



(51) International Patent Classification:  
H04N 5/21 (2006.01)

(21) International Application Number:  
PCT/US2011/062787

(22) International Filing Date:  
1 December 2011 (01.12.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
61/418,853 1 December 2010 (01.12.2010) US  
13/308,411 30 November 2011 (30.11.2011) US

(71) Applicant (for all designated States except US):  
**GOOGLE INC.** [US/US]; 1600 Amphitheatre Parkway,  
Mountain View, CA 94043 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **KWATRA, Vivek**  
[IN/US]; c/o Google Inc., 1600 Amphitheatre Parkway,  
Mountain View, CA 94043 (US). **HAN, Mei** [CN/US]; c/o  
Google Inc., 1600 Amphitheatre Parkway, Mountain View,  
CA 94043 (US).

(74) Agents: **SACHS, Robert R.** et al.; FENWICK & WEST  
LLP, Silicon Valley Center, 801 California Street, Moun-  
tain View, CA 94041 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

(54) Title: REMOVING ILLUMINATION VARIATION FROM IMAGES

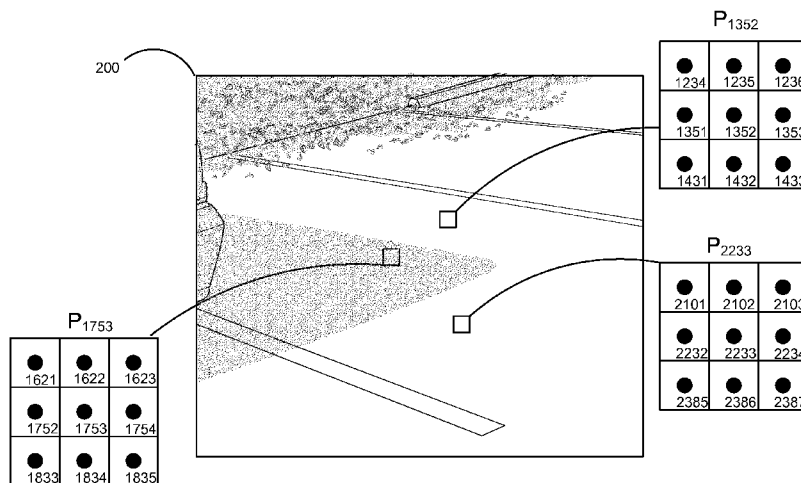


Fig. 2

(57) Abstract: An image comprising varying illumination is selected. Patches of pixels from among the plurality of pixels with the image are identified. Similarities between pairs of patches of pixels based on pixel intensities associated with the pairs of patches of pixels are calculated. Illumination values for the plurality of pixels within the image based on the calculated similarities between the pairs of patches of pixels is calculated. The illumination variation from the image is removed based on the calculated illumination values for the plurality of pixels within the image.

WO 2012/075227 A1

# REMOVING ILLUMINATION VARIATION FROM IMAGES

INVENTORS:

VIVEK KWATRA

MEI HAN

## 1. CROSS-REFERENCE TO RELATED APPLICATIONS

**[0001]** This application claims the benefit of U.S. Provisional Application No. 61/418,853, filed December 1, 2010, which is incorporated by reference in its entirety.

## 2. FIELD OF ART

**[0002]** The present disclosure generally relates to the field of digital image processing, and more specifically, to methods of estimating illuminations in images.

## 2. BACKGROUND

**[0003]** Many current information systems, such as map applications, store large quantities of digital images that provide views of the surrounding area for a given map location. Many of these images include undesirable characteristics such as uneven ambient illumination that degrade the content of the images. For example, some map systems contain aerial images which often contain undesirable shadows caused by clouds or other objects. Thus, such images may include an uneven illumination caused by objects which occlude the light source of the image. The presence of the uneven illumination in the images thereby results in undesirable characteristics due to the lack of uniformity of color intensity in the images. As a result, these systems provide users with images of less than desirable quality.

## SUMMARY

**[0004]** The problems described above are met by a computer-implemented method, a non-transitory computer-readable storage medium and a computer system for removing illumination variation from an image. One embodiment of the computer-implemented method comprises selecting an image containing varying illumination across a plurality of pixels within the image. Patches of pixels from among the plurality of pixels within the image are identified. Similarities between pairs of patches of pixels are calculated based on pixel intensities associated with the pairs of patches of pixels. Illumination values for the plurality of pixels within the image are calculated based on the calculated similarities

between the pairs of patches of pixels. The illumination variation from the image is removed based on the calculated illumination values for the plurality of pixels within the image.

**[0005]** Embodiments of the non-transitory computer-readable storage medium store computer-executable code for removing illumination variation from an image. The code are executable to perform steps comprising selecting an image containing varying illumination across a plurality of pixels within the image and identifying patches of pixels from among the plurality of pixels within the image. The steps further include calculating similarities between pairs of patches of pixels based on pixel intensities associated with the pairs of patches of pixels and calculating illumination values for the plurality of pixels within the image based on the calculated similarities between the pairs of patches of pixels. Furthermore, the steps comprise removing the illumination variation from the image based on the calculated illumination values for the plurality of pixels within the image.

**[0006]** Embodiments of the computer system for removing an illumination variation from an image comprises a computer processor and a non-transitory computer-readable storage medium storing executable instructions configured to execute on the computer processor. The instructions when executed by the computer processor are configured to perform steps comprising selecting an image containing varying illumination across a plurality of pixels within the image and identifying patches of pixels from among the plurality of pixels within the image. The steps further include calculating similarities between pairs of patches of pixels based on pixel intensities associated with the pairs of patches of pixels and calculating illumination values for the plurality of pixels within the image based on the calculated similarities between the pairs of patches of pixels. Furthermore, the steps comprise removing the illumination variation from the image based on the calculated illumination values for the plurality of pixels within the image.

**[0007]** The features and advantages described in the specification are not all inclusive and, in particular, many additional features and advantages will be apparent to one of ordinary skill in the art in view of the drawings, specification, and claims. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter.

## BRIEF DESCRIPTION OF DRAWINGS

- [0008] FIG. 1 is a high-level block diagram of an image processing server in accordance with one embodiment.
- [0009] FIG. 2 illustrates an image that includes an illumination in accordance with one embodiment.
- [0010] FIG. 3 illustrates the image of FIG. 2 with the illumination removed from the image in accordance with one embodiment.
- [0011] FIG. 4 illustrates a flowchart of the steps performed by the image processing server to remove illumination variation from an image in accordance with one embodiment.
- [0012] FIG. 5 is a high-level block diagram illustrating a typical computer for use as an image processing server according to one embodiment.
- [0013] The figures depict, and the detail description describes, various non-limiting embodiments for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles described herein.

## DETAILED DESCRIPTION

### SYSTEM ARCHITECTURE

- [0014] FIG. 1 is a high-level block diagram illustrating a detailed view of an image processing server 100 for removing illumination variation from images in accordance with one embodiment. In one embodiment, an illuminated image comprises an image showing a scene having ambient illumination such as shadows. An illuminated image may be represented as an underlying non-illuminated image and an illumination map. In one embodiment, the illumination map describes the illumination value at each location (x, y) in an image.
- [0015] Note that the illumination described by the illumination map may vary across the illuminated image. For example, part of the illuminated image can be in shadow and therefore have decreased illumination relative to the remainder of the image. Under this formulation, moreover, an “illumination” can describe both a shadow where light is occluded from an object, and a bright area. Shadows and bright areas are simply variations in the illumination map.

[0016] Thus, this description uses the term “illumination” to encompass variations in pixel intensity caused by occlusions. An occlusion-type illumination (i.e., a shadow) can be present in an image, for example, when sunlight incident upon an object (e.g., the ground) shown in the image is partially occluded by a cloud or other object. Thus, the occlusion results in a portion of the image being darker than other portions.

[0017] In one embodiment, the observed pixel intensity  $\tilde{I}$  of a given pixel in an illuminated image at location (x,y) is based on the following factors:

- The luminance  $\tilde{L}$  that describes a scalar illumination value of the illumination map for the pixel at location (x, y); and
- The reflectance  $\tilde{R}$  that describes the pixel intensity of the pixel’s underlying reflectance value (i.e., the non-illuminated value) for the pixel at location (x, y).

[0018] The relationship of these factors to each other is represented by the following equations:

$$\tilde{I}_p = \tilde{L}_p \cdot \tilde{R}_p \quad (1)$$

$$I_p = \log \tilde{I}_p = \log \tilde{L}_p + \log \tilde{R}_p = L_p + R_p \quad (2)$$

[0019] In equation 1, the observed intensity  $\tilde{I}_p$  for pixel  $p$  is expressed in the intensity domain. As shown in equation 1, the observed intensity  $\tilde{I}_p$  for pixel  $p$  in the intensity domain is the product of the luminance  $\tilde{L}_p$  of the pixel  $p$  and the reflectance  $\tilde{R}_p$  of pixel  $p$ . In contrast, the observed intensity  $I_p$  for pixel  $p$  is expressed in the log domain in equation 2. As shown in equation 2, the observed intensity  $I_p$  for pixel  $p$  in the log domain is the sum of the log of the luminance  $\tilde{L}_p$  and the log of the reflectance  $\tilde{R}_p$  of pixel  $p$ . In the following discussion, the observed intensities for pixels will be discussed with respect to the log domain.

[0020] Generally, the image processing server 100 estimates values of illumination (i.e., illumination intensities) present in an illuminated image using an information theoretic approach that recognizes that any illumination change in an image tends to increase the diversity of observed texture intensities of the reflectance of the image. In one embodiment, the texture at a given pixel location is described as the appearance of the local neighborhood around that location. The information theoretic interpretation of this effect is that the entropy of the texture is increased. Similarly, the presence of texture

in the image increases the entropy of the illumination function, which is otherwise mostly smooth except at illumination boundaries.

**[0021]** In one embodiment, the image processing server 100 estimates an illumination value  $L$  of a pixel  $p$  from an illuminated image based on the illumination values of randomly selected pixels from the image and the illumination values of neighboring pixels of pixel  $p$ . In one embodiment, the image processing server 100 minimizes an energy function using the illumination values of the randomly selected pixels and the illumination values of the neighboring pixels to determine the illumination value  $L$  of pixel  $p$ . Thus, the image processing server 100 can estimate the illumination values  $L$  that contribute to the pixel intensities  $I_p$  in an illuminated image. Alternatively, the illumination value  $L$  of a pixel  $p$  may be estimated based on all pixels in the illuminated image.

**[0022]** In one embodiment, the image processing server 100 enhances an illuminated image by removing illumination variation from the image based on the estimated illumination values of the pixels in the image. That is, the image processing server 100 corrects the illumination variation present in the image so that the intensity of the illumination in the image is uniform. By correcting the illumination variation in the image, the image processing server 100 removes any shadows that were previously present in the image. Thus, the values of illumination in the illumination map are zero or as close to zero as possible. In other words, the variation in the illumination map is minimized as much as possible. Values of the illumination map may be equal to zero (in the log domain) if the illumination of the brightest part of an illuminated image is set to zero.

**[0023]** As illustrated in FIG. 1, the image processing server 100 comprises various modules including an initialization module 101, a training module 103, a similarity module 105, an image enhancement module 107, an optimization module 109, and an image database 111. As is known in the art, the term “module” refers to computer program logic utilized to provide the specified functionality. Thus, a module can be implemented in hardware, firmware, and/or software. In one embodiment, program modules are stored on a non-transitory storage device (i.e., a computer program product), loaded into a memory, and executed by a computer processor. Additionally, those of skill in the art will recognize that other embodiments of the image processing server 100

shown in FIG. 1 can have different and/or other modules than the ones described here, and that the functionalities can be distributed among the modules in a different manner.

**[0024]** The image database 111 stores a set of illuminated images. The term “image” as employed herein includes image data in general, such as individual still image files or frames of images from a video file. In one embodiment, the images in the image database 111 may be obtained from a variety of sources, such as from an organization producing aerial or street view images of a portion of a city for use with a map system, such as GOOGLE MAPS, STREET VIEW or GOOGLE EARTH. Likewise, images (including videos) may be obtained from users, e.g. as submitted by a user uploading images to a photo repository such as PICASA WEB, or a video repository such as YOUTUBE or GOOGLE VIDEO.

**[0025]** In one embodiment, at least some of the illuminated images stored in the image database 111 are aerial images. An aerial image comprises an image taken off the ground from an elevated position such as from an aircraft. These aerial images may include shadows caused by clouds or other objects. In one embodiment, at least some of the illuminated images stored in the image database 111 exhibit a repetitive pattern. For example, an image may show a façade of a building with a repeating pattern of windows or an image may show a rug that exhibits a repetitive pattern in its design.

**[0026]** Additionally, the image database 111 stores enhanced versions of the illuminated images. In one embodiment, an enhanced version of an illuminated image comprises only the underlying non-illuminated image. That is, the image database 111 stores versions of the images with the variations in the illumination map removed, so that the illumination across the image is uniform and any shadows or other illumination-related artifacts are absent.

**[0027]** The initialization module 101 initializes the image processing server 100 with initial illumination values that can be used to remove illuminations from illuminated images. For a given image, the initialization module 101 determines an initial illumination value that can be used to remove the illuminations from the image. For a given pixel in the image, the image processing server 100 attempts to improve upon the initial illumination value determined by the initialization module 101 until an illumination value  $L$  is determined that removes or reduces the illumination from the pixel.

[0028] Generally, the initialization module 101 automatically specifies, for an illuminated image, a mask comprising only non-illuminated (e.g., non-shadowed) pixels. The initialization module 101 constrains the illumination for the pixels in the mask such that the luminance  $L$  is zero.

[0029] To identify the mask, the initialization module 101 applies k-means clustering (e.g.,  $k$  is equal to 4) to the illuminated image. As a result, the illuminated image is divided into a plurality of pixel clusters where each cluster has an associated scalar illumination value corresponding to the average scalar illumination value of the pixels in the cluster. The initialization module 101 identifies a pixel cluster from the plurality of pixel clusters that represents non-illuminated pixels of the image based on the scalar illumination values of the clusters.

[0030] In one embodiment, for façade images with periodic patterns, the initialization module 101 applies a box filter to the image to divide the image into a plurality of pixel clusters. For an illuminated façade image, the initialization module applies the box filter to the image and determines the average scalar illumination value of the pixels within the box defined by the filter. The initialization module 101 identifies the plurality of pixel clusters based on the average scalar illumination values of the illuminated façade image. According to one embodiment, the size of the box filter in the  $x$  and  $y$  dimensions (e.g., 15 pixels by 17 pixels) is equivalent to the period of the repetitive pattern in the  $x$  and  $y$  dimensions for the illuminated image.

[0031] To identify the pixel cluster representing non-illuminated pixels of the image, the initialization module 101 extracts features from the plurality of clusters to form feature vectors that describes the clusters. In one embodiment, the extracted features comprise the mean scalar illumination value  $\bar{L}_i$  of the pixels  $i$  from the cluster, the cluster size (i.e., number of pixels in the cluster), the variance of the scalar illumination values of the pixels in the cluster, the mean intensity value  $\bar{I}_i$  of the image, the mean height (in pixels) of the clusters, and the normalized mean intensity value of other clusters  $j$  that are not selected as the non-illuminated cluster  $i$  such that:  $\bar{I}_j \cdot \frac{L_i}{L_j}$  where

pixel  $i \neq$  pixel  $j$ . The intuition is that if the selected pixel cluster  $i$  represents the non-illuminated cluster, the illumination value  $L_i$  and the illumination value  $L_j$  together would normalize the pixels from the other clusters, including illuminated clusters, to the right

intensity.

**[0032]** When applied to an image—or more specifically, to the feature vectors of the image—the initialization module 101 generates information that describes a measure of how strongly each cluster represents a cluster of non-illuminated pixels. To generate the information, the initialization module 101 applies a set of weights that are associated with features of non-illuminated pixels to a feature vector. Based on the applied weights and the feature vector, the initialization module 101 calculates a score which may be a Boolean score representing whether or not a given cluster represents a non-illuminated pixel cluster. Alternatively, the score is a real number (e.g., ranging from 0.0 to 1.0), integer, or other scalar value representing a measure of likelihood or probability that the cluster represents a non-illuminated pixel cluster. In one embodiment, numbers or scores exceeding some threshold can be considered to indicate that the cluster represents a non-illuminated pixel cluster. In one embodiment, the cluster with a highest score that exceeds the threshold is considered the non-illuminated pixel cluster and the illumination value  $\bar{L}_i$  representing the mean illumination values of the identified cluster is used to initialize the image processing server 100 when estimating the illumination values of the image.

**[0033]** In one embodiment, the training module 103 trains the initialization module 101 to identify pixel clusters that represent non-illuminated pixels using a training set of illuminated images (e.g., 90 images). Each image from the training set includes an indication (e.g., a label) of non-illuminated (e.g., non-shadowed) portions of the image. The training module 103 identifies pixel clusters of the image and extracts the features of the image as previously described above. In one embodiment, the training module 103 assigns a positive label (e.g., +1) to the dominant pixel cluster (i.e., the largest pixel cluster) representing the non-illuminated portion of the image which was designated in the image and assigns a negative label (e.g., -1) to all other clusters. The pixel clusters assigned the negative label include other pixel clusters representing non-illuminated portions of the image and other illuminated portions of the image which are not dominant.

**[0034]** The training module 103 applies a training algorithm to the initialization module 101 to learn the set of weights on the features of the training set that are associated with the dominant pixel cluster as previously described above so that the initialization module 101 can recognize non-illuminated portions of illuminated images.

In one embodiment, the training algorithm is AdaBoost, the details of which are known to one having ordinary skill in the art.

**[0035]** The similarity module 105 determines the similarity of pixels in illuminated images. The similarity of pixels in an illuminated image is used to determine the illumination values which cause the illumination in the image. The similarity of pixels can be used to determine the illumination values because it is assumed that for pixels with a similar underlying reflectance (i.e., similar pixels), the difference in pixel intensity of the illuminated pixel and the non-illuminated pixel in the log domain (or ratios in the intensity domain) is due to the difference in the illumination values of the pixels.

**[0036]** To determine the similarity, the similarity module 105 identifies local pixel neighbors for each pixel  $i$  in an illuminated image. In one embodiment, the local pixel neighbors of a given pixel  $i$  comprise pixels that are spatially located directly adjacent to the pixel  $i$ . Pixel  $i$  and its local pixel neighbors form a patch  $P_i$  with pixel  $i$  located at the center of patch  $P_i$  in one embodiment. For example, Fig. 2 illustrates an illuminated image 200 and patches  $P_{1753}$ ,  $P_{1352}$ , and  $P_{2233}$ . Patch  $P_{1753}$  illustrates the local pixel neighbors for pixel  $p_{1753}$  which is located at the center of patch  $P_{1753}$ . Similarly, patch  $P_{1352}$  illustrates the pixel neighbors for pixel  $p_{1352}$  which is located at the center of patch  $P_{1352}$  and patch  $P_{2233}$  illustrates the local pixel neighbors for pixel  $p_{2233}$  which is located at the center of patch  $P_{2233}$ .

**[0037]** Referring back to Fig. 1, the similarity module 105 calculates for each patch  $P_i$ , the reflectance  $R_i$  for pixel  $i$  located at the center of the patch  $P_i$  according to the following equation:

$$R_i = I_i - L_P \quad (3)$$

In equation 3, the reflectance  $R_i$  for the pixel  $i$  located at the center of the patch  $P_i$  is a function of the intensity of the pixel (which is a known value) and luminance. Note that equation 3 assumes that the illumination value  $L_P$  is constant for patch  $P_i$  except for at the boundaries of the patch. That is, equation 3 assumes that the illumination value  $L_P$  is constant across all the pixels included in patch  $P_i$ .

**[0038]** In one embodiment, the similarity module 105 identifies a plurality of random pixels  $j$  and their local pixel neighbors to compare against pixel  $i$ . In one embodiment, the random pixels  $j$  represent non-illuminated pixels (i.e., non-shadowed). Each random pixel  $j$  and its local pixel neighbors collectively form a patch  $P_j$  as

previously described above. The similarity module 105 calculates the reflectance  $R_j$  for the pixel  $j$  located at the center of each patch  $P_j$  as described with respect to equation 3.

**[0039]** The similarity module 105 compares pairs of patches to determine the similarity of the patches with respect to reflectance. That is, the similarity module compares patch  $P_i$  with each of the plurality of patches  $P_j$  to determine the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$ . In one embodiment, the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  describes the difference in reflectance of the patches  $P_i$  and  $P_j$ . In one embodiment, the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  is represented by the following equation:

$$C(P_i, P_j) = |R_i - R_j|^2 \quad (4)$$

As shown in equation 4, the similarity module 105 may determine the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  based on the difference of the reflectance  $R_i$  of pixel  $i$  from patch  $P_i$  and the reflectance  $R_j$  of pixel  $j$  from patch  $P_j$ .

**[0040]** Alternatively, the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  is represented by the following equation:

$$C(P_i, P_j) = \sum_k (R_{ik} - R_{jk})^2 \quad (5)$$

where  $k$  represents each corresponding pixel from patch  $P_i$  and patch  $P_j$ .

**[0041]** In equation 5, the similarity module 105 determines the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  based on the reflectance value of all pixels in patch  $P_i$  and patch  $P_j$ . Equation 5 illustrates that the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  is based on the summation of the difference of reflectance values across corresponding pairs of pixels from patch  $P_i$  and patch  $P_j$ . For example, for each pixel  $k$  in patch  $P_i$  and its corresponding pixel  $k$  in patch  $P_j$ , the similarity module 105 determines the difference of the reflectance values of the pair of pixels. The similarity module 105 then sums the square of the differences.

**[0042]** In another embodiment, the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  is represented by the following equation:

$$C(P_i, P_j) = |Hist(P_i) - Hist(P_j)|^2 \quad (6)$$

**[0043]** The similarity module 105 may determine a histogram of the reflectance values of pixels within patch  $P_i$  which is represented by  $Hist(P_i)$  in equation 6. The histogram describes the distribution of the reflectance values of pixels within patch  $P_i$ .

The similarity module 105 also determines a histogram of the reflectance values of pixels within patch  $P_j$  which is represented by  $Hist(P_j)$ . In equation 6, the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  is based on the difference between the distribution of the reflectance values of pixels within patch  $P_i$  and patch  $P_j$ .

**[0044]** In one embodiment, the similarity module 105 determines a weight  $W_{ij}$  for the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$ . The weight  $W_{ij}$  describes the influence that the similarity  $C(P_i, P_j)$  between patch  $P_i$  and patch  $P_j$  has on the determination of the luminance value used to remove the illumination from pixel  $p_i$ . In one embodiment, the weight  $W_{ij}$  is represented by the following equation:

$$W_{ij} = e^{-C(P_i, P_j)} \quad (7)$$

**[0045]** Because the weight is based on a decaying exponential function, the similarity module 105 assigns a higher weight to the similarity  $C(P_i, P_j)$  of patch  $P_i$  and patch  $P_j$  if the patches are similar (i.e., a smaller value of  $C(P_i, P_j)$ ) and the similarity module 105 assigns a lower weight to the similarity  $C(P_i, P_j)$  of patch  $P_i$  and patch  $P_j$  if the patches are not similar (i.e., a larger value of  $C(P_i, P_j)$ ).

**[0046]** Referring now to Fig. 2, the following description is an example of how to determine weights for the pixel patches of image 200. The similarity module 105 selects pixel  $p_{1753}$ , which is an illuminated pixel. The similarity module 105 identifies the local pixel neighbors for pixel  $p_{1753}$  which is represented by patch  $P_{1753}$ . The similarity module 105 calculates the pixel intensity  $P_{1753}I_{1753}$  for pixel  $p_{1753}$  according to equation 3 and the mean pixel intensity  $P_{1753}I_M$  for patch  $P_{1753}$  according to equation 4. Based on the pixel intensity  $P_{1753}I_{1753}$  and the mean pixel intensity  $P_{1753}I_M$ , the similarity module 105 calculates the similarity  $P_{1753}^M$  of the pixel intensity  $P_{1753}I_{1753}$  and the mean pixel intensity  $P_{1753}I_M$  according to equation 5.

**[0047]** As described previously, the similarity module 105 also identifies random pixels and their associated patches for comparison against a given pixel. In Fig. 2, the similarity module 105 may randomly identify pixel  $p_{1352}$  and pixel  $p_{2233}$ . Note that the similarity module 105 may identify more than two pixels to compare against pixel  $p_{1753}$ . However, for ease of description only two randomly selected pixels are illustrated in this example.

**[0048]** The similarity module 105 respectively identifies patch  $P_{1352}$  and patch  $P_{2233}$

as the local pixel neighbors for pixel  $p_{1352}$  and pixel  $p_{2233}$ . The similarity module 105 calculates the pixel intensities  $P_{1352}I_{1352}$  and  $P_{2233}I_{2233}$  for pixels  $p_{1352}$  and  $p_{2233}$  and the mean pixel intensities  $P_{1352}I_M$  and  $P_{2233}I_M$  for patch  $P_{1352}$  and  $P_{2233}$  according to equations 3 and 4.

[0049] Based on the pixel intensities  $P_{1352}I_{1352}$  and  $P_{2233}I_{2233}$  and the mean pixel intensities  $P_{1352}I_M$  and  $P_{2233}I_M$ , the similarity module 105 calculates the similarity  $P_{1352}^M$  of the pixel intensity  $P_{1352}I_{1352}$  and the mean pixel intensity  $P_{1352}I_M$  and the similarity  $P_{2233}^M$  of the pixel intensity  $P_{2233}I_{2233}$  and the mean pixel intensity  $P_{2233}I_M$  according to equation 5. The similarity module 105 determines the respective similarities  $C(P_{1753}, P_{1352})$  and  $C(P_{1753}, P_{2233})$  between patch  $P_{1753}$  and patches  $P_{1352}$  and  $P_{2233}$  and their associated weights according to equations 6 and 7.

[0050] Referring back to Fig. 1, the optimization module 109 applies an energy function  $E$  to solve for illumination values for pixels in an illuminated image. Particularly, the optimization module 109 applies the energy function  $E$  to solve for the value of  $L_i$  of pixel  $i$  in order to remove the illumination from pixel  $i$ . In one embodiment, the energy function is represented by the following equation:

$$E = \sum_i \sum_j W_{ij} (L_i - L_j - I_{ij})^2 + \sum V_{ik} (L_i - L_k)^2 \quad (8)$$

where:

- $I_{ij} = I_i - I_j$ ; and
- $V_{ik} = \sum e^{-(L_i - L_k)^2}$  where  $k$  represents a local pixel neighbor of pixel  $i$ .

[0051] As shown above, the energy  $E$  is a sum of two terms. The first term measures the difference between luminance values of a given pixel  $i$  of patch  $P_i$  and a random pixel  $j$  from patch  $P_j$  across the plurality of randomly selected pixels  $j$  for pixel  $i$ . The weight  $W_{ij}$  controls the influence that the patches from which pixel  $i$  and pixel  $j$  are located in the determination of the value of  $L_i$ . Typically, the pixel intensity  $I_{ij}$  of a given pixel is known and is subtracted from the luminance values of pixel  $i$  and pixel  $j$ .

[0052] In one embodiment, the second term of the energy function smoothes the illumination values within a region of the image corresponding to patch  $P_i$  (i.e., the local pixel neighbors of pixel  $i$ ). The optimization module 105 operates under the assumption that the illumination values of pixels within patch  $P_i$  should be of equivalent or similar

intensity. In the second term of the energy function, the optimization module 109 compares illumination values of pixel  $i$ 's local pixel neighbors  $k$ . The second term of the energy function is based on the summation of the difference between the illumination value  $L_i$  of pixel  $i$  and the illumination values  $L_k$  of neighboring pixel instances.

**[0053]** Furthermore, the second term of the energy function  $E$  is multiplied by a weight  $V_{ik}$ . In one embodiment, weight  $V_{ik}$  controls the smoothing factor of the illumination values represented by the second term of the energy function  $E$  and is based on a decaying exponential function. Because the weight  $V_{ik}$  is based on a decaying exponential function, the smaller the variance in the luminance values between pixel  $i$  and neighboring pixels  $k$ , the higher the weight  $V_{ik}$ .

**[0054]** In one embodiment, to solve for the illumination values  $L_i$ , the optimization module 109 initializes the solution for the energy function  $E$  using the initial luminance values determined by the initialization module 101. The optimization module 109 solves for the illumination values  $L_i$  using the initial luminance values and iteratively optimizes the energy function  $E$  for each pixel location to solve for the illumination value that minimizes the energy function for all locations in order to improve upon the values provided by the initialization module 101.

**[0055]** In one embodiment, the optimization module 109 applies the technique of iteratively re-weighted least squares to solve for the illumination values thereby producing sharper changes in the illumination map. The optimization module 109 creates a matrix based on the weights  $W_{ij}$  and  $V_{ik}$  that represents the illumination map responsive to solving the energy function. The matrix comprises illumination values for every pixel location in the illuminated image. The illumination values included in the matrix are associated with the weights indicative of the most similar pixels (e.g., the weights with the highest value). Each illumination value in the matrix describes the intensity of the illumination at a particular location in the image.

**[0056]** The image enhancement module 107 enhances illuminated images based on the optimized illumination values described in the matrix. In one embodiment, the image enhancement module 107 enhances an illuminated image by removing illumination variations from the image. To remove the illumination variation (i.e., correct the varying illumination in the image), the optimization module 107 multiplies the pixel intensity of each pixel in the illuminated image in the intensity domain by the inverse of the

optimized illumination value for that pixel as determined by the optimization module 109, thereby removing the value of the illumination component from the pixel. By multiplying the inverse of the optimized illumination values for the pixels of the illuminated image by the pixel intensities of the pixels, the optimization module 109 removes the illumination map component of the illuminated image. The enhanced image has uniform intensity such that any shadows or other illumination-related artifacts are absent from the image. The image enhancement module 107 stores the enhanced image in the image database 111.

**[0057]** Referring now to Fig. 3, image 300 illustrates an enhanced image of illuminated image 200 illustrated in Fig. 2. As shown in image 300, the illumination (i.e., the shadow) from image 200 has been removed. Image 300 illustrates uniform pixel intensity of pixels across the image 300.

**[0058]** Fig. 4 is a flowchart illustrating steps performed by the image processing server 100 to remove illumination variation from an illuminated image according to one embodiment. Other embodiments perform the illustrated steps in different orders, and/or perform different or additional steps.

**[0059]** In one embodiment, the image processing server 100 selects 401 from the image database 111 an illuminated image comprising varying illumination. The selected image may represent an aerial image, an image exhibiting a repetitive pattern, or any image that includes varying illumination. The image processing server 100 calculates 403 an illumination value to initialize the image processing server 100 that will be used as a starting point to determine the illumination values of the selected image.

**[0060]** For each pixel in the image, the image processing server 100 identifies 405 patches of pixels for the given pixel. The patches may include a patch associated with the local neighbor pixels and patches of randomly selected pixels. The image processing server 100 calculates the similarity 407 between the patches. That is, the image processing server 100 calculates the similarity of pairs of patches where each pair includes a patch comprising the given pixel and a patch associated with a randomly selected pixel. The image processing server 100 then optimizes an energy function based on the calculated similarity to produce an illumination value for each pixel in the image. The image processing server 100 removes 411 the illumination variation from the image by multiplying the intensity of each pixel with its corresponding inversed illumination

value determined in step 409.

**[0061]** FIG. 5 is a high-level block diagram of a computer 500 for acting as an image processing server 100 according to one embodiment. Illustrated are at least one processor 502 coupled to a chipset 504. Also coupled to the chipset 504 are a memory 506, a storage device 508, a keyboard 510, a graphics adapter 512, a pointing device 514, and a network adapter 516. A display 518 is coupled to the graphics adapter 512. In one embodiment, the functionality of the chipset 504 is provided by a memory controller hub 520 and an I/O controller hub 522. In another embodiment, the memory 506 is coupled directly to the processor 502 instead of the chipset 504.

**[0062]** The storage device 508 is any non-transitory computer-readable storage medium, such as a hard drive, compact disk read-only memory (CD-ROM), DVD, or a solid-state memory device. The memory 506 holds instructions and data used by the processor 502. The pointing device 514 may be a mouse, track ball, or other type of pointing device, and is used in combination with the keyboard 510 to input data into the computer system 500. The graphics adapter 512 displays images and other information on the display 518. The network adapter 516 couples the computer system 500 to a local or wide area network.

**[0063]** As is known in the art, a computer 500 can have different and/or other components than those shown in FIG. 5. In addition, the computer 500 can lack certain illustrated components. In one embodiment, a computer 500 acting as an image processing server 100 lacks a keyboard 510, pointing device 514, graphics adapter 512, and/or display 518. Moreover, the storage device 508 can be local and/or remote from the computer 500 (such as embodied within a storage area network (SAN)). As is known in the art, the computer 500 is adapted to execute computer program modules for providing functionality previously described herein. In one embodiment, program modules are stored on the storage device 508, loaded into the memory 506, and executed by the processor 502.

**[0064]** The disclosure herein has been described in particular detail with respect to one possible embodiment. Those of skill in the art will appreciate that other embodiments may be practiced. First, the particular naming of the components and variables, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant, and the mechanisms

that implement the invention or its features may have different names, formats, or protocols. Also, the particular division of functionality between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead performed by a single component.

**[0065]** Some portions of above description present features in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. These operations, while described functionally or logically, are understood to be implemented by computer programs. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules or by functional names, without loss of generality.

**[0066]** Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as “determining” or “displaying” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

**[0067]** Certain aspects of the embodiments disclosed herein include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions could be embodied in software, firmware or hardware, and when embodied in software, could be downloaded to reside on and be operated from different platforms used by real time network operating systems.

**[0068]** The algorithms and operations presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will be apparent to those of skill in the art, along with equivalent variations. In addition, the present invention is not described with reference to any particular programming language. It is appreciated that a variety of

programming languages may be used to implement the teachings of the present invention as described herein, and any references to specific languages are provided for invention of enablement and best mode of the present invention.

**[0069]** The embodiments disclosed herein are well suited to a wide variety of computer network systems over numerous topologies. Within this field, the configuration and management of large networks comprise storage devices and computers that are communicatively coupled to dissimilar computers and storage devices over a network, such as the Internet.

**[0070]** Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure herein is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

What is claimed is:

1. A computer-implemented method of removing illumination variation from an image, the method performed by a computer and comprising:
  - selecting an image containing varying illumination across a plurality of pixels within the image;
  - identifying patches of pixels from among the plurality of pixels within the image;
  - calculating similarities between pairs of patches of pixels based on pixel intensities associated with the pairs of patches of pixels;
  - calculating illumination values for the plurality of pixels within the image based on the calculated similarities between the pairs of patches of pixels; and
  - removing the illumination variation from the image based on the calculated illumination values for the plurality of pixels within the image.
2. The computer-implemented method of claim 1, wherein identifying the patches of pixels comprises:
  - determining a plurality of adjacent pixels that are adjacent to a pixel in the image;
  - and
  - assigning the plurality of adjacent pixels and the pixel to a patch that is associated with the pixel.
3. The computer-implemented method of claim 2, wherein calculating the similarities comprises:
  - randomly selecting a second pixel from the image;
  - determining a second plurality of adjacent pixels that are adjacent to the second pixel in the image;
  - assigning the second plurality of pixels and the second pixel to a second patch that is associated with the second pixel; and
  - comparing the patch and the second patch based on a similarity of the pixel and the first patch and a similarity of the second pixel and the second patch.
4. The computer-implemented method of claim 1, wherein calculating the similarities between the pairs of patches comprises, for each patch:
  - calculating a reflectance value of a pixel included in the patch.
5. The computer-implemented method of claim 4, further comprising:
  - calculating a similarity between a first patch and a second patch based on a

difference of reflectance values associated with the first patch and the second patch; and

calculating a weight for the similarity between the first patch and the second patch, the weight describing an influence that the similarity between the first patch and the second patch has on the calculation of the illumination values for the plurality of pixels within the image.

6. The computer-implemented method of claim 5, wherein calculating the similarity between the first patch and the second patch comprises:

calculating a difference of a first reflectance value of a first pixel from the first patch and a second reflectance value of a second pixel from the second patch, wherein the first pixel is located at a center of the first patch and the second pixel is located at a center of the second patch.

7. The computer-implemented method of claim 5, wherein calculating the similarity between the first patch and the second patch comprises for each corresponding pair of pixels from the first patch and the second patch:

calculating a difference of a first reflectance value of a first pixel from the first patch and a second reflectance value of a second pixel from the second patch, wherein the first pixel and second pixel are respectively located at a same position in the first patch and the second patch.

8. The computer-implemented method of claim 5, wherein calculating the similarity between the first patch and the second patch comprises:

calculating a difference of a first distribution of reflectance values of pixels within the first patch and a second distribution of reflectance values of pixels within the second patch.

9. The computer-implemented method of claim 1, wherein calculating the illumination values comprises:

calculating an illumination value for a pixel within the image based on a difference between the illumination value for the pixel and an illumination value for a second pixel from one of the identified patches and a difference between a pixel intensity of the pixel and a pixel intensity of the second pixel.

10. The computer-implemented method of claim 9, wherein calculating the illumination

values comprises:

optimizing an energy function to calculate the illumination value for the pixel within the image based on the difference between the illumination value for the pixel and the illumination value for the second pixel from one of the identified patches and the difference between the pixel intensity of the pixel and the pixel intensity of the second pixel and further based on a difference between the illumination value for the pixel and an illumination value for each third pixel that is directly adjacent to the pixel in the image; and

wherein an illumination value for a defined region of the image is used as an initial illumination value to optimize the energy function.

11. The computer-implemented method of claim 1, wherein the calculated illumination values are inverse illumination values and wherein removing the illumination variation from the image comprises:

for each pixel in the image, multiplying a pixel intensity of the pixel by an inverse illumination value calculated for the pixel.

12. A non-transitory computer-readable storage medium storing computer-executable code for removing illumination variation from an image, the code executable to perform steps comprising:

selecting an image containing varying illumination across a plurality of pixels within the image;

identifying patches of pixels from among the plurality of pixels within the image;

calculating similarities between pairs of patches of pixels based on pixel intensities associated with the pairs of patches of pixels;

calculating illumination values for the plurality of pixels within the image based on the calculated similarities between the pairs of patches of pixels; and

removing the illumination variation from the image based on the calculated illumination values for the plurality of pixels within the image.

13. The non-transitory computer-readable storage medium of claim 12, wherein the code is executable to perform further steps comprising:

determining a plurality of adjacent pixels that are adjacent to a pixel in the image; and

assigning the plurality of adjacent pixels and the pixel to a patch that is associated with the pixel.

14. The non-transitory computer-readable storage medium of claim 12, wherein the code is executable to perform further steps comprising, for each patch:

calculating a reflectance value of a pixel included in the patch.

15. The non-transitory computer-readable storage medium of claim 14, wherein the code is executable to perform further steps comprising:

calculating a similarity between a first patch and a second patch based on a difference of reflectance values associated with the first patch and the second patch; and

calculating a weight for the similarity between the first patch and the second patch, the weight describing an influence that the similarity between the first patch and the second patch has on the calculation of the illumination values for the plurality of pixels within the image.

16. The non-transitory computer-readable storage medium of claim 15, wherein calculating the similarity between the first patch and the second patch comprises:

calculating a difference of a first reflectance value of a first pixel from the first patch and a second reflectance value of a second pixel from the second patch, wherein the first pixel is located at a center of the first patch and the second pixel is located at a center of the second patch.

17. The non-transitory computer-readable storage medium of claim 15, wherein calculating the similarity between the first patch and the second patch comprises for each corresponding pair of pixels from the first patch and the second patch:

calculating a difference of a first reflectance value of a first pixel from the first patch and a second reflectance value of a second pixel from the second patch, wherein the first pixel and second pixel are respectively located at a same position in the first patch and the second patch.

18. The non-transitory computer-readable storage medium of claim 15, wherein calculating the similarity between the first patch and the second patch comprises:

calculating a difference of a first distribution of reflectance values of pixels within the first patch and a second distribution of reflectance values of pixels within the second patch.

19. The non-transitory computer-readable storage medium of claim 12, wherein the code is executable to perform further steps comprising:

calculating an illumination value for a pixel within the image based on a difference between the illumination value for the pixel and an illumination value for a second pixel from one of the identified patches and a difference between a pixel intensity of the pixel and a pixel intensity of the second pixel.

20. The non-transitory computer-readable storage medium of claim 19, wherein the code is executable to perform further steps comprising:

optimizing an energy function to calculate the illumination value for the pixel within the image based on the difference between the illumination value for the pixel and the illumination value for the second pixel from one of the identified patches and the difference between the pixel intensity of the pixel and the pixel intensity of the second pixel and further based on a difference between the illumination value for the pixel and an illumination value for each third pixel that is directly adjacent to the pixel in the image; and

wherein an illumination value for a defined region of the image is used as an initial illumination value to optimize the energy function.

21. The non-transitory computer-readable storage medium of claim 12, wherein the calculated illumination values are inverse illumination values and wherein the code is executable to perform further steps comprising:

for each pixel in the image, multiplying a pixel intensity of the pixel by an inverse illumination value calculated for the pixel.

22. A computer system for removing illumination variation from an image, the system comprising:

a computer processor; and

a non-transitory computer-readable storage medium storing executable instructions configured to execute on the computer processor, the instructions when executed by the computer processor are configured to perform steps comprising:

selecting an image containing varying illumination across a plurality of pixels within the image;  
identifying patches of pixels from among the plurality of pixels within the image;  
calculating similarities between pairs of patches of pixels based on pixel intensities associated with the pairs of patches of pixels;  
calculating illumination values for the plurality of pixels within the image based on the calculated similarities between the pairs of patches of pixels; and  
removing the illumination variation from the image based on the calculated illumination values for the plurality of pixels within the image.

23. The computer system of claim 22, wherein the instructions when executed by the computer processor are configured to perform further steps comprising:

determining a plurality of adjacent pixels that are adjacent to a pixel in the image;  
and  
assigning the plurality of adjacent pixels and the pixel to a patch that is associated with the pixel.

24. The computer system of claim 22, wherein the instructions when executed by the computer processor are configured to perform further steps comprising for each patch:

calculating a reflectance value of a pixel included in the patch.

25. The computer system of claim 24, wherein the instructions when executed by the computer processor are configured to perform further steps comprising:

calculating a similarity between a first patch and a second patch based on a difference of reflectance values associated with the first patch and the second patch; and  
calculating a weight for the similarity between the first patch and the second patch, the weight describing an influence that the similarity between the first patch and the second patch has on the calculation of the illumination values for the plurality of pixels within the image.

26. The computer system of claim 25, wherein calculating the similarity between the first patch and the second patch comprises:

calculating a difference of a first reflectance value of a first pixel from the first patch and a second reflectance value of a second pixel from the second patch, wherein the first pixel is located at a center of the first patch and the second pixel is located at a center of the second patch.

27. The computer system of claim 25, wherein calculating the similarity between the first patch and the second patch comprises for each corresponding pair of pixels from the first patch and the second patch:

calculating a difference of a first reflectance value of a first pixel from the first patch and a second reflectance value of a second pixel from the second patch, wherein the first pixel and second pixel are respectively located at a same position in the first patch and the second patch.

28. The computer system of claim 25, wherein calculating the similarity between the first patch and the second patch comprises:

calculating a difference of a first distribution of reflectance values of pixels within the first patch and a second distribution of reflectance values of pixels within the second patch.

29. The computer system of claim 22, wherein the instructions when executed by the computer processor are configured to perform further steps comprising:

calculating an illumination value for a pixel within the image based on a difference between the illumination value for the pixel and an illumination value for a second pixel from one of the identified patches and a difference between a pixel intensity of the pixel and a pixel intensity of the second pixel.

30. The computer system of claim 29, wherein the instructions when executed by the computer processor are configured to perform further steps comprising:

optimizing an energy function to calculate the illumination value for the pixel within the image based on the difference between the illumination value for the pixel and the illumination value for the second pixel from one of the identified patches and the difference between the pixel intensity of the pixel and the pixel intensity of the second pixel and further based on a difference between the illumination value for the pixel and an illumination value for each third pixel that is directly adjacent to the pixel in the image;

and

wherein an illumination value for a defined region of the image is used as an initial illumination value to optimize the energy function.

31. The computer system of claim 22, wherein the instructions when executed by the computer processor are configured to perform further steps comprising:

for each pixel in the image, multiplying a pixel intensity of the pixel by an inverse illumination value calculated for the pixel.

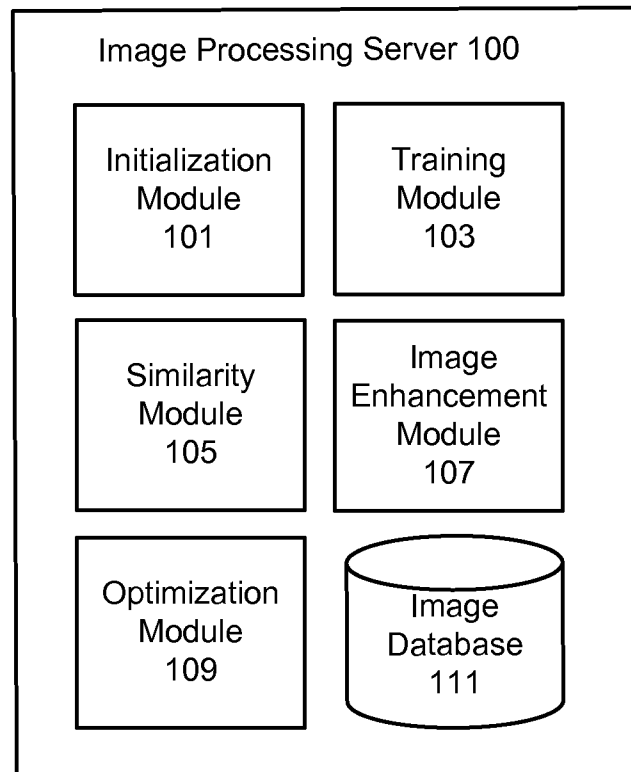


Fig. 1

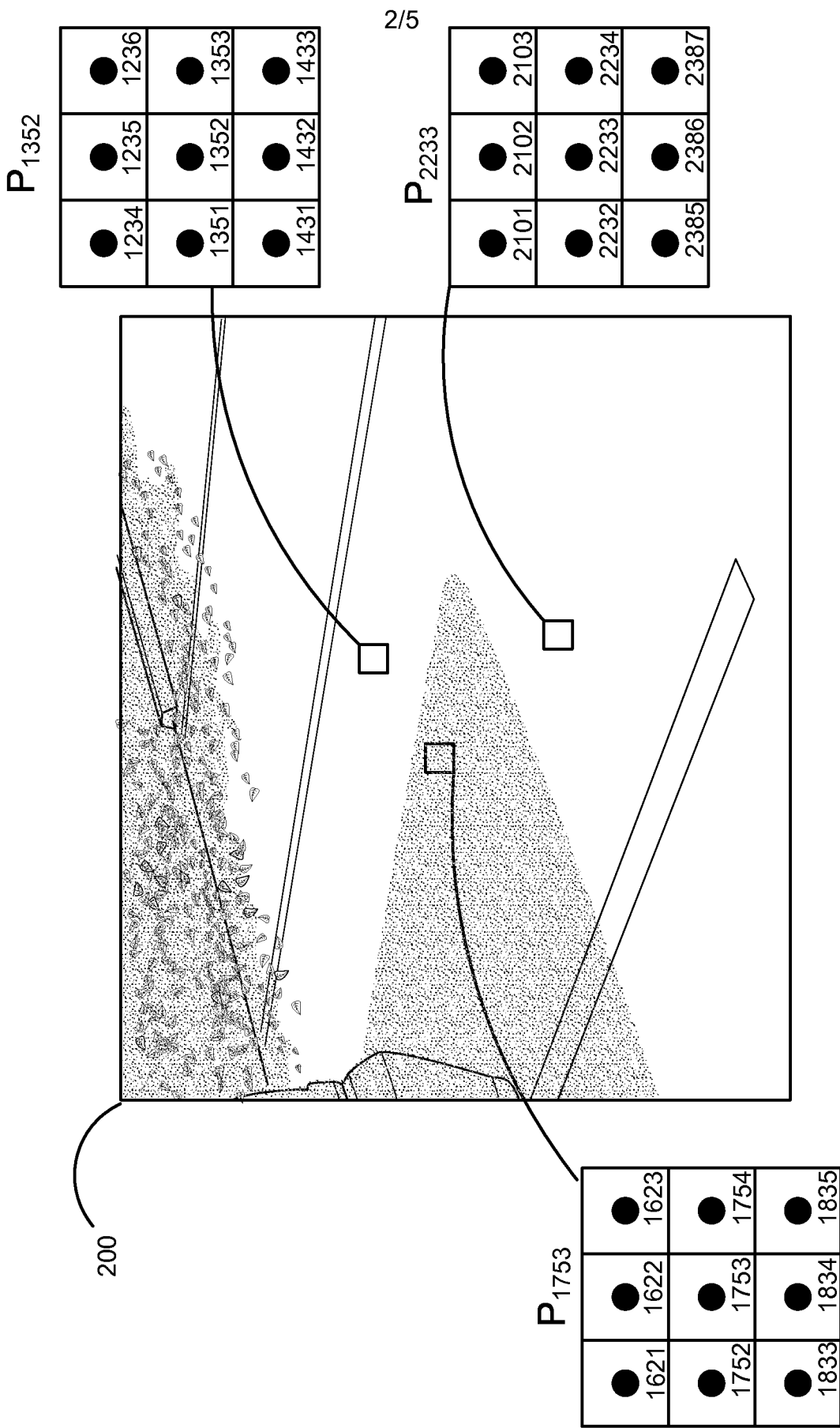


Fig. 2

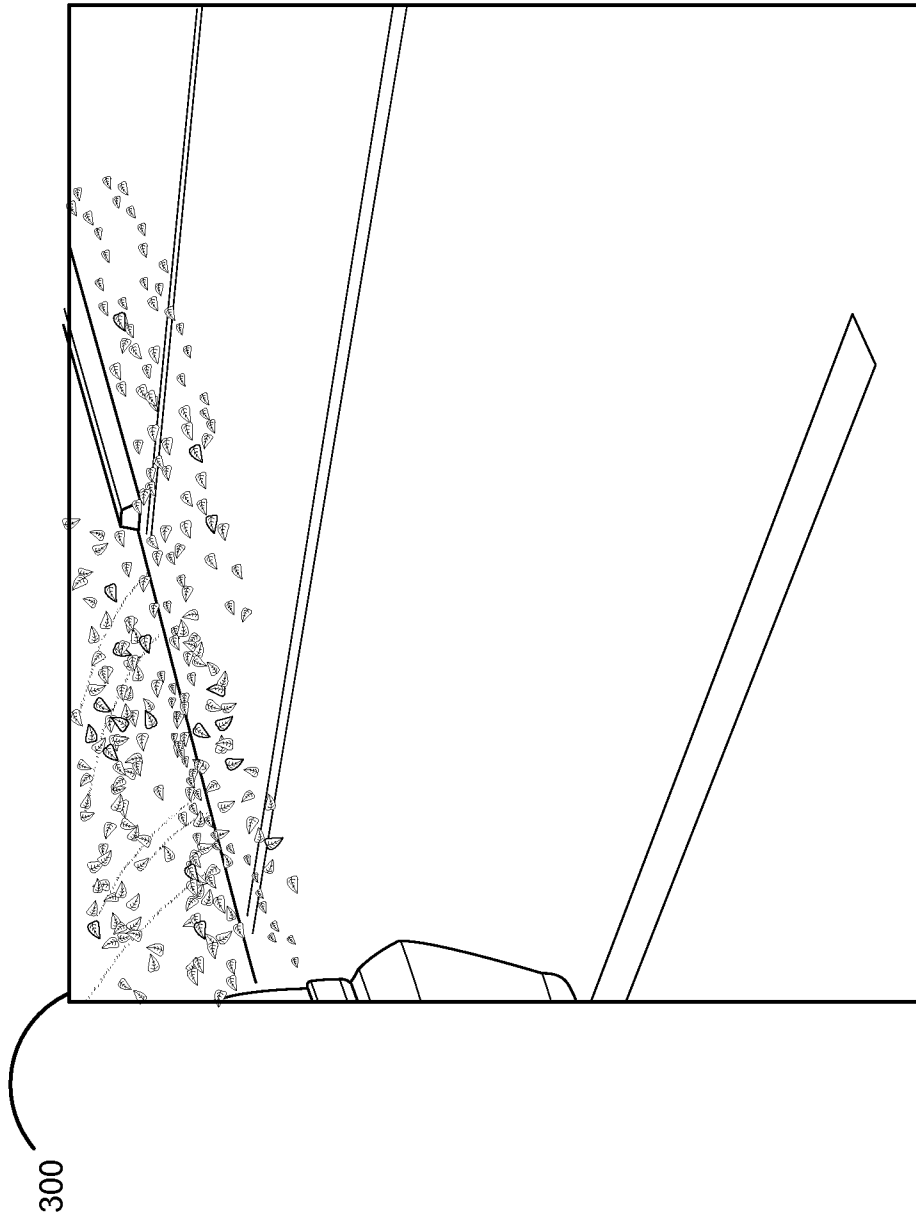


Fig. 3

4/5

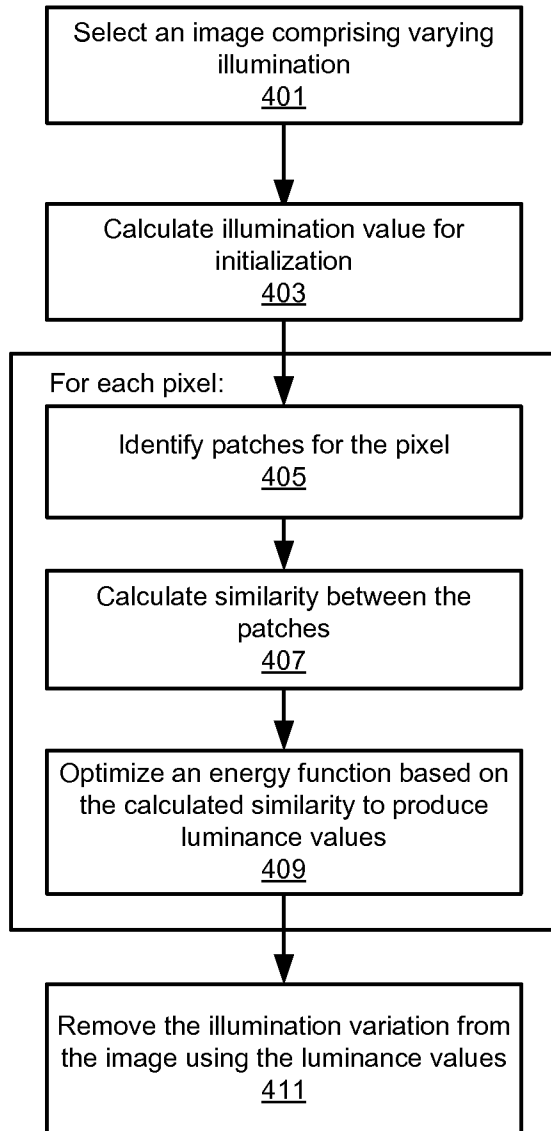
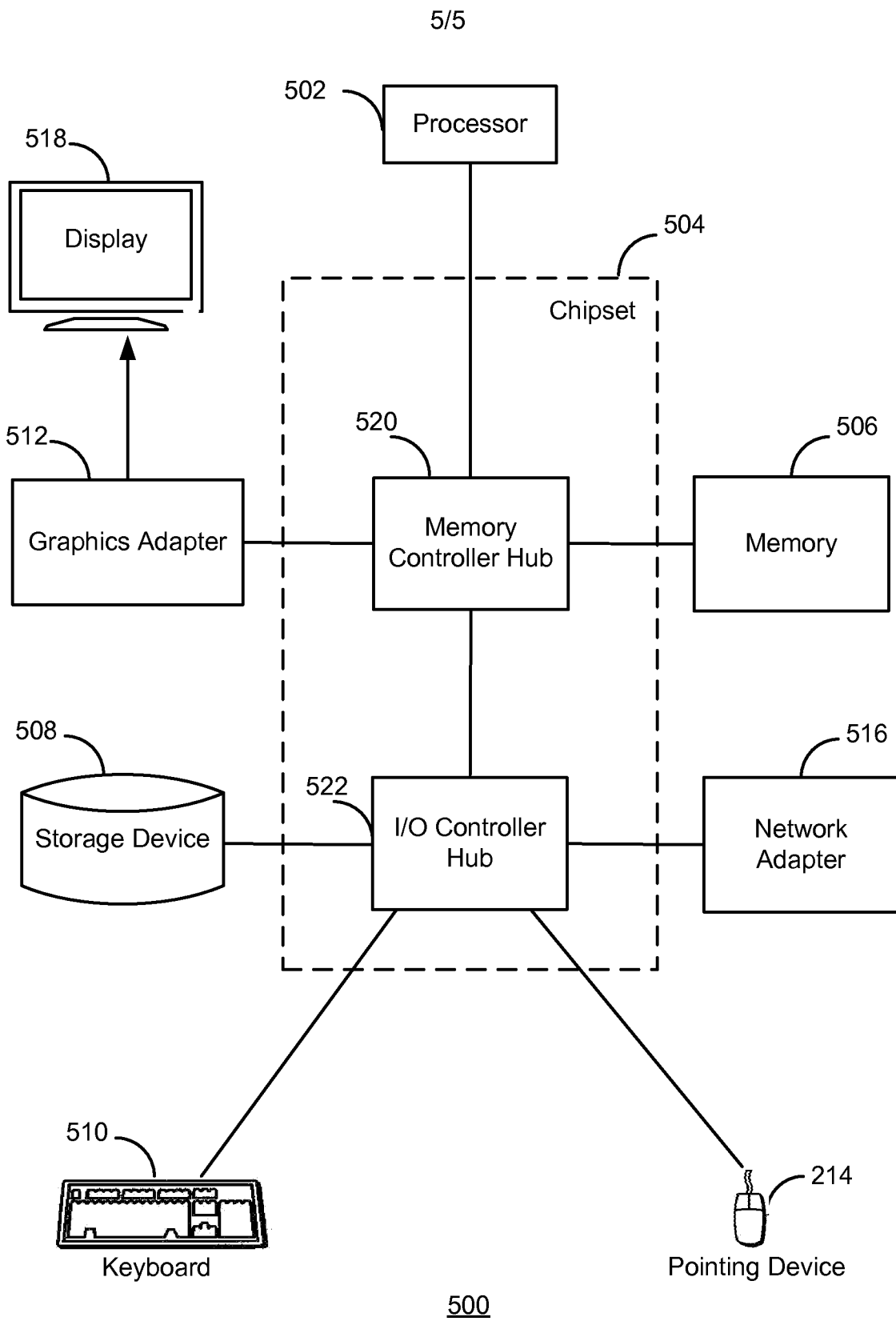


Fig. 4



**FIG. 5**

**A. CLASSIFICATION OF SUBJECT MATTER****H04N 5/21(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

H04N 5/21; G06K 9/46; G06K 9/40; G06K 9/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) &amp; Keywords: "image, illumination, variation, removing, pixel, patch, similarity"

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2005-0163397 A1 (LEO BALDWIN) 28 July 2005 See abstract, figure 4, paragraphs [0024]-[0028], and claims 1-6.	1-31
A	US 6249601 B1 (JEONG-YEOP KIM et al.) 19 June 2001 See abstract, figures 1B,7, column 7, line 17 - column 8, line 35, column 9, line 6 - column 11, line 15, and claims 1-3.	1-31
A	US 2009-0034835 A1 (BRUCE ALLEN MAXWELL et al.) 05 February 2009 See abstract, figures 3B-3C, 8-9, paragraphs [0039]-[0040], [0065]-[0076], and claims 1-4,6-7.	1-31

 Further documents are listed in the continuation of Box C. See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

27 APRIL 2012 (27.04.2012)

Date of mailing of the international search report

**04 MAY 2012 (04.05.2012)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office  
Government Complex-Daejeon, 189 Cheongsa-ro,  
Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KIM, Kwang Sik

Telephone No. 82-42-481-8355



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2011/062787**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005-0163397 A1	28.07.2005	CN 1906633 A	31.01.2007
		DE 112004002663 T5	07.12.2006
		GB 0614372 D0	30.08.2006
		GB 2437372 A	24.10.2007
		GB 2437372 B	06.08.2008
		JP 2007-524943 A	30.08.2007
		KR 10-0820722 B1	11.04.2008
		KR 10-2006-0116224 A	14.11.2006
		TW 290695 B	01.12.2007
		US 7599576 B2	06.10.2009
		WO 2005-076213 A1	18.08.2005
US 6249601 B1	19.06.2001	DE 19908211 A1	08.06.2000
		JP 04-266461 B2	27.02.2009
		JP 2000-171304 A	23.06.2000
		KR 10-0304663 B1	29.09.2001
		KR 10-2000-0038213 A	05.07.2000
US 2009-0034835 A1	05.02.2009	CN 101802842 A	11.08.2010
		EP 2171645 A1	07.04.2010
		JP 2010-535383 A	18.11.2010
		US 2009-0161950 A1	25.06.2009
		US 7760912 B2	20.07.2010
		US 8144978 B2	27.03.2012
		WO 2009-017616 A1	05.02.2009