

Color Restoration for Objects of Interest using Robust Image Features

David Chen ^{#1}, Vijay Chandrasekhar ^{#2}, Gabriel Takacs ^{#3}, Jatinder Singh ^{*4}, Bernd Girod ^{#5}

[#] *Information Systems Laboratory, Stanford University, Stanford, CA 94305, USA*

¹dmchen@stanford.edu, ²vijayc@stanford.edu, ³gtakacs@stanford.edu, ⁵bgirod@stanford.edu

^{*} *Deutsche Telekom Laboratories, San Francisco, CA 94105, USA*

⁴jatinder.singh@telekom.de

Abstract—Illumination distortion due to uncontrolled lighting can severely degrade the color appearance of a photo. Frequently, the desired colors for objects in a newly taken *query* image are found in a previously stored *database* image. Then, the goal is to change the colors in the query image to match the colors in the database image. This paper presents a color restoration system that automatically retrieves a database image which matches the query image, even if the two images are taken from different viewpoints and under different illuminations. Robust features enable both accurate retrieval from the database and efficient sampling of the color differences between the query and database images. A spatially varying color mismatch model is generated, and the colors of the query image are effectively restored.

I. INTRODUCTION

Photos are often taken in non-ideal lighting environments. When the illumination is not carefully controlled or there is camera glare, the resulting colors for objects of interest (OOIs) in the photos can differ significantly from their intended colors. The illuminant's spectral distribution has as much influence on the resulting colors as an object's own spectral distribution [1].

Post-photography color restoration is an effective remedy for photometric distortion. A previously stored image, called a *database* image, that shows the ideal colors for OOIs can provide guidance for color correction in the newly captured image, called a *query* image. For example, Fig. 1 shows a query-database pair for a CD cover. In this case, the OOIs are the different objects on the CD cover, and the color mismatches are very noticeable. The database and query images will generally be taken from different viewpoints, as in Fig. 1, which introduces geometric distortions between their corresponding OOIs. Given a large database, an automated restoration system must reliably select a database image containing matching OOIs, despite the geometric differences. The system must also be able to restore the colors of each query OOI in its given pose.

Many techniques have been previously invented to correct color distortion. Color restoration of faded movies in [2] uses global contrast enhancement. The technique cannot correct for spatially varying distortions like linear gradients. Similarly, adaptive color equalization is used in [3] to improve color appearance of old movies. Both [2] and [3] measure restoration subjectively and do not reference clean database versions.

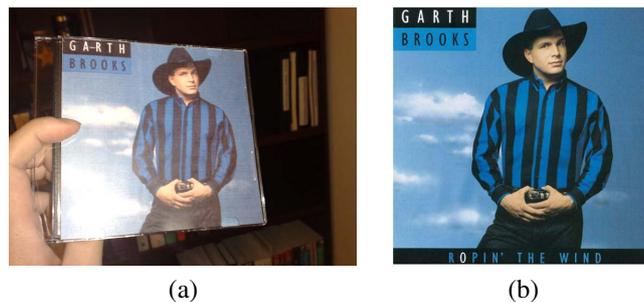


Fig. 1. (a) Color-distorted query CD cover image and (b) corresponding clean database CD cover image.

Restoration of paintings with reference to database images is presented in [4], in which color histograms are used to retrieve database matches. This retrieval method, however, assumes the color differences between the query image and matching database image are small. Color histograms are not robust against large photometric distortions which commonly occur in practice. In [5], the sample means of patches in a query painting are adjusted to match the sample means of corresponding patches in a database painting. The authors of [4] and [5] assume there are no geometric distortions between OOIs of query and database, i.e. color differences can be sampled from collocated pixels.

In this paper, we propose a color restoration system that can handle both photometric and geometric distortions between query and database OOIs. The key strategy is to use localized image features which are strongly robust against photometric and geometric distortions [6] [7]. First, the robust features enable accurate retrieval of a database image containing the same OOIs as the query image, even if the two images show different viewing angles and lighting conditions. Second, the geometric distortions between the matching OOIs in the two images can be calculated from the feature correspondences. Thus, the features allow sampling of the color differences in a much more general setting than [4] or [5]. If the database exists on a server, features can be sent using far fewer bits than the compressed query image itself. As special cases, our system can also correct distorted intensities in a grayscale query image or add color to grayscale query images.

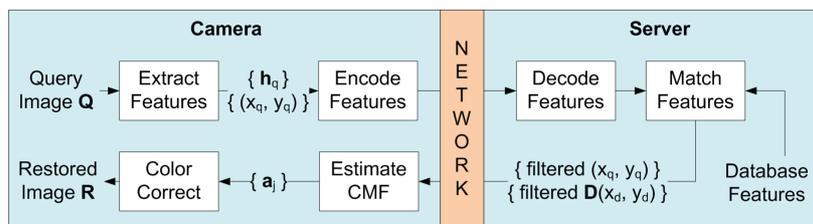


Fig. 2. Design of the proposed featured-based color restoration system.

Our proposed method of feature-based color sampling has a broader application than color restoration. The method can be used to efficiently compare two images in the presence of geometric distortions. Typically, two images are compared by measuring the differences between their collocated pixels. This approach would characterize the images in Fig. 1 as being very different, even though they share many visual similarities. In our framework, the geometric distortions separating two images can be found from robust feature correspondences, so differences between non-collocated pixels can be easily measured. This geometry-aware approach captures the important color differences after geometric compensation.

We will present feature-based color restoration as follows. Sec. II discusses feature-based image matching and presents a new technique on feature-based color difference sampling. A spatially varying color mismatch function (CMF) can be estimated from the samples, which describes how the query image should be color-corrected relative to the database image. Sec. III describes accurate but simple approximations for the CMF, leading to efficient modeling by linear regression. Experimental results for images of CD covers are presented in Sec. IV. The proposed color restoration system gives significant improvement, both subjectively and in terms of objective mean-squared-error (MSE) measurements.

II. FEATURE-BASED COLOR RESTORATION

An overview of the proposed system is given in Fig. 2. The top half of the system represents feature-based image matching. Features are first extracted from the query image on the camera, as explained in Sec. II-A. These features are encoded and decoded for transmission over the network. Then, on the server side, the query features are matched to database features, as explained in II-B. The bottom half of the system represents feature-based color restoration, which will be discussed in Sec. II-C. Color restoration uses the features and retrieved database image found in the top half.

A. Extraction of Robust Image Features

The first step in the image matching half of Fig. 2 is feature extraction from the query image. We utilize the speeded-up robust features (SURF) algorithm [7], which is similar to an earlier proposed scale-invariant feature transform (SIFT) algorithm [6]. SURF provides more compact feature descriptors and faster matching than SIFT.

SURF finds keypoints in an image corresponding to extrema in scale space. These keypoints can be reliably identified



Fig. 3. Query image with SURF points of interest overlaid.



Fig. 4. Query and database images with a perspective model between feature correspondences.

despite photometric and geometric distortions. The algorithm searches for keypoints at different scales, or different levels of a multiresolution pyramid, to be robust against scaling changes. Fig. 3 shows the keypoints displayed on top of the query image from Fig. 1(a). For the i^{th} keypoint, a 64-dimensional descriptor $\mathbf{h}_{q,i}$ is generated from histograms of gradients in a local neighborhood around the keypoint. Each histogram is calculated at the optimally selected scale found earlier and also along an optimally selected orientation. The complete query SURF feature consists of the descriptor $\mathbf{h}_{q,i}$ and the keypoint location $(x_{q,i}, y_{q,i})$.

B. Matching with Image Features

The M extracted query features $\{\mathbf{h}_{q,i}, (x_{q,i}, y_{q,i})\}_{i=1}^M$ are encoded for efficient transmission over the network, using the Karhunen-Loeve transform (KLT), scalar quantization, and Huffman coding. Sec. IV will show that the size of the compressed feature set is a small fraction of the size of a compressed query image. If the database of reference images resides on a remote server, the network represents a communication link such as a Bluetooth or WiFi channel.

At the server, the query descriptor set $\mathbf{H}_q \equiv \{\mathbf{h}_{q,i}\}_{i=1}^M$ is decoded. First, the query descriptors are classified through

a scalable vocabulary tree (SVT), constructed during a one-time training phase by hierarchical k-means clustering of all database descriptors. The SVT is used to narrow down the database search space to a small set of most likely database images [8].

Then, for each likely database image found by the SVT, its descriptor set can be effectively compared against the query descriptor set using the ratio test [6]. The ratio test can reject non-distinct descriptors which typically hinder, rather than aid, the matching process. Let the k^{th} database image have descriptor set $\mathbf{H}_{d,k} \equiv \{\mathbf{h}_{d,k,i}\}_{i=1}^{M_k}$, where M_k is the size of the set. Comparing \mathbf{H}_q to $\mathbf{H}_{d,k}$ using the ratio test results in number N_k^{RT} of common features which pass the ratio test.

The final step in feature-based image matching is geometric consistency checking (GCC) [6], which calculates a physically sensible affine or perspective transformation between the keypoints in the query and database images. An example of a valid perspective transformation between the keypoints for the query and database images of Fig. 1 is shown in Fig. 4. Only the N_k^{RT} features for the k^{th} database image which pass the ratio test are further tested by GCC. Then, a very reliable measure of matching accuracy is N_k^{GCC} , the number of features which pass both GCC and the ratio test ($N_k^{\text{GCC}} \leq N_k^{\text{RT}}$). Thus, our color restoration system selects the $\max_k N_k^{\text{GCC}}$ database image as the closest match. Sec. IV will demonstrate that feature-based retrieval provides very high matching accuracy.

C. Sampling at Feature Locations and Restoration

Robust images features enable accurate retrieval of a database image closely matched to the current query image. The features which pass both the ratio test and GCC also provide valuable information about the geometric distortions between the corresponding keypoints in the query and database images, as evident in Fig. 4. Larger OOIs can be defined as clusters of keypoints, so the geometric distortions between OOIs are also known. Color differences between query and database OOIs can now be sampled.

Suppose the set of N filtered query keypoints, which have passed the ratio test and GCC, is $\mathbf{P}_q \equiv \{(x_{q,i}, y_{q,i})\}_{i=1}^N$. Likewise, suppose the corresponding set of database keypoints is $\mathbf{P}_d \equiv \{(x_{d,i}, y_{d,i})\}_{i=1}^N$, where the i^{th} elements of \mathbf{P}_q and \mathbf{P}_d are geometrically matched. Then, samples of the color difference between query image \mathbf{Q} and database images \mathbf{D} are given by

$$\Delta_j(x_{q,i}, y_{q,i}) \equiv \mathbf{Q}_j(x_{q,i}, y_{q,i}) - \mathbf{D}_j(x_{d,i}, y_{d,i}) \quad j = 1, 2, 3 \quad i = 1, \dots, N, \quad (1)$$

where j indexes color channels, while q and d differentiate between query and database image spaces. Eq. (1) can be interpreted as samples of the color mismatch function (CMF). The CMF is a spatially varying description of how the query colors differ from the database colors. If the entire CMF can be accurately estimated, not only at the sample keypoints but across all of image space, then the query image can be color-corrected.

The filtered query keypoints and the pixel values at the filtered database keypoints need to be communicated from the database to the camera. Because the filtered keypoints represent a sparse sampling of image space, this information can be sent in very few bits, as shown in Sec. IV.

The remaining task is to reconstruct the CMF from the sparse set of samples in (1). In Sec. III, we will present practical techniques for generating an accurate estimate $\hat{\Delta}$. Subtracting $\hat{\Delta}$ from the query image \mathbf{Q} gives us a restored image \mathbf{R} :

$$\mathbf{R}_j(x_q, y_q) \equiv \mathbf{Q}_j(x_q, y_q) - \hat{\Delta}_j(x_q, y_q) \quad j = 1, 2, 3 \quad (x_q, y_q) \in \{\text{query image space}\}. \quad (2)$$

The restored image will show each OOI in the same pose as in the query image but now with colors correctly matched to the database image. Noticeable improvements from this restoration process will be shown in Sec. IV.

III. COLOR MISMATCH MODELS

In this section, we present two computationally efficient models for the CMF described in Sec. II. Both models perform well at approximating typical photometric distortions.

A. Linear Model

Oftentimes, a linear model is sufficiently accurate for describing the photometric distortion observed in a query image. It can correct two of the most common photometric distortions: constant shifts and linear gradients. We can calculate a CMF estimate of the form

$$\Delta_{\text{lin},j}(x_q, y_q) = a_j \cdot x_q + b_j \cdot y_q + c_j \quad j = 1, 2, 3, \quad (3)$$

where a separate set of parameters $\{a_j, b_j, c_j\}$ is used for each color channel. Given the samples in (1), the model parameters can be estimated by linear regression.

Assuming Δ_{lin} models Δ well for the N samples from (1), we can form the approximate matrix relationship

$$\underbrace{\begin{bmatrix} \Delta_j(x_{q,1}, y_{q,1}) \\ \vdots \\ \Delta_j(x_{q,N}, y_{q,N}) \end{bmatrix}}_{\Delta_j} \approx \underbrace{\begin{bmatrix} x_{q,1} & y_{q,1} & 1 \\ \vdots & \vdots & \vdots \\ x_{q,N} & y_{q,N} & 1 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} a_j \\ b_j \\ c_j \end{bmatrix}}_{\mathbf{a}_j} \quad j = 1, 2, 3. \quad (4)$$

If $N > 3$, which is always the case for practical feature-based image retrieval, the system of equations is overdetermined, and a least squares solution for the model parameters is appropriate for minimizing the modeling error at the sample keypoints. This solution is given by

$$\hat{\mathbf{a}}_{\text{glob},j} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \Delta_j \quad j = 1, 2, 3. \quad (5)$$

After calculating (5), we can estimate $\hat{\Delta}$ across all of query image space using (3) and subsequently perform color restoration by (2).

The proposed technique can be applied in any three-channel color space, such as RGB or CIELAB spaces. Additionally,

as special cases, if the query image is grayscale while the database image is color, we can (i) apply the restoration in a single channel to correct intensity distortions or (ii) convert the query image into a color version.

B. Piecewise Linear Model

Higher order CMF models, like quadratic and cubic functions, tend to overcompensate in regions of the query image away from dense clusters of feature keypoints. Overcompensation occurs even with moderate regularization. When extreme regularization is applied, the higher order model essentially becomes the linear model. An intermediate model is a piecewise linear model, which can approximate nonlinear distortions better than the linear model but at the same time avoid the overcompensating behavior of higher order models.

To generate a piecewise linear model, the feature keypoints must first be separated into several clusters. A simple approach is to use the k-means clustering algorithm [9]. Alternatively, the query image can be divided into multiple OOIs by some suitable segmentation algorithm [10] [11], and the keypoints within each segmented OOI would form their own cluster.

Suppose there are K clusters of query keypoints and K corresponding clusters of database keypoints. For each of the clusters, we fit a different linear model. Instead of directly applying the techniques discussed in Sec. III-A, however, proper regularization must be added to ensure a stable solution for each cluster. Regularization is particularly important for any cluster with a small number of elements, because the least squares solution is very sensitive to noise when only a few samples are available. Our piecewise linear model utilizes Tikhonov regularization, which biases each local solution towards a stable global solution [12].

The stable global solution is the least squares solution $\hat{\mathbf{a}}_{\text{glob},j}$ ($j = 1, 2, 3$) given by (5). Then, the local Tikhonov-regularized solution is

$$\hat{\mathbf{a}}_{\text{loc } k,j} = \hat{\mathbf{a}}_{\text{glob},j} + (\mathbf{X}_k^T \mathbf{X}_k + \lambda_k \mathbf{I})^{-1} \mathbf{X}_k^T (\Delta_{k,j} - \mathbf{X}_k \hat{\mathbf{a}}_{\text{glob},j}) \quad j = 1, 2, 3 \quad k = 1, \dots, K, \quad (6)$$

where \mathbf{I} is an appropriately sized identity matrix, and \mathbf{X}_k and $\Delta_{k,j}$ are defined analogous as \mathbf{X} and Δ_j in (4), except using only the elements of the k^{th} cluster. Additionally, λ_k is the regularization parameter for the k^{th} cluster, where larger values of λ_k bias the local solution more towards the global solution. Intuitively, clusters with a lower number of elements need more guidance from the global solution. Thus, our selection is λ_k is

$$\lambda_k \equiv \frac{\text{num. elements overall}}{\text{num. elements in } k^{\text{th}} \text{ cluster}}. \quad (7)$$

As the number of elements in the k^{th} cluster decreases, $\hat{\mathbf{a}}_{\text{loc } k,j}$ moves closer to $\hat{\mathbf{a}}_{\text{glob},j}$.

Near region boundaries, the different piecewise models need to be smoothly merged together to avoid visually unpleasant boundary artifacts. We propose a merging technique in which all K local models exert some influence at each point of image

space, but a local model's influence diminishes rapidly as the distance increases from the cluster centroid. Specifically, the merged model at point (x_q, y_q) is

$$\hat{\mathbf{a}}_{\text{mer},j}(x_q, y_q) = \sum_{k=1}^K \alpha_k(x_q, y_q) \hat{\mathbf{a}}_{\text{loc } k,j} \\ \alpha_k(x_q, y_q) = \frac{\| (x_q, y_q) - (x_{\text{cen } k}, y_{\text{cen } k}) \|^2}{\sum_{k'=1}^K \| (x_q, y_q) - (x_{\text{cen } k'}, y_{\text{cen } k'}) \|^2} \\ j = 1, 2, 3, \quad (8)$$

where $(x_{\text{cen } k}, y_{\text{cen } k})$ is the centroid of the k^{th} cluster. Near a cluster centroid, the merged model becomes effectively that cluster's local model. Between neighboring clusters, the merged model is an average of those clusters' local models.

IV. EXPERIMENTAL RESULTS

Applying the color restoration system described in Sec. II along with the models introduced in Sec. III, the color appearance of query images can be significantly improved. On the camera side, photos of 50 different CD covers were taken, and each query image has resolution of 2592×1944 pixels. Uneven illumination and camera glare can be observed in most of the query photos. Most query CD covers differ noticeably in color appearance from their clean database versions, as seen in the first two columns of Fig. 5. On the database side, 10,000 clean CD covers are stored, and each database image has resolution of 500×500 pixels. Results reported here are for color restoration in RGB space, but similar results are obtained for color restoration in CIELAB space. For the piecewise linear model, the centroids of the four quadrants of the CD cover are used for the initialization of the k-means algorithm.

A. Image Matching Accuracy

Query and database SURF features are matched using the criteria discussed in Sec. II. The result is that 48 of the 50 query images are correctly matched to their database versions. SURF features are robust against the challenging photometric and geometric distortions encountered in the query images.

B. Visual Color Improvements

The restored images are much closer in color appearance to the database images than the unprocessed query images. Fig. 5 shows four sets of images, each set consisting of the unprocessed query image, the clean database image, the linearly restored image, and the piecewise linearly restored image. Both models perform well at shifting the query colors towards the reference database colors. The piecewise linear model is able to make more precise local adjustments.

C. MSE Reductions

Improvements from color restoration can also be quantified using an objective mean squared error (MSE) metric. To measure MSE between query and database CD covers, the geometric distortion between them must first be removed. We project each query cover into a rectified 500×500 square so that it can be directly compared against the database cover. The



Fig. 5. (a-d) Unprocessed query images. (e-h) Database images. (i-l) Restoration with linear model. (m-p) Restoration with piecewise linear model.

projection uses a perspective transformation calculated from the filtered feature keypoints from Sec. II-B. Image registration by integer-pel shifting over a small search range of $[-10,10]$ is also applied to reduce misalignment after projection.

Because the MSE fluctuates from one image to the next due to changing lighting conditions, each MSE reported here is normalized by the MSE between the (projected) unprocessed query cover and the database cover. The normalized MSE of the unprocessed query cover is always unity. If the normalized MSE of the color-corrected query cover falls below unity, color restoration has successfully reduced the MSE.

Labels under the images in Fig. 5 show the normalized MSE values. All four sets show significant reductions in MSE after color restoration, as much as 87 percent. Table I lists the mean normalized MSE values, averaged over all 48 query covers

TABLE I
MEAN NORMALIZED MSE VALUES FOR QUERY CD COVERS.

Method	Mean Normalized MSE
Unprocessed	1.000
Linear	0.330
Piecewise Linear	0.327

that are correctly matched. Piecewise linear restoration slightly outperforms linear restoration.

D. Model Comparison

As shown in Sec. IV-B and IV-C, the linear and piecewise linear models are both effective at reducing color distortion in the query images. Both models are accurate approximations

TABLE II
AVERAGE TRANSMISSION COSTS FOR COMPRESSED DATA.

Quantity	Average Size (Kilobytes)
Query Image (2592 x 1944)	660
Database Image (500 x 500)	59
Feature Set	11
CMF Sample Set	1

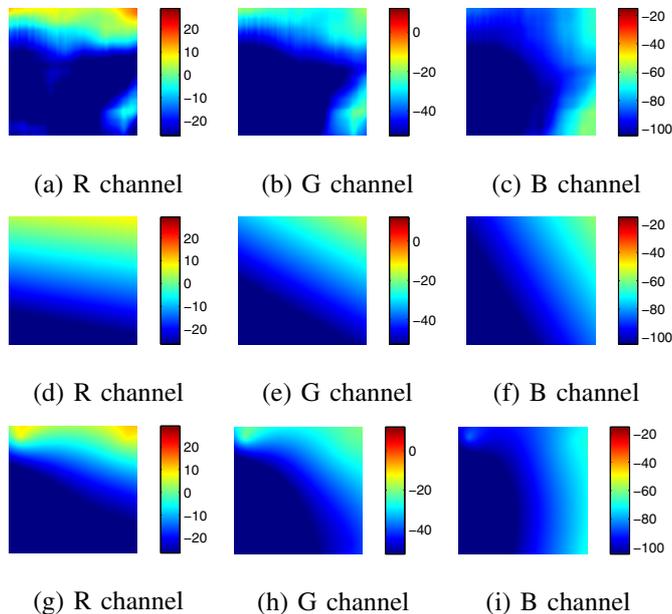


Fig. 6. (a-c) True CMF. (d-f) Linear model CMF. (g-i) Piecewise linear model CMF. Test images come from third row of Fig. 5.

of the true CMF. Fig. 6 plots the true CMF and model CMFs for the test images in the third row of Fig. 5. The large scale variations of the true CMF across image space are accurately captured by both models. Also, as designed, the piecewise linear model can better adapt to these local variations.

E. Transmission Costs

Transmission costs over the network for image retrieval and color restoration are low compared to the transmission costs for compressed images, as shown in Table II. Images are compressed at medium quality using JPEG. Feature sets are transformed by KLT, scalar quantized, and entropy coded, and a compressed bitstream is sent from the camera to the server. In response, the server sends back the keypoint locations for features which pass the ratio test and GCC and the associated pixel values.

F. Complexity Analysis

The complexity of the color restoration algorithm is dependent on the number of features N_f that pass the ratio

test and GCC. Sampling the color differences in (1) is an $O(N_f)$ operation. Calculating a least-squares model from the N_f color difference samples has $O(N_f^3)$ complexity because matrix inversion and multiplication are required in (5) and (6). Finally, color-correcting the query image in (2) requires $O(N_p)$ complexity, where N_p is the number of pixels.

V. CONCLUSION

This paper has presented an effective color restoration system for correcting photometric distortions in any query image. The system accurately uses robust features to retrieve a matching database image that serves as color reference, even when geometric and photometric distortions exist between common objects in the query and database images. By efficiently reusing the feature keypoints to sample the color differences between query and database images, the photometric distortion can be reliably estimated and diminished. Restored query images show significant color improvements, both subjectively and objectively. Transmission costs for both retrieval and color restoration are fairly low, and the proposed method can be easily integrated into any existing feature-based technologies. Feature-based color sampling also has a broader application for comparing two images whose corresponding pixels are not collocated.

REFERENCES

- [1] B. A. Wandell, *Foundations of Vision*. Sinauer Associates, Inc., 1995.
- [2] M. Chambah and B. Bessner, "Digital color restoration of faded motion pictures," in *Proceedings of International Conference on Color in Graphics and Image Processing*, Saint-Etienne, France, October 2000, pp. 381-395.
- [3] A. Rizzi, C. Gatta, C. Slanzi, and G. Ciocca, "Unsupervised color film restoration using adaptive color equalization," in *Proceedings of Visual Information and Information Systems*, Amsterdam, The Netherlands, July 2005, pp. 1-12.
- [4] X. Li, D. Lu, and Y. Pan, "Color restoration and image retrieval for Dunhuang fresco preservation," *IEEE Multimedia Magazine*, vol. 7, no. 2, pp. 38-42, June 2000.
- [5] M. Pappas and I. Pitas, "Digital color restoration of old paintings," *IEEE Transactions Image Processing*, vol. 9, no. 2, pp. 291-294, February 2000.
- [6] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, November 2004.
- [7] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *Proceedings of European Conference on Computer Vision*, Graz, Austria, May 2006, pp. 404-417.
- [8] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 2006, pp. 2161-2168.
- [9] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100-108, 1979.
- [10] Y. L. Chang and X. Li, "Adaptive image region-growing," *IEEE Transactions Image Processing*, vol. 3, no. 6, pp. 868-872, November 1994.
- [11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, September 2004.
- [12] A. Tarantola, *Inverse Problem Theory*. Society for Industrial and Applied Mathematics, 2004.