):76–84, 2001. 4

[23] Youngbae Hwang, Joon-Young Lee, In So Kweon, and Seon Joo Kim. Color transfer using probabilistic moving least squares. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[24] B. Jähne. *Spatio-temporal image processing*. Springer, 1993. 2

[25] Hans Knutsson. Representing local structure using tensors. In *Scandinavian Conference on Image Analysis, Oulu, Finland*, 1989. 2

[26] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[27] Joon-Young Lee, Kalyan Sunkavalli, Zhe Lin, Xiaohui Shen, and In So Kweon. Automatic content-aware color and tone stylization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2470–2478, 2016. 3

[28] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[29] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conf. (BMVC)*, 2002. 2

[30] Mila Nikolova and Gabriele Steidl. Fast ordering algorithm for exact histogram specification. *IEEE Transactions on Image Processing*, 23(12):5274–5283, 2014. 1, 6, 7, 8

[31] Thomas Oskam, Alexander Hornung, Robert W Sumner, and Markus Gross. Fast and stable color balancing for images and augmented reality. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 49–56. IEEE, 2012. 2, 5

[32] Jaesik Park, Yu-Wing Tai, Sudipta N Sinha, and In So Kweon. Efficient and robust color consistency for community photo collections. In *Proceedings of the IEEE Con-

ference on Computer Vision and Pattern Recognition, pages 430–438, 2016. 1, 2, 6, 7, 8

[33] Francois Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107(1-2):123–137, 2007. 1, 6, 7, 8

[34] Konstantinos N Plataniotis and Anastasios N Venetsanopoulos. *Color image processing and applications*. Springer Science & Business Media, 2013. 3

[35] Tania Pouli, Douglas W Cunningham, and Erik Reinhard. Image statistics and their applications in computer graphics. In *Eurographics (STARs)*, pages 83–112, 2010. 3

[36] Mahendra S Rao and Marcus Jacobson. *Developmental neurobiology*. Springer Science & Business Media, 2006. 2

[37] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 1

[38] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010. 7

[39] Erik Reinhard and Tania Pouli. Colour spaces for colour transfer. In *International Workshop on Computational Color Imaging*, pages 1–15. Springer, 2011. 3

[40] Mark A Robertson, Sean Borman, and Robert L Stevenson. Dynamic range improvement through multiple exposures. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 159–163, 1999. 7

[41] Daniel L Ruderman, Thomas W Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *JOSA A*, 15(8):2036–2045, 1998. 3

[42] Feng Shao, Gangyi Jiang, Mei Yu, and Ken Chen. A content-adaptive multi-view video color correction algorithm. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 1, pages I–969. IEEE, 2007. 1

[43] Zhuo Su, Kun Zeng, Li Liu, Bo Li, and Xiaonan Luo. Corruptive artifacts suppression for example-based color transfer. *IEEE Transactions on Multimedia*, 16(4):988–999, 2014. 3

[44] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1455–1461, 2016. 1

[45] Qi-Chong Tian and Laurent D Cohen. Histogram-based color transfer for image stitching. *Journal of Imaging*, 3(3):38, 2017. 1

[46] Qi Wang, Xi Sun, and Zengfu Wang. A robust algorithm for color correction between two stereo images. In *Asian Conference on Computer Vision*, pages 405–416. Springer, 2009. 1, 2

[47] Wei Xu and Jane Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 6

[48] Kenji Yamamoto, Tomohiro Yendo, Toshiaki Fujii, Masayuki Tanimoto, and David Suter. Colour correction for multiple-camera system by using correspondences. *The journal of the institute of Image Information and Television Engineers*, 61(2):213–222, 2007. 2

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6, 7, 8

[50] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2