Interactive Illumination Invariance

Han Gong, Graham Finlayson
University of East Anglia, UK

Abstract

Illumination effects cause problems for many computer vision algorithms. We present a user-friendly interactive system for robust illumination-invariant image generation. Compared with the previous automated illumination-invariant image derivation approaches, our system enables users to specify a particular kind of illumination variation for removal. The derivation of illumination-invariant image is guided by the user input. The input is a stroke that defines an area covering a set of pixels whose intensities are influenced predominately by the illumination variation. This additional flexibility enhances the robustness for processing non-linearly rendered images and the images of the scenes where their illumination variations are difficult to estimate automatically. Finally, we present some evaluation results of our method.

Introduction

Illumination conditions affect many computer vision algorithms from fundamental level to high level. For instance, unwanted shadows can cause artefacts in image segmentation, tracking, and object recognition. This is due to the difficulty in distinguishing the different image edges caused by illumination discontinuity, surface material [10], occlusion [16], and object boundary [4]. An illumination-invariant image is therefore in great demand in both computer vision and computer graphics communities.

A shadow is an area where light from a light source is blocked by an obstacle. This is perhaps the most important phenomenon caused by illumination. There has been much previous work [9, 7, 13, 15, 12, 11] focusing on removing shadows from single images. These approaches aim at restoring an image with predominant shadows removed. In this restored image, most surface texture information is intact. The minor shadows are usually kept and they are regarded as the surface textures. The diffuse (extremely soft) shadows are also not removed because these approaches rely on image segmentation or image edge detection but the soft shadow boundaries are not easily detectable. Even when the soft shadow edges are detected, it is still very difficult to restore the penumbra information as the surface textures are the strong noise that affect illumination estimation. In fact, the recovery of penumbra is the most difficult part of this problem. Only a few of them [15, 11] can preserve the penumbra texture for a wide range of different shadows, such as broken shadows, coloured shadows, and soft shadows. The recovery process is also computation-costly that even some fast methods [15, 11] can still take several seconds to solve, by using an average machine.

Intrinsic image decomposition divides a single image into two components: a shading component and a reflectance component which is independent of illumination. The original single image is the product of these two components. These approaches [2, 14, 3] commonly make two assumptions: (1) Neighbouring pixels have the same reflectance if their chromaticities are similar; (2) Intensity discontinuities in the luminance of an image are caused by sharp reflectance changes and the illumination change is only smooth. The decomposed components are useful for image editing, such as re-colouring and surface material replacement. Most of them require a sliding window to process a local image block and an optimisation energy function to enforce the smoothness of decomposition results for different local image blocks. The optimisation process is usually very computational-costly as it requires the adjustment for every single pixel in each iteration. The resulting reflectance component can often contain visible illumination traces.

Illumination-invariant image computation is related to the intrinsic image derivation. An image is derived that depends only on reflectance. An important illuminant invariant image is derived in work [8] where it was shown that there existed a projection direction in a log RGB image that depended only on reflectance. Initially, the calibration direction was found using a calibration procedure but [6] adopts an entropy minimisation procedure to find the best projection direction. The intuition here is that the same surface in a log RGB space (and assuming Planckian Illumination) will as the illumination changes produces values that fall on a line. Different surfaces induce different parallel line. The projection direction, orthogonal to the direction of variance, was shown to minimise entropy. Unfortunately, for the entropy minimisation strategy to work the input should be raw linear images. This is a serious limitation - one which we address here - because many of the applications of intrinsic images are for rendered, non-linear, images (e.g. you want to remove a shadow on a family photo). The entropy minimisation strategy also does not guarantee the best invariant direction when the predominant 2D projection direction is not caused by the prevailing illumination. The derivation process can also remove important texture information when the intensity variation caused by texture is similar to the estimated illumination variation. When the illumination invariant direction is known, these algorithms are very fast (usually only take some milliseconds). Although the invariant methods [6] and [9] at a first step produce a single linear combination of log chromaticity values that is invariant to illumination, this can be converted into a pseudo colour chromaticity image [5].

This paper is concerned with a novel interactive approach for illumination-invariant image derivation. Instead of finding the invariant direction by either camera calibration [9] or entropy minimisation [6], we require the user to supply some easy hints (usually only one single stroke) to guide the derivation of the illumination variation direction. The required user input is a simple stroke that indicates an area covering a set of pixels whose intensity variations are predominately caused by a variation in illumination. This additional user task is easy and quick to perform. It also makes the illumination-invariant derivation steerable for the best result. Our method works in a very simple way: Given a segment of an image defined by a single input stroke, the pixels of that segment are divided into two groups according to the features of their intensities and coordinates. To get this illumination invariant direction, all the RGB intensity values of the image are first converted to 2D log geometric mean chromaticites by projecting their 3D log geometric mean chromaticities onto a plane [6]. It is assumed that the intensity variation of the pixels in this segment is mainly caused by the illumination. On this 2D chromaticty plane, the intensity difference between these two groups thus represents the illumination direction. As the given rough segment can contain noise and outliers (usually caused by surface texture or surface material change), some outlier amendment steps are also applied to resolve this issue. To get the 1D illumination-invariant image, our method projects the 2D chromaticities onto a line orthogonal to the illumination invariant direction vector. The 2D illumination-invariant image can be obtained by applying a 1D-to-3D projection to the projected 1D illumination-invariant values.

This paper is organised as follows. Firstly, we describe our user interaction design and the pre-processing step for user input. Secondly, we describe our approach for illumination-invariant image generation guided by user input. Finally, some evaluation results of our interactive illumination-invariant image are shown.

User Interaction And Pre-Processing

In this section, we describe our user interaction design and its algorithm.

User Interaction

Fig. 1 shows the prototype of our user interface and the description of its controls. The required mark is a segment in a shape that is similar to an ellipse, e.g., the divided mark in blue and red in Fig. 1. The direction of its semi-major axis indicates the potentially strongest intensity gradient change direction. This requirement is easy to fulfil and it gives an additional cue to make our user input pre-processing more resistant to strong surface textures. Our system also allows users to supply multiple marks to improve the result. The proposed drawing actions have two types:
Mark Addition Mark addition is equivalent to adding or sometimes merging a new part to the previous marks. This is done by applying a logical *or* image operation as follows:

$$N_n = N \vee N_a \tag{1}$$

where N refers to the current mask of mark, N_a is the additional mask of mark drawn by user, N_n is the updated mask of mark.
Mark Subtraction Mark subtraction is similar to a eraser which is used to remove the existing marks. Similar to Eq. 1, mark subtraction is done by applying two logical image operations as follows:

$$N_n = N \wedge \neg N_a \tag{2}$$

Pre-Processing

Given the marked pixels of an image, our pre-processing step divides them into two groups. The marked pixels' RGB intensities and pixel coordinates are used as the features for division. For each marked illuminant change, we find the RGB that is in the shadow and the one outside the shadow by a K-Means++ clustering [1]. The marked pixels' 2D pixel coordinates are not directly used as the features. Instead, they are processed using PCA [17] and only their scores of the first principle component are used as the feature of location. We further normalise the scores of the first principle such that their values are in the range $[0,1]$. The intuition is that the vector of the first principle dimension is close to the major semi-axis of an ellipse. The direction of this semi-axis indicates the direction of the strongest illumination change. The RGB intensities are also normalised to the range $[0,1]$. Therefore, our final feature space has 4 dimensions: three for RGB intensities and one for pixel location. Our main configuration for this K-Means++ clustering are: (1) Distance Measure: Squared Euclidean distance; (2) Maximum number of iteration: 100. The cluster with the lowest mean for its RGB intensity is considered as a shadow cluster and vice versa. For each cluster, we find the median RGB intensities.

Illumination-Invariant Image

Using the algorithm described in [6], the RGB intensities of the original image and the two single median RGB intensities (analysed from the user input) are first projected onto a 2D log chromaticity plane such that each pixel has two log chromaticity values. The user input provides an important cue for finding the accurate projection direction under such conditions. Given the converted 2D log chromaticity vectors C^L (lit) and C^S (shadow) for the two single median RGB intensities, the normalised illumination direction vector $P^{\perp}$ is

$$P^{\perp} = \frac{C^L - C^S}{|C^L - C^S|}. \tag{3}$$

The projection (i.e. illumination-invariant) vector P is orthogonal to $P^{\perp}$ and can be computed by rotating $P^{\perp}$ by 90 degrees clockwise as the follows

$$P = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} P^{\perp}. \tag{4}$$

An example of this projection is shown in Fig. 2. Given a pixel's 2D log chromaticity vector and the illumination-invariant vector P, its 1D illumination-invariant value and 3D L1 log chromaticity value can be computed using the projection algorithm described in [6]. An example of this illumination-invariant image is shown in Fig 1.

Evaluation

In this section, we first show some typical examples with a result comparison highlighting the advantage of using user interaction. Finally, some failure cases are shown.

Results

In Fig. 3, some typical examples are shown. This set of examples covers a wide range of different illumination conditions, such as broken shadows (row 3, 4, 9, and 10), soft shadows (row 2 and 7), coloured shadows (row 1 and 11) as well as the other trivial shadows. Compared with the automated entropy minimisation approach [6], our method produces more clean (shadow-free) illumination-invariant images for row 1, 2, 6, 8, 9, and 11. Our results in row 3, 4, 7, and 10 also retain more surface texture and

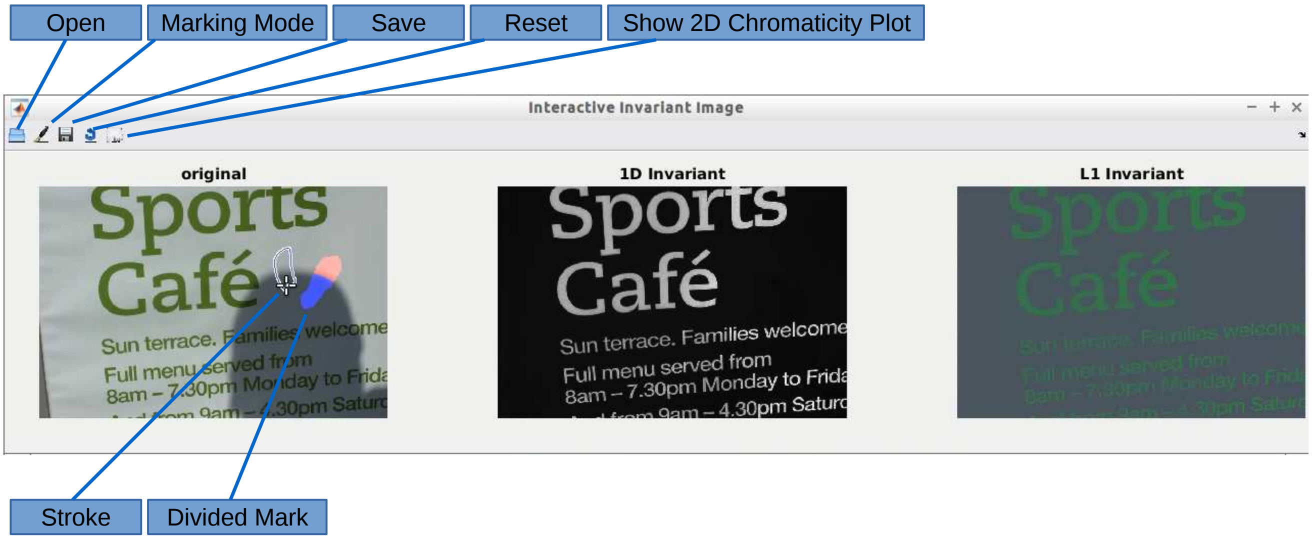

Figure 1: Graphical user interface: (1) open – open an image file; (2) marking mode – choose either to draw or to erase marks; (3) save – save the illumination-invariant images; (4) reset – clear all marks; (5) stroke – a stroke enclosing an area of mark; (6) divided mark – the mark being divided into a lit part (in red) and a shadow part (in blue). The three images shown in the interface are: left – original RGB image; middle – 1D illumination-invariant image; right – L1 illumination-invariant log chromaticity image. The input image is taken from a shadow removal dataset [11].

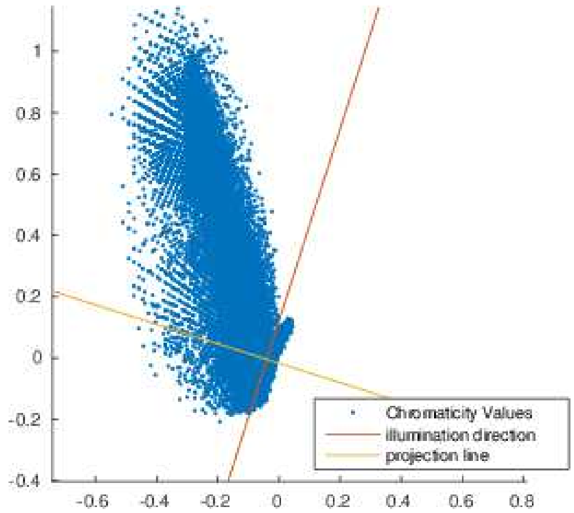

Figure 2: Example of illumination-invariant projection in a 2D log chromaticity space defined in [6]. The log chromaticity points are projected on to the projection line (yellow). The data displayed in this figure corresponds to the image data in Fig. 1.

chromaticity information. For the images in row 5, the energy minimisation approach produces slightly better results than ours. For a 0.3 mega-pixel image, our single thread MATLAB implementation usually takes 0.3s to process on a 2.4GHz machine. In summary, with the assistance of user input, our approach produces better results for most difficult images at a very fast processing speed.

Failure Cases

Since we still assume the near linearity of pixel intensities, our approach fails when the image has undergone a heavy non-linear rendering process, e.g., low quality JPEG compression. A failure cases is shown in Fig. 4. In Fig. 4, the shadow and the JPEG blocks are clearly visible in the 1D illumination-invariant image and its L1 log chromaticity image almost contains no information. Fig. 5 shows the 2D log chromaticity plot of Fig. 4.

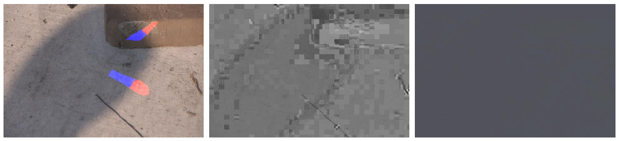

Figure 4: Failure cases (heavily JPEG compressed): left – original image with user input; middle – 1D illumination-invariant image; right – L1 log chromaticity illumination-invariant image.

Conclusion

We have presented a novel approach for interactive illumination-invariant image derivation. Using our system, users are only required to supply a simple stroke defining an area in an image where the illumination change is significant. Compared with other automated methods, this additional hint significantly enhances the robustness of finding the illumination-invariant direction and makes the illumination-invariant image derivation steerable. Future work includes: (1) A more noise-resistant user input analysis; (2) A more adaptive algorithm to deal with the non-linearity of rendered images.

References

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the annual symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] Jonathan T Barron and Jitendra Malik. Color constancy, intrinsic

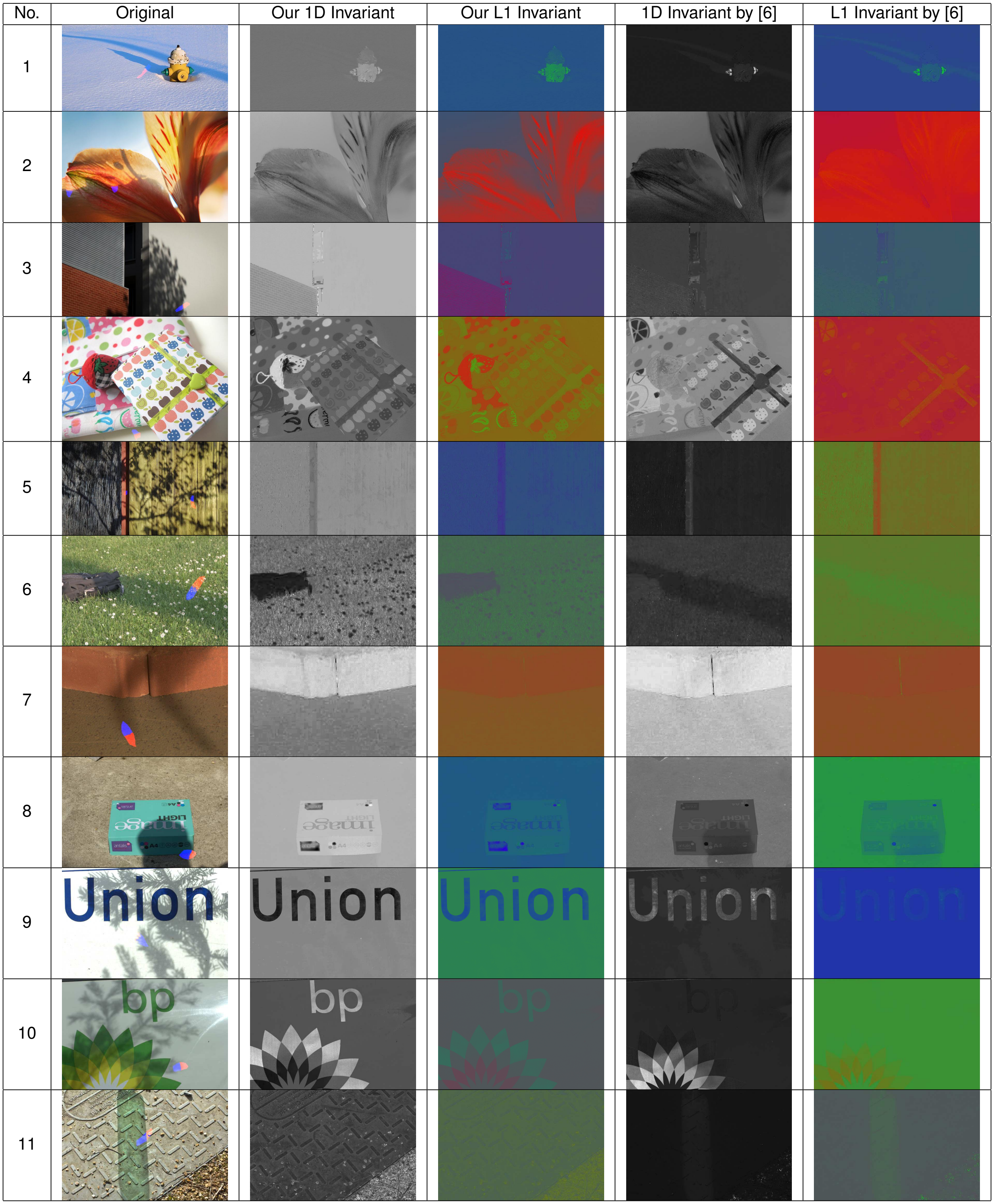

Figure 3: Results: All the images in this figure are gamma corrected. All the images, except for the bottom four images, are JPEG compressed. The guidance marks are displayed on each original image (please zoom in the electronic version to examine some minor marks). These non-linear operations have severally confound the entropy minimisation approach [6]. Images 1-5, 6-7, and 8-11 are sourced from **Flickr**, [13], and [11] respectively.

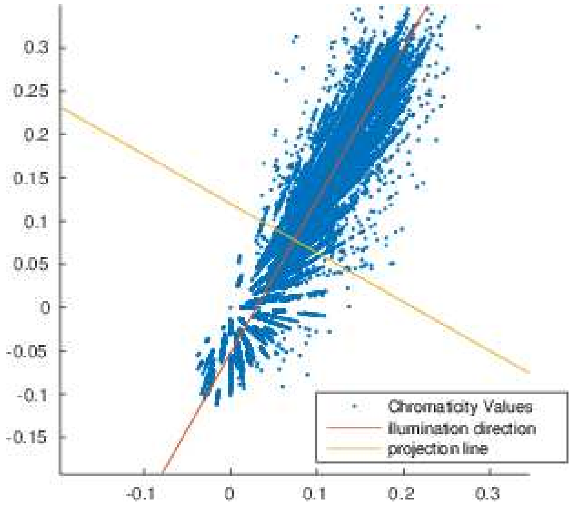

Figure 5: 2D log chromaticity of a JPEG compressed image: The lines of chromaticity points are not parallel but spread in different directions. This is due to the non-linear post-processing operations.

images, and shape estimation. In *European Conference on Computer Vision*, pages 57–70. Springer, 2012.

[3] Adrien Bousseau, Sylvain Paris, and Frédo Durand. User-assisted intrinsic images. volume 28, page 130. ACM, 2009.

[4] Piotr Dollar, Zhuowen Tu, and Serge Belongie. Supervised learning of edges and object boundaries. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1964–1971. IEEE, 2006.

[5] Mark S Drew, Graham D Finlayson, and Steven D Hordley. Recovery of chromaticity image free from shadows via illumination invariance. In *Workshop on Color and Photometric Methods in Computer Vision*, pages 32–39. IEEE, 2003.

[6] Graham D Finlayson, Mark S Drew, and Cheng Lu. Intrinsic images by entropy minimization. In *European Conference on Computer Vision*, pages 582–595. Springer, 2004.

[7] Graham D Finlayson, Mark S Drew, and Cheng Lu. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 85(1):35–57, 2009.

[8] Graham D Finlayson and Steven D Hordley. Color constancy at a pixel. *Journal of the Optical Society of America A*, 18(2):253–264, 2001.

[9] Graham D Finlayson, Steven D Hordley, Cheng Lu, and Mark S Drew. On the removal of shadows from images. *Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006.

[10] Ron Gershon, Allan D Jepson, and John K Tsotsos. Ambient illumination and the determination of material changes. *Journal of the Optical Society of America A*, 3(10):1700–1707, 1986.

[11] Han Gong and Darren Cosker. Interactive shadow removal and ground truth for variable scene categories. In *British Machine Vision Conference*. BMVA, 2014.

[12] Han Gong, Darren Cosker, Chuan Li, and Matthew Brown. User-aided single image shadow removal. In *International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2013.

[13] Ruiqi Guo, Qieyun Dai, and Derek Hoiem. Paired regions for shadow detection and removal. *Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967, 2013.

[14] Carsten Rother, Martin Kiefel, Lumin Zhang, Bernhard Schölkopf, and Peter V Gehler. Recovering intrinsic images with a global sparsity prior on reflectance. In *Advances in neural information processing systems*, pages 765–773, 2011.

[15] Ya-Fan Su and Homer H Chen. A three-stage approach to shadow field estimation from partial boundary information. *Transactions on Image Processing*, 19(10):2749–2760, 2010.

[16] Patrik Sundberg, Thomas Brox, Michael Maire, Pablo Arbeláez, and Jitendra Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *Conference on Computer Vision and Pattern Recognition*, pages 2233–2240. IEEE, 2011.

[17] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52, 1987.