

(19)



EP 2 333 720 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
05.03.2014 Bulletin 2014/10

(51) Int Cl.:
G06T 7/40 (2006.01) G06T 5/00 (2006.01)

(21) Application number: 10191232.7

(22) Date of filing: 15.11.2010

(54) System and method for detection of specularity in an image

System und Verfahren zur Erkennung des Reflektionsgrades in einem Bild

Système et procédé de détection de spécularité dans une image

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR

(30) Priority: 17.11.2009 US 590949

(43) Date of publication of application:
15.06.2011 Bulletin 2011/24

(73) Proprietor: Tandent Vision Science, Inc.
San Francisco, California 94111 (US)

(72) Inventors:
• Garg, Kshitiz
Pittsburgh, PA 15217 (US)
• Aldrin, Neil
Pittsburgh, PA 15217 (US)

(74) Representative: Reichert, Werner Franz
Reichert & Kollegen
Bismarckplatz 8
93047 Regensburg (DE)

(56) References cited:
WO-A1-2008/085815

- TAPPEN M F ET AL: "Recovering intrinsic images from a single image", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE SERVICE CENTER, LOS ALAMITOS, CA, US, vol. 27, no. 9, 1 September 2005 (2005-09-01), pages 1459-1472, XP001512627, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2005.185
- ROGER GROSSE ET AL: "Ground truth dataset and baseline evaluations for intrinsic image algorithms", COMPUTER VISION, 2009 IEEE 12TH INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 29 September 2009 (2009-09-29), pages 2335-2342, XP031672758, ISBN: 978-1-4244-4420-5
- YUN-CHUNG CHUNG ET AL: "Intrinsic Image Extraction from a Single Image", JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, TAIPEI, TW, vol. 25, no. 6, 1 November 2009 (2009-11-01), pages 1939-1953, XP009166127, ISSN: 1016-2364
- ANDREY DELPOZO ET AL: "Detecting Specular Surfaces on Natural Images", CVPR '07. IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION; 18-23 JUNE 2007; MINNEAPOLIS, MN, USA, IEEE, PISCATAWAY, NJ, USA, 1 June 2007 (2007-06-01), pages 1-8, XP031114472, ISBN: 978-1-4244-1179-5
- TODD ZICKLER ET AL: "Color Subspaces as Photometric Invariants", INTERNATIONAL JOURNAL OF COMPUTER VISION, KLUWER ACADEMIC PUBLISHERS, BO, vol. 79, no. 1, 30 October 2007 (2007-10-30), pages 13-30, XP019616636, ISSN: 1573-1405

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Description

[0001] Specularity in an image refers to the occurrence of a mirror-like specular reflection of light from a material surface depicted in the image. Materials such as glass, metal, human skin and most painted material surfaces are quite glossy and may exhibit prominent specularities when depicted in an image. In the execution of a computer processing technique for the analysis of the image, the occurrence of specularities can interfere with a correct and accurate performance of the executed technique. For example, in an image segregation technique, to segregate intrinsic material and illumination aspects of an image, there is an assumption that the materials depicted in the image are lambertian. However, materials exhibiting specular reflections do not satisfy the lambertian assumption, causing a segregation operation to fail. Accordingly, there is a need for the development of methods for the detection and removal of specular reflections that may appear in a recorded image.

[0002] The article "Recovering Intrinsic Images from a Single Image" by M. F. Tappen et al., IEEE Transactions on pattern analysis and machine intelligence, vol 27, p. 1459 - 1472, discusses a method for recovering shading and reflectance images from an input image. The method is based on derivatives which are computed from the input image and subsequently passed to a classifier. Both color and gray-scale information are used in the method.

[0003] The article "Ground truth dataset and baseline evaluations for intrinsic image algorithms" by R. Grosse et al., Computer Vision, 2009, IEEE 12th Intern. Conference, p. 2335 - 2342, presents a dataset of decompositions of various images into Lambertian shading, reflectance, and specularities. Several algorithms to achieve such a separation are quantitatively compared.

[0004] The article "Intrinsic Image Extraction from a Single Image" by Yun-Chung Chung et al., Journal of information science and engineering, Taiwan, vol. 25, p. 1939 - 1953, describes the separation of an input image into reflectance and illumination components. The input image is convolved with a prescribed set of derivative filters to obtain derivative images. The pixels of the derivative images are classified as being reflectance or illumination according to three measures: chromatic, intensity contrast, and edge sharpness. The reflectance and illumination images are then obtained by a deconvolution of the classified derivative image.

[0005] The international patent application PCT/US2008/000015, published as WO 2008/085815 A1, describes video-based sensing for lighting control. For better assessment of lighting levels in a scene, specularities in the video image are identified by transforming into an SUV color space.

[0006] The article "Detecting Specular Surfaces on Natural Images" by A. DelPozo et al., IEEE conference on computer vision and pattern recognition 2007, p. 1 - 8, discusses the recognition and localisation of specular surfaces from a single image. The method characterises static specular flows, which describe the distortion of a surrounding scene when reflected off regions of high curvature or occluding contours.

[0007] The present invention provides a system and method for the detection and removal of specularity in an image.

[0008] In an exemplary embodiment of the present invention, an automated, computerized method is provided for processing an image. According to a feature of the present invention, the method comprises the steps of providing an image file depicting an image, in a computer memory, detecting specularity in the image and segregating the image into intrinsic images with reference to the detected specularity.

[0009] In another embodiment of the present invention, a computer program product is provided. According to a feature of the present invention, the computer program product is disposed on a computer readable media, and the product includes computer executable process steps operable to control a computer to: provide an image file depicting an image, in a computer memory, detect specularity in the image and segregate the image into intrinsic images with reference to the detected specularity.

[0010] In accordance with yet further embodiments of the present invention, computer systems are provided, which include one or more computers configured (e.g., programmed) to perform the method described above. In accordance with other embodiments of the present invention, computer program products, including computer readable media, are provided which have stored thereon computer executable process steps operable to control a computer(s) to implement the embodiments described above. The present invention contemplates a computer readable media as any product that embodies information usable in a computer to execute the methods of the present invention, including instructions implemented as a hardware circuit, for example, as in an integrated circuit chip. The automated, computerized methods can be performed by a digital computer, analog computer, optical sensor, state machine, sequencer, integrated chip or any device or apparatus that can be designed or programed to carry out the steps of the methods of the present invention.

Figure 1 is a block diagram of a computer system arranged and configured to perform operations related to images.

Figure 2 shows an $n \times m$ pixel array image file for an image stored in the computer system of figure 1.

Figure 3 is a flow chart for performing an image segregation operation, according to a feature of the present invention.

Figure 4 is a flow chart for performing an alternative approach to an image segregation, according to a feature of the present invention.

Figure 5 is a flow chart for detecting specularity in an image, according to a feature of the present invention.

Figure 6 is a flow chart for training a specularity classifier, according to a feature of the present invention.

Figure 7 is a flow chart for removing specularity from an image, according to a feature of the present invention.

Figure 8 is a flow chart for estimating material reflectance and illumination aspects of an image depicted in an image file, according to a feature of the present invention.

Figure 9 depicts graphic illustrations of (a) a log chromaticity equation, (b) a tone mapping equation and (c) a combination of the previous two equations.

Figure 10 is a flow chart for performing monotonicity maintenance in an image.

[0011] Referring now to the drawings, and initially to figure 1, there is shown a block diagram of a computer system 10 arranged and configured to perform operations related to images. A CPU 12 is coupled to a device such as, for example, a digital camera 14 via, for example, a USB port. The digital camera 14 operates to download images stored locally on the camera 14, to the CPU 12. The CPU 12 stores the downloaded images in a memory 16 as image files 18. The image files 18 can be accessed by the CPU 12 for display on a monitor 20, or for print out on a printer 22.

[0012] Alternatively, the CPU 12 can be implemented as a microprocessor embedded in a device such as, for example, the digital camera 14 or a robot. The CPU 12 can also be equipped with a real time operating system for real time operations related to images, in connection with, for example, a robotic operation or an interactive operation with a user.

[0013] As shown in figure 2, each image file 18 comprises an $n \times m$ pixel array. Each pixel, p , is a picture element corresponding to a discrete portion of the overall image. All of the pixels together define the image represented by the image file 18. Each pixel comprises a digital value corresponding to a set of color bands, for example, red, green and blue color components (RGB) of the picture element. The present invention is applicable to any multi-band image, where each band corresponds to a piece of the electro-magnetic spectrum. The pixel array includes n rows of m columns each, starting with the pixel $p(1,1)$ and ending with the pixel $p(n,m)$. When displaying or printing an image, the CPU 12 retrieves the corresponding image file 18 from the memory 16, and operates the monitor 20 or printer 22, as the case may be, as a function of the digital values of the pixels in the image file 18, as is generally known.

[0014] According to a feature of the present invention, in an image process, the CPU 12 operates to analyze and process information, for example, the RGB values of the pixels of an image stored in an image file 18, to achieve various objectives, such as, for example, a correct and accurate identification of illumination and material aspects of the image. The present invention provides a method and system for generation of intrinsic images. The intrinsic images correspond to an original image, for example, an image depicted in an input image file 18. The intrinsic images include, for example, an illumination image, to capture the intensity and color of light incident upon each point on the surfaces depicted in the image, and a material reflectance image, to capture reflectance properties of surfaces depicted in the image (the percentage of each wavelength of light a surface reflects).

[0015] A fundamental observation underlying a basic discovery of the present invention, is that an image comprises two components, material and illumination. All changes in an image are caused by one or the other of these components. Spatio-spectral information is information relevant to contiguous pixels of an image depicted in an image file 18, such as spectral relationships among contiguous pixels, in terms of color bands, for example RGB values of the pixels, and the spatial extent of the pixel spectral characteristics relevant to a characteristic of the image, such as, for example, a single material depicted in the image or illumination flux effecting the image. When one of material and illumination is known in an image, the other can be readily deduced.

[0016] Referring now to figure 3 there is shown a flow chart for performing an image segregation operation, according to a feature of the present invention. An input image file 18 is input to the CPU 12, operating as an image processor (step 50). The CPU 12 performs each of a specularity detection process 52 and a specularity removal process 54, to output a specular-free image 56. The specular-free image 56 is then input to the CPU 12, now operating as a segregation engine 58, to generate intrinsic images corresponding to the input image, for example, each of an illumination map 60 and a material map 62. The illumination map 60 comprises an intrinsic illumination image corresponding to the original input image file 18, to capture the intensity and color of light incident upon each point on the surfaces depicted in the image. The material map 62 comprises an intrinsic material reflectance image corresponding to the original input image file 18, to capture reflectance properties of surfaces depicted in the image (the percentage of each wavelength of light a surface reflects). The input of a specular-free image assures a more accurate and correct performance of the image

segregation.

[0017] Figure 4 shows a flow chart for performing an alternative approach to an image segregation, according to a feature of the present invention. In this exemplary embodiment of the present invention, the input image file 50 is input to the CPU 12, operating both as an image processor (for specularity detection 52) and as a segregation engine 58.

The specularity detection process 52 outputs a specular mask 64, which identifies all pixels of the image file 18 exhibiting specular reflection. The specular mask 64 is also input to the segregation engine 58. The segregation engine 58 generates the illumination map 60 and the material map 62 with reference to the specular mask 64. The pixel information on specularities contained in the specular mask 64 is either treated as a noise factor in the segregation processing or is handled as a component of the illumination aspects of the image in the generation of the illumination map. In the alternative embodiment of figure 4, only detection of specularities is performed, reducing the computational overhead of the overall process.

[0018] Figure 5 shows a flow chart for detecting specularity in an image (step 52 in figures 3 and 4), according to a feature of the present invention. The input image 50 is processed by the CPU 12 to identify regions in the image, for example, to identify tokens in the image (step 64). A token can comprise, for example, a uniform, homogeneous Type C token, which comprises an image region having a single robust color measurement among contiguous pixels of the region. For example, all of the pixels of a Type C token have colors within the noise level of the device used to record the image, such as the digital camera 14.

[0019] A seed test can be utilized by the CPU 12 to identify Type C tokens in an image file 18. The CPU 12 selects a first pixel, $l = (1, 1)$ for example, the pixel at the upper left corner of a first $N \times N$ sample of the image file 18 (see figure 2). The pixel is then tested to determine if the selected pixel is part of a good seed. The test can comprise a comparison of the color value of the selected pixel to the color values of a preselected number of its neighboring pixels as the seed, for example, the $N \times N$ array. The color value comparison can be with respect to multiple color band values (RGB in our example) of the pixel, relative to the noise level of the digital camera 14.

[0020] When a good seed is found, the CPU 12 proceeds to examine pixels contiguous to the seed, to build out a token. The CPU 12 marks each identified token with a unique label, and lists for the token each pixel forming the token. For a more detailed description of token identification, reference is made to U. S. Patent Application Publication No. 2009/0034835, published on February 5, 2009.

[0021] Tokens are used to improve the computational efficiency of the process by analyzing groups of pixels having a single robust color measurement, that is analyzing a representation of all of the constituent pixels of a token at one time. The CPU 12 analyzes each token identified in step 64, one at a time, in respect of properties or characteristics of specularity. The characteristics include:

- specularities are typically much brighter than surrounding regions of the image;
- specularities possess the color of the illuminating source;
- specularities have characteristic shapes, for example, circular spots for spherical objects and linear spots for cylindrically shaped objects
- brightness at a region of specular pixels, falls abruptly, from illumination color to material color
- specular regions show a T shaped color distribution when analyzed in a color space, at a local region.

[0022] Each token is represented by the average color value for the recorded color values of the constituent pixels of the token. The CPU 12 also provides the specular mask 64, corresponding to the input image file 18, wherein an indication location is provided for each identified token. Each identification location will be set at an indication of 1 or 0, 1 representing an indication that the respective token is specular, and 0 representing an indication that the token is not specular.

[0023] Initially, in step 66, the CPU 12 determines if a selected token is saturated. Saturation refers to color values for bands of a color space that are close to a maximum value possible for the color band. For example, in RGB space, with an 8 bit indication of color value, if any of the Red, Green or Blue bands has a value close to 255, the corresponding token is considered to be saturated. In step 66, if the token under consideration is saturated, then the indication location for the token in the specular mask 64 is set at 1 (68). If the token under consideration is not saturated, the CPU 12 proceeds to step 70.

[0024] In step 70, the CPU 12 converts the color representing the token under consideration into an SUV color space value. SUV space is a source dependent color space wherein the axes of an RGB coordinate system are rotated such that one of the axes (for example, the red axis) is aligned with RGB source vector s . The SUV space separates specular and diffuse reflection effects from one another. In this manner, the S channel of the SUV space encodes the entire specular component and an unknown portion of the diffuse component. The UV channels are specular invariant. For a more detailed description of an SUV space, reference is made to "Color Subspaces as Photometric Invariants," Todd Zickler et al., International Journal of Computer Vision, 79(1): 13-30, August 2008 (hereinafter the Zickler et al. Reference).

[0025] As described in the Zickler et al. reference, the color of the source illumination is determined to implement the conversion into SUV space. In general, light source colors tend to be neutral or close to white. If an image is white

balanced, the illumination source color can be assumed to be white. Otherwise, according to a feature of the present invention, in step 72, a bi-illuminant dichromatic reflection model (BIDR model), can be used to calculate the illumination source color. The BIDR model predicts that when a color change is due to an illumination change (i.e. an increase in the intensity of the incident illuminant in the interplay between the incident or direct illuminant and the ambient illuminant, in the illumination field) over a single material, from full shadow to fully lit, a line is defined in RGB space, depicting the color variations for a pixel, from full shadow to fully lit, as a function of the percentage of incident illuminant in the illumination field.

[0026] For purposes of describing, identifying and using the model, the BIDR model can be stated as:

$$I(x, y, z, \theta, \phi, \lambda) = c_b(\lambda) I_d(\lambda) \gamma_b + M_a(\lambda) c_b(\lambda),$$

where:

$I(x, y, z, \theta, \phi, \lambda)$ is the radiance of a surface point at (x, y, z) in the direction θ, ϕ for the wavelength λ , $c_b(\lambda)$ is the geometry independent body reflectance of a surface for the wavelength λ , $I_d(\lambda)$ is the incident illuminant for the wavelength λ , γ_b is the product of a shadow factor S_x, y, z and a geometric factor $mb(\theta)$, and $M_a(\lambda)$ is the integral of the ambient illuminant and geometric body reflectance over a hemisphere, excluding the incident illuminant. For more detailed information on the BIDR model, reference should be made to US Patent Application Publication No. 2007/0176940, published on August 2, 2007, and "A Bi-illuminant Dichromatic Reflection Model for Understanding Images," CVPR08 (1-8) (2008).

[0027] In step 72, in the event the image is not white balanced, the CPU 12 generates a plot according to the BIDR model, in log RGB space, to determine the illumination color value for use in the SUV conversion of step 70, as described in sections 3 and 4 of "A Bi-illuminant Dichromatic Reflection Model for Understanding Images," CVPR08 (1-8) (2008).

[0028] In step 74, the SUV values for the token under consideration, as determined in step 70, are evaluated by the CPU 12, to determine whether the token is a candidate for classification as a specular token. To that end, the ratio of $s/(\sqrt{u^2 + v^2})$ is calculated. If the ratio has a value such that $(s >> \sqrt{u^2 + v^2})$ there is a high likelihood that the token under consideration is specular. Moreover, the value of $\sqrt{u^2 + v^2}$ is compared to a threshold value, for example, the value of $\sqrt{u^2 + v^2}$ is not lower than, for example, 0.01 to 0.005. A value lower than the threshold is an indication that the material color of the token under consideration may be similar to the illumination color value, and is most likely not a specular token.

[0029] Thus, in step 74 the CPU 12 evaluates the conditions $(s >> \sqrt{u^2 + v^2})$ and $\sqrt{u^2 + v^2}$ is not lower than, for example, 0.01 to 0.005. If both conditions are met, the CPU 12 designates the token under consideration as a candidate for being classified as a specular token. If either of the conditions are not met, the CPU 12 designates the token under consideration as not being a candidate.

[0030] In decision block 76, if a token under consideration is not a candidate, the CPU 12 sets the indication in the specular mask for the token under consideration to 0 (step 78). If the token is a candidate, the CPU 12 proceeds to step 80.

[0031] In step 80, the CPU 12 begins a more extensive analysis of the token under consideration to substantiate the status of the token as a specular token. In the further analysis, the CPU 12 evaluates cues relevant to specularly. The cues are based upon the various properties or characteristics exhibited by specular pixels, as described in detail above. The cues can be represented, for example, by features computed by the CPU 12 by application of Gabor filters to pixels of a token under consideration, in each of several different color spaces, for example, in each of RGB, SUV, HSV, color opponent space and log RGB value color spaces. The Gabor filters can be in each of several different orientations and wavelengths, such as, for example, orientations of 0, 45, 90 and 135 degrees and wavelengths of 6 pixels and 4 pixels, respectively. The Gabor features, relevant to each of the different color spaces, for a token under consideration, are concatenated in a feature vector that is input to a known computer learning operation such as, for example, a Support Vector Machine (SVM) technique (step 82).

[0032] In the SVM technique of step 82, an SVM classifier is applied to an input feature vector representing a candidate token, for classification as a specular token. The classifier is a process or algorithm that takes an input feature vector extracted from the candidate token and outputs a prediction or classification regarding a likelihood that the candidate token is a specular token. The classification can be expressed as a score. The score can be in the range of from 0 to 1, a 0 corresponding to no likelihood that the candidate token is specular and a 1 corresponding to a high likelihood that the candidate token is specular.

[0033] Referring now to figure 6, there is shown a flow chart for training a specularly classifier, such as an SVM classifier, according to a feature of the present invention. A computer learning technique, such as the SVM approach, typically includes the step of providing a training set, with the training set comprising a series of sample images selected from an image database. For each of the images of the training set, local image features are identified to provide representations of, for example, features of interest and relevant, in our example, to specularly characteristics and

properties. Parameters based upon the features, from each of the images of the image database, are arranged in feature vectors for use in training.

[0034] In a learning technique, a classifier is trained from the training set, for use in the learning technique, to, for example, classify regions of an input image as having specularity. As noted above, a classifier in our exemplary embodiment of the present invention, is trained to be a process or algorithm that takes a feature vector extracted from an input candidate token and outputs a prediction or classification score regarding the likelihood that the candidate token is a specular token.

[0035] To that end, in step 101 of figure 6, an image database is provided. The image database includes sample images with specularity, for positive examples, and sample images without specularity, as negative examples, as is generally known in the computer learning art. In step 103, each image of the image database is processed by the CPU 12 in the same manner as the candidate token, as described above, that is, the CPU 12 computes Gabor features for each image in each of RGB, SUV, HSV, color opponent space and log RGB value color spaces. The Gabor features for each image are concatenated to provide a feature vector corresponding to each image of the image database.

[0036] For conversion of an image to an SUV color space, the CPU 12 utilizes BIDR model information, as described above, to determine the illumination color value for a respective image of the image database (step 105). If the images of the image database are white balanced, the illumination color is assumed to be white. The resulting feature vectors derived from the image database are input as input training to an SVM training engine 107.

[0037] According to a further feature of the present invention, each image (or a subset of the images) of the image database (labeled data set 109) is manually coded to indicate regions of specularity in the image depicted in the respective image of the database (for explicit positive examples) (data subset 111) and samples of regions with non-specular properties (for explicit negative examples) (data subset 113). In an exemplary embodiment of the present invention, the coding is a color coding wherein a human operator views each image of the labeled data set 109, and in a touch screen, brush stroke operation, marks off each region of specularity (for the positive examples) or of non-specular properties (for negative examples), and controls the computer to assign an arbitrary color to each marked region. All of the labeled images are input to the SVM training engine 107.

[0038] Upon input of all of the training images, the SVM classifier is trained as per the known Support Vector Machine technique. The output of the SVM Training engine is an SVM classifier (step 115).

[0039] Referring once again to figure 5, in step 82, the CPU 12 applies the SVM classifier (output 115 of figure 6), to the feature vector for an input candidate token. The SVM classifier generates a score for the candidate token, of from 0 to 1, as described above. The CPU 12 can evaluate the score in relation to a threshold value, for example, 0.8. If the score for the candidate token is 0.8 or more, the indication for the candidate token in the specular mask is set at 1 (step 84). If the score for the candidate token is less than 0.8, the indication for the candidate token in the specular mask is set at 0 (step 86).

[0040] According to a feature of the present invention, steps 66 to 86 of figure 5 are performed for each token identified in step 64 of the process of figure 5, to generate a complete specular mask for the given image 50.

[0041] Figure 7 is a flow chart for removing specularity from an image, according to a feature of the present invention (step 54 in figure 3). In the method for removing specularities, the input image 50 is processed by the CPU 12 in a calculation of image gradients (step 117). Gradients are often utilized to provide a more suitable representation of an image for purposes of computer processing. A gradient is a measure of the magnitude and direction of color and/or color intensity change within an image, as for example, across edges caused by features of objects depicted in the image. Any known method can be implemented to calculate the image gradients, as for example, convolution of pixels using a Sobel filter.

[0042] In step 119, the CPU 12 computes a specular mask for the input image as per the routine of figure 5, as described above. In the decision block of step 121, the CPU 12 examines the specular mask to determine if any of the tokens of the input image have been designated by a 1, indicating a specular token. If no, the CPU 12 can exit the specular removal routine (step 123).

[0043] If yes, the CPU 12 proceeds to step 125. In step 125, the CPU 12 blurs those regions of the image corresponding to specular tokens, as indicated in the specular mask. The blurring can be a Gaussian blur.

[0044] In step 127, the CPU 12 modifies the gradients of all pixels within specular tokens, as specified in the specular mask, to remove specular characteristics. For example, each specular gradient is set to zero. In step 129, the CPU 12 reintegrates the gradient representation of the image, with the modified versions of the gradients for pixels in the specular tokens, to regenerate the original image, however, now without specularities, due to the modified specular gradients. The reintegration can be performed utilizing any well known image reintegration technique, such as, for example, a Poisson solver with Neumann boundary conditions.

[0045] Due to the masking and modification of specular gradients, the modified specular regions of the original image will appear flat and without any texture in the reintegrated image. Accordingly, in step 131, the CPU 12 operates to restore texture to those regions of the image that have been modified to eliminate specularities. The restoration is achieved utilizing information from the UV channels of the SUV color space representations of the image. To that end,

the ratio of the UV magnitude of the original image to the UV magnitude in the modified, reintegrated image, is used to modulate the intensity of the former specular regions of the reintegrated image, to thereby restore a textured appearance.

[0046] Upon completion of step 131, the CPU 12 outputs a specular free image 56.

[0047] Referring once again to figures 3 and 4, either the specular free image 56 (the example of figure 3) or the given image 50 (together with the specular mask 64) (the example of figure 4) is input to the segregation engine 58, as the case may be. Figure 8 is a flow chart for estimating material reflectance and illumination aspects of an image depicted in an image file 18 (either as processed as the specular free image 56, or comprising the given image 50, with reference to the specular mask 64), to implement the segregation engine 58 of figures 3 and 4, for generation of the illumination map 60 and the material map 62, according to a feature of the present invention.

[0048] Referring now to figure 8, in step 100, the input image file 18 is given to the CPU 12. Upon input of an image file 18, the CPU 12 executes, independently, either serially, or in parallel, operations for performing each of a tone mapping (steps 102 and 104) and a calculation of log chromaticity values (step 106) for the image depicted in the input image file 18, as will appear. In the event the input is the given image 50, and the specular mask 64, the CPU 12 treats all pixels in tokens designated as specular in the specular mask 64, as noise, for example, each such specular pixel is set to zero or ignored in processing, or assigns the pixel color values for the specular tokens to the illumination map.

[0049] Tone mapping is a technique for preserving local variation within an image, while reducing global variation. Specific known methods for performing tone mapping on an image are described in "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images," Frédo Durand and Julie Dorsey, ACM Transactions of Graphics (Proceedings of the ACM SIGGRAPH '02 Conference), and "A Fast Approximation of the Bilateral Filter using a Signal Processing Approach," Sylvain Paris and Frédo Durand, European Conference on Computer Vision (ECCV '06).

[0050] Performance of the technique separates an image into a "level" channel and a "detail" channel. The level channel depicts large scale variations of the image, without details, which are depicted in the detail channel. As such, the level channel is a reasonable approximation of log illumination intensity of the image, and the detail channel is a reasonable approximation of the log material intensity. The output of a standard tone mapping technique is an intensity specification for each pixel of the image, such as, for example, a pixel $p(n, m, R, G, B)$ is specified by the value $(R + G + B)/3$.

[0051] According to a feature of the present invention, a standard tone mapping technique is modified in a manner to accommodate the objective of segregating illumination and material aspects of an image from one another. In a standard tone mapping operation, the dynamic range of the level channel is reduced and then recombined with the detailed channel to provide a reduction in global contrast while preserving local contrast. In an exemplary implementation of the present invention, the tone mapping is used in a manner that attempts to remove global variation rather than just reduce global variation, to segregate out an illumination invariant material aspect of the image.

[0052] A formula for implementing a reconstruction of an intensity image with a reduction of the dynamic range of the level channel, can be expressed in terms of a simple linear transform. For example, let \max = the maximum value in the level channel, \min = the minimum value in the level channel, and δ represents a desired contrast ($\max - \min$). Then a dynamic range reduction can be performed by multiplying each value in the level channel by $\log(\delta)/(\max - \min)$. A typical value for δ , in a standard application of tone mapping, is 40, which corresponds to the dynamic range of print media, wherein black ink is 40 times darker than white paper.

[0053] Thus, a standard reconstruction for an intensity image I , can be expressed by:

$$I = \exp(l * \log(40)/(\max - \min) + 1 * d)$$

wherein l is the level channel and d is the detail channel.

[0054] According to a feature of the present invention, the desired contrast (in an attempt to eliminate the effects of illumination) is set at 1 or lower, for example, 0.85. Thus, pursuant to an exemplary embodiment of the present invention, a reconstruction to provide a material intensity image (M) is expressed by:

$$M = \exp(l * \log(0.85)/(\max - \min) + 1.2 * d)$$

[0055] As indicated in the above formula for M , the local contrast of the detail channel d is also modified by multiplying the detail channel by 1.2. Thus, according to a feature of the present invention, a unique material intensity based value can be determined for each pixel of an image by modifying a tone mapping technique toward an attempt to eliminate the effects of illumination, as described above.

[0056] Log chromaticity is a technique for developing a two dimensional illumination invariant chromaticity space. A method and system for separating illumination and reflectance using a log chromaticity space is disclosed in U. S. Patent

Application Publication No. 2007/0242878, published on October 18, 2007. Log chromaticity representation values provide illumination invariant values for pixels of the image. Logarithmic values of the color band values of the image pixels are plotted on a log-color space graph. The logarithmic values are then projected to a log-chromaticity projection plane oriented as a function of the BIDR model. As noted above, the BIDR Model predicts that differing color measurement values fall generally along a line in RGB space, from a dark end (in shadow) to a bright end (lit end), along a positive slope, when the color change is due to an illumination change forming a shadow over a single material of a scene depicted in the image.

[0057] Thus, RGB values of a material color, from full light to full shadow, vary generally along a gentle one dimensional curve in RGB space. For more detailed information on the BIDR model, reference should be made to U. S. Patent Application No. US 2007/0176940, published on August 2, 2007.

[0058] A measured color C of a pixel, according to the BIDR model, can be expressed as:

$$C_{\lambda} = (A_{\lambda} + \gamma D_{\lambda}) M_{\lambda},$$

where A is the ambient illumination, D is the direct or incident illumination, γ is a scalar from 0 to 1, representing the fraction of the direct illumination, D, falling upon the point in the image represented by the respective pixel, M is the material reflectance of the surface and λ is the wavelength or color band, such as red, green or blue.

[0059] In log space, the equation is stated as: $\log C_{\lambda} = \log (A_{\lambda} + \gamma D_{\lambda}) + \log M_{\lambda}$.

[0060] If a measurement is taken of color value at two locations on the same material ($C_{1,\lambda}$, $C_{2,\lambda}$), one in full light (for $C_{1,\lambda,\gamma} = 1$), and the other in full shadow (for $C_{2,\lambda,\gamma} = 0$), the difference in color values is stated by: $\log C_{1,\lambda} - \log C_{2,\lambda} = \log (A_{\lambda} + D_{\lambda}) - \log A_{\lambda}$.

[0061] It should be noted that the difference equation is independent of material reflectance. Accordingly, in a log RGB color space, the orientation of the positive slope is the same for the BIDR cylinders for all materials depicted in the image. Thus, projections of the BIDR cylinders onto a plane perpendicular to the common orientation can be utilized to provide an unique, illumination invariant value (based upon chromaticity) for each pixel of an image.

[0062] According to a feature of the present invention, a unique, illumination invariant material color for each pixel can be determined as a function of a combination of tone mapping and log chromaticity information relevant to the pixel. As shown in figure 9(a), a plot of the log chromaticity equation defines, for a particular material, a gently curving, one dimensional line in, for example, an RGB space, corresponding to the varying colors of the material, from full light, to full shadow. Tone mapping for a pixel specifies a particular intensity for the pixel, but not a specific color. As shown in figure 9(b), this property of tone mapping defines a two dimensional plane in RGB space. There is exactly one color value that satisfies both the log chromaticity and tone mapping intensity equations (see figure 9(c)).

[0063] Log chromaticity can be expressed by the following equation:

$$\log M_{\lambda} = \log C_{\lambda} + \gamma \log N_{\lambda},$$

where M is the material color, C is the measured image color, as stored in an image file 18, N is the normal to the log chromaticity plane (corresponding to the common orientation of the BIDR cylinders in log RGB space), and γ is the unknown factor parameterizing the log chromaticity line in log RGB space.

[0064] In linear space, the log chromaticity equation becomes:

$$M_{\lambda} = \exp(\log C_{\lambda} + \gamma \log N_{\lambda}).$$

In RGB space, the equation defines the one dimensional curve depicted in figure 4(a), that specifies all the varying colors for the material represented by a respective pixel, from fully lit to full shadow.

[0065] As noted above, tone mapping specifies a particular intensity for each pixel. The intensity can be measured in many ways. For example:

$$I = (R + G + B)/3.$$

[0066] A formula such as the above stated intensity equation is a single-valued function that specifies a two dimensional manifold, such as the plane illustrated in figure 9(b). To the extent that illumination variation is large relative to material variation (i.e. shadows are sufficiently deep such that shadowed bright materials are significantly darker than lit dark materials) tone mapping provides a reasonable estimate of the intensity of material reflectance at each pixel in an image. Thus, the two dimensional manifold represents a range of colors that includes a full-lit version of the material represented by a pixel.

[0067] According to a feature of the present invention, the one color value in RGB space that is both on the log chromaticity line and in the tone mapping intensity plane, for a pixel (see figure 9(c)) is taken as the material color for that pixel. The one color value that satisfies both the log chromaticity and tone mapping equations can be determined by constraining a pixel color value for a pixel P_0 , to the requirements of both equations.

[0068] In the simplest case, where $I = (R + G + B)/3$, the following equations specify an intensity for a color value for P_0 that satisfies both of the log chromaticity and tone mapping equations:

$$M_{\text{red}} = \exp(\log(C_{\text{red}}) + \gamma \log(N_{\text{red}}))$$

$$M_{\text{green}} = \exp(\log(C_{\text{green}}) + \gamma \log(N_{\text{green}}))$$

$$M_{\text{blue}} = \exp(\log(C_{\text{blue}}) + \gamma \log(N_{\text{blue}}))$$

where Intensity (M) = V (the actual value returned by execution of the tone mapping), the C values are the known measured values for the pixel P_0 , as stored in an image file 18, N is the normal to the log chromaticity plane, as determined via execution of the methods taught in the above cited U. S. Patent Application Publication No. 2007/0242878, and γ is to be found, set at a value such that the following condition is met:

$$((\exp(\log(C_{\text{red}}) + \gamma \log(N_{\text{red}}))) + (\exp(\log(C_{\text{green}}) + \gamma \log(N_{\text{green}}))) + (\exp(\log(C_{\text{blue}}) + \gamma \log(N_{\text{blue}}))))/3 - V = 0$$

[0069] An approximate solution for a value of γ that satisfies the above condition can be determined using a standard root-finding method, such as Newton's method. When an approximate solution for γ has been determined, the determined value is plugged into the above equations to solve for M_{red} , M_{green} , and M_{blue} to thereby provide a material reflectance value for the pixel P_0 .

[0070] Referring once again to figure 8, in step 102, the CPU 12 operates to calculate an intensity value for each pixel of the input image file 18. In step 104, the CPU 12 executes a standard tone mapping operation, modified as described above, toward an attempt to eliminate the effects of illumination. Thus, through execution of steps 102 and 104, the CPU 12 determines a V value for each pixel of the input image file 18.

[0071] In a separate operation, performed serially or in parallel to steps 102 and 104, the CPU 12 executes step 106, to calculate a log chromaticity image. In this manner, the CPU 12 can determine a value for N , for the input image file 18. The CPU 12 can also determine a log chromaticity value for each pixel of the input image file 18, that corresponds to a location on a log chromaticity plane, as taught in U. S. Patent Application Publication No. 2007/0242878.

[0072] Optionally, in step 108, the CPU 12 operates to quantize the tone mapping intensity and/or log chromaticity values for improved operating results. Initially, the CPU 12 clusters the log chromaticity values of the pixels, as determined by the CPU 12 in step 106. This can be performed by any known, standard clustering method, such as lead-follower clustering. Then within each chromaticity cluster, the CPU 12 clusters the tone mapping intensity values. For each pixel within a chromaticity cluster, the CPU 12 checks to determine which tone mapping intensity cluster the pixel is in, and then sets the intensity value for the pixel to the mean intensity value for the respective cluster. Quantizing the intensity in this manner reduces the influence of slow intensity change in an image, often due to shading, which can remain in the tone-mapped output.

[0073] Log chromaticity values can be subject to a quantizing by setting the log chromaticity value for each pixel to the mean of a respective cluster.

[0074] In step 110, for each pixel in the input image file 18, the CPU 12 utilizes the determined values of V and N to solve for γ , and then, with the solved value for γ , solves the M_{red} , M_{green} , and M_{blue} equations with the determined values

of γ , V and N , to provide an estimated material reflectance value for each pixel. This provides an intrinsic material image corresponding to the image depicted in the input image file 18. For an illumination image, the CPU 12 can execute the formula $I = ML$, where I is the image color, as recorded and stored in the respective image file 18, M the material component of the recorded image color, as solved for with the M_{red} , M_{green} , and M_{blue} equations and L the illumination component of the recorded image color, to be solved for in the equation.

[0075] In step 112, the CPU 12 optionally performs post-processing, such as, for example, monotonicity maintenance and the use of a bilateral blurring filter on the illumination image. Tone mapping can be prone to certain types of errors, particularly "halo effects." Halo effects occur next to edges of high contrast. Close to the edge, the bright region becomes even brighter, and the dark region becomes even darker. Monotonicity maintenance can aid in the removal of the halo effects.

[0076] A monotonic linear transition in an image relates to blend pixels. Blend pixels are pixels between two differently colored regions of an image. If the colors between the two regions are plotted in RGB space, there is a linear transition between the colors, with each blend pixel, moving from one region to the next, being a weighted average of the colors of the two regions.

[0077] As one moves in color space, from a first region to a second region, the weighted average of the colors of the blend pixels transition linearly, from a weighted average weighted in favor of the color of the first region to a weighted average weighted in favor of the color of the second region. The two regions can be regions of differently colored materials, or a region of a single material with a transition from lit to shadowed regions of the single material. In the performance of the method of the present invention, monotonic linear transitions can be distorted in the generated intrinsic images, such as the halo effect.

[0078] Referring now to figure 10, there is shown a flow chart for performing monotonicity maintenance in an image. In step 200, the CPU 12 is provided with the original input image file 18, and intrinsic material and illumination images corresponding to the image file 18, for example, as generated through performance of the routine of figure 8. The CPU 12 identifies tokens in the input image file 18, as described above.

[0079] In step 202, the CPU 12 identifies pairs of tokens, each pair exhibiting certain properties, for example:

1. The two tokens of the pair are within a preselected distance d , (for example, $d = 15$ pixels), of each other within the image space.
2. The two tokens have a minimum seed size s_1 , (for example, $s_1 = 2$).
3. The two tokens have at least p_1 pixels each, (for example, $p_1 = 20$).
4. The two tokens have a color relationship such that one is brighter than the other in all color bands.
5. The two tokens have a color relationship such that the two colors are sufficiently different from one another.

[0080] In step 204, the CPU 12 identifies, for each token pair, all tokens that are both spatially between the tokens of the pair, and satisfy the following criteria:

1. The "between" token has a color close to the line in RGB space that extends between the colors of the tokens of the respective pair (to satisfy a monotonic linear transition condition).
2. The "between" token has a seed size of at most s_2 , (for example, $s_2 = 2$).
3. The "between" token has at most p_2 pixels, (for example, $p_2 = 20$).

[0081] In step 206, the CPU 12 records, for each identified "between" token, the pair of tokens it is located between and the position of its color along the color line in RGB space between the colors of the respective token pair.

[0082] In step 208, the CPU 12 modifies the material image based upon the information recorded for "between" tokens. According to a feature of the present invention, the CPU 12 identifies tokens in the material image that correspond, pixel by pixel, to the tokens of the input image file 18. The CPU 12 then proceeds to recolor each corresponding "between" token of the material image, to be the color at the same fractional distance along the color line in RGB space between the colors of the respective corresponding token pair.

[0083] In step 210, the CPU 12 recalculates the illumination image, as a function of the re-colored material image, by execution of the $I = ML$ equation.

[0084] Moreover, detail of the image may appear in the illumination image. Execution of the $I = ML$ equation can result in image detail being forced into the illumination image. To force the detail back into the material image, the illumination image is blurred by passing the image through a bilateral blurring filter, and then the material image is recalculated.

[0085] Referring once again to figure 8, in step 114, the CPU 12 outputs a material reflectance image (material map 62 of figures 3 and 4) and an illumination image (illumination map 60 of figures 3 and 4).

Claims

1. An automated, computerized method for processing an image, comprising the steps of:

- providing an image file (18) depicting an image, in a computer memory (16);
- detecting specularity in the image by
 - tokenizing the image, and for each token
 - marking the token as specular if the token is saturated, otherwise proceeding by:
 - transforming the color of the token into an SUV color space, an SUV color space being a color space where the color of the source illumination is determined to implement the conversion into SUV space.
 - considering the token as a candidate for a specular token if both the component of the token's color in the S-direction is large compared with the component in the UV-plane and if the component in the UV-plane is above a predefined threshold, otherwise marking the token as non-specular;
 - proceeding in case of a token which is a candidate for a specular token to extracting a feature vector from the token, wherein the features in the feature vector are indicative of specularity;
 - inputting the feature vector to a classifier;
 - marking the token as a specular token if a score output from the classifier is above a predefined threshold, otherwise marking the token as non-specular.
- and
- segregating the image into intrinsic images with reference to the detected specularity, wherein the intrinsic images include an illumination map and a material map.

2. The method of claim 1, wherein the feature vector is calculated from the token by application of Gabor filters to pixels of the token.

3. The method of claim 1 or 2, wherein the step of segregating the image into intrinsic images with reference to the detected specularity is carried out by removing the detected specularity from the image to provide a specular free image.

4. The method of claim 1 or 2, wherein the step of segregating the image into intrinsic images with reference to the detected specularity is carried out by generating a specular mask and segregating the image as a function of the specular mask.

5. A computer program product, disposed on a computer readable media, the product including computer executable process steps operable to control a computer (10) to:

provide an image file depicting an image, in a computer memory (16), detect specularity in the image by

- tokenizing the image, and for each token
- marking the token as specular if the token is saturated, otherwise proceeding by:
 - transforming the color of the token into an SUV color space, an SUV color space being a color space where
 - considering the token as a candidate for a specular token if both the component of the token's color in the S-direction is large compared with the component in the UV-plane and if the component in the UV-plane is above a predefined threshold, otherwise marking the token as non-specular;
 - proceeding in case of a token which is a candidate for a specular token to extracting a feature vector from the token, wherein the features in the feature vector are indicative of specularity;
 - inputting the feature vector to a classifier
 - marking the token as a specular token if a score output from the classifier is above a predefined threshold, otherwise marking the token as non-specular

and segregate the image into intrinsic images with reference to the detected specularity, wherein the intrinsic images include an illumination map and a material map.

6. The computer program product of claim 5, wherein the feature vector is calculated from the token by application of Gabor filters to pixels of the token.

①: the colour of the source illumination is determined to implement the conversion into SUV space.



7. The computer program product of claim 5 or 6, wherein the step of segregating the image into intrinsic images with reference to the detected specularity is carried out by removing the detected specularity from the image to provide a specular free image.

5 8. The computer program product of claim 5 or 6, wherein the step of segregating the image into intrinsic images with reference to the detected specularity is carried out by generating a specular mask and segregating the image as a function of the specular mask.

10 Patentansprüche

1. Automatisiertes, computergestütztes Verfahren zur Bearbeitung eines Bildes, welches folgende Schritte umfasst:

- Bereitstellen einer Bilddatei (18), auf der ein Bild dargestellt ist, in einem Computerspeicher (16);
- 15 • Feststellen der Spiegelung im Bild durch
 - o Tokenisieren des Bildes, und für jedes Token
 - o Kennzeichnen des Tokens als spiegelnd, wenn das Token gesättigt ist, andernfalls Fortsetzen wie folgt:
 - 20 o Übertragen der Farbe des Tokens in einen SUV-Farbraum, wobei ein SUV-Farbraum ein Farbraum ist, in dem die Farbe der Quellbeleuchtung dazu bestimmt ist, die Konvertierung in den SUV-Raum zu implementieren;
 - o Betrachten des Tokens als Kandidaten für ein spiegelndes Token, wenn sowohl die Komponente der Farbe des Tokens in S-Richtung im Vergleich zur Komponente in der UV-Ebene groß ist als auch die
 - 25 Komponente in der UV-Ebene über einem bestimmten Schwellwert liegt, andernfalls Kennzeichnen des Tokens als nicht-spiegelnd,
 - o Im Fall eines Tokens, der ein Kandidat für einen spiegelnden Token ist, Übergang zum Extrahieren eines Merkmalsvektors aus dem Token, wobei die Merkmale im Merkmalsvektor indikativ für die Spiegelungseigenschaft sind;
 - 30 o Eingeben des Merkmalsvektors in einen Klassifikator;
 - o Kennzeichnen des Tokens als spiegelndes Token, wenn eine Wertausgabe aus dem Klassifikator über einem bestimmten Schwellwert liegt, andernfalls Kennzeichnen des Tokens als nicht-spiegelnd;
- und
- 35 • Trennen des Bildes in intrinsische Bilder unter Bezugnahme auf die festgestellte Spiegelung, wobei die intrinsischen Bilder einen Beleuchtungsplan und einen Materialplan enthalten.

2. Verfahren gemäß Anspruch 1, wobei der Merkmalsvektor aus dem Token durch die Anwendung von Gabor-Filtern auf Pixel des Tokens berechnet wird.

40 3. Verfahren gemäß Anspruch 1 oder 2, wobei der Schritt der Trennung des Bildes in intrinsische Bilder unter Bezugnahme auf die festgestellte Spiegelung durch Entfernen der festgestellten Spiegelung aus dem Bild zur Bereitstellung eines spiegelungsfreien Bildes durchgeführt wird.

45 4. Verfahren gemäß Anspruch 1 oder 2, wobei der Schritt der Trennung des Bildes in intrinsische Bilder unter Bezugnahme auf die festgestellte Spiegelung durch Generieren einer Spiegelungsmaske und Trennen des Bildes in Abhängigkeit von der Spiegelungsmaske durchgeführt wird.

50 5. Computerprogrammprodukt, das auf einem computerlesbaren Medium abgelegt ist, wobei das Produkt computergestützte Prozessschritte zur Steuerung eines Computers (10) umfasst, um:

in einem Computerspeicher (16) eine Bilddatei zu schaffen, die ein Bild darstellt, und Spiegelung in dem Bild festzustellen durch

- Tokenisieren des Bildes, und für jedes Token
- Kennzeichnen des Tokens als spiegelnd, wenn das Token gesättigt ist, andernfalls Fortsetzen wie folgt:
- Übertragen der Farbe des Tokens in einen SUV-Farbraum, wobei ein SUV-Farbraum ein Farbraum ist, in dem die Farbe der Quellbeleuchtung dazu bestimmt ist, die Konvertierung in den SUV-Raum zu imple-

mentieren;

• Betrachten des Tokens als Kandidaten für ein spiegelndes Token, wenn sowohl die Komponente der Farbe des Tokens in S-Richtung im Vergleich zur Komponente in der UV-Ebene groß ist als auch die Komponente in der UV-Ebene über einem bestimmten Schwellwert liegt, andernfalls Kennzeichnen des Tokens als nicht-spiegelnd;

• Im Fall eines Tokens, der ein Kandidat für einen spiegelnden Token ist, Übergang zum Extrahieren eines Merkmalsvektors aus dem Token, wobei die Merkmale im Merkmalsvektor indikativ für die Spiegelungseigenschaft sind;

• Eingeben des Merkmalvektors in einen Klassifikator;

• Kennzeichnen des Tokens als spiegelndes Token, wenn eine Wertaussage aus dem Klassifikator über einem bestimmten Schwellwert liegt, andernfalls Kennzeichnen des Tokens als nicht-spiegelnd,

und Trennen des Bildes in intrinsische Bilder unter Bezugnahme auf die festgestellte Spiegelung, wobei die intrinsischen Bilder einen Beleuchtungsplan und einen Materialplan enthalten.

6. Computerprogrammprodukt gemäß Anspruch 5, wobei der Merkmalsvektor aus dem Token durch die Anwendung von Gabor-Filtern auf Pixel des Tokens berechnet wird.

7. Computerprogrammprodukt gemäß Anspruch 5 oder 6, wobei der Schritt der Trennung des Bildes in intrinsische Bilder unter Bezugnahme auf die festgestellte Spiegelung durch Entfernen der festgestellten Spiegelung aus dem Bild zur Bereitstellung eines spiegelungsfreien Bildes durchgeführt wird.

8. Computerprogrammprodukt gemäß Anspruch 5 oder 6, wobei der Schritt der Trennung des Bildes in intrinsische Bilder unter Bezugnahme auf die festgestellte Spiegelung durch Generieren einer Spiegelungsmaske und Trennen des Bildes in Abhängigkeit von der Spiegelungsmaske durchgeführt wird.

Revendications

1. Procédé automatique sur ordinateur pour le traitement d'une image, comprenant les étapes de :

- fourniture d'un fichier d'image (18) décrivant une image dans une mémoire d'ordinateur (16) ;
- détection d'une spécularité dans l'image en

- segmentant (tokenisant) l'image, et pour chaque segment (token),

- en marquant le segment comme spéculaire si le segment est saturé, et en continuant sinon :

- en transformant la couleur du segment en espace de couleur SUV, un espace de couleur SUV étant un espace de couleur dans lequel la couleur de l'éclairage de la source est déterminée de façon à réaliser la conversion en espace SUV ;

- en considérant le segment comme un segment spéculaire potentiel si la composante de la couleur du segment dans la direction S est grande par rapport à la composante dans le plan UV et si la composante dans le plan UV est supérieure à un seuil prédéfini, et sinon en marquant le segment comme non spéculaire ;

- en poursuivant, dans le cas d'un segment spéculaire potentiel, de façon à extraire un vecteur de caractéristique du segment, les caractéristiques du vecteur de caractéristique étant indicatives d'une spécularité ;

- en entrant le vecteur de caractéristique dans un classificateur ;

- en marquant le segment comme un segment spéculaire si un score sorti par le classificateur dépasse un seuil prédéfini, ou en marquant sinon le segment comme non spéculaire ;

- et discriminer (ségréguer) l'image en images intrinsèques par rapport à la spécularité détectée, les images intrinsèques incluant une carte d'éclairage et une carte de matériau.

2. Procédé selon la revendication 1, dans lequel le vecteur de caractéristique est calculé à partir du segment par application de filtres de Gabor à des pixels du segment.

3. Procédé selon la revendication 1 ou 2, dans lequel l'étape de discrimination de l'image en images intrinsèques par rapport à la spécularité détectée est réalisée en éliminant la spécularité détectée de l'image pour obtenir une image sans spécularité.

EP 2 333 720 B1

4. Procédé selon la revendication 1 ou 2, dans lequel l'étape de discrimination de l'image en images intrinsèques par rapport à la specularité détectée est exécutée en générant un masque spéculaire et en discriminant l'image en fonction du masque spéculaire.

- 5 5. Produit de programme pour ordinateur disposé sur un support pouvant être lu par un ordinateur, lequel produit comprend des étapes de processus exécutables par un ordinateur permettant de commander un ordinateur (10) afin de :

fournir un fichier d'image décrivant une image dans une mémoire d'ordinateur (16) ;
10 détecter une specularité dans l'image en

- segmentant (tokenisant) l'image, et pour chaque segment (token) ;
- en marquant le segment comme spéculaire si le segment est saturé, et en continuant sinon :

- 15 • en transformant la couleur du segment en espace de couleur SUV, un espace de couleur SUV étant un espace de couleur dans lequel la couleur de l'éclairage de la source est déterminée de façon à réaliser la conversion en espace SUV ;
- 20 • en considérant le segment comme un segment spéculaire potentiel si la composante de la couleur du segment dans la direction S est grande par rapport à la composante dans le plan UV et si la composante dans le plan UV est supérieure à un seuil prédéfini, et sinon en marquant le segment comme non spéculaire ;
- 25 • en poursuivant, dans le cas d'un segment spéculaire potentiel, de façon à extraire un vecteur de caractéristique du segment, les caractéristiques du vecteur de caractéristique étant indicatives d'une specularité ;
- en entrant le vecteur de caractéristique dans un classificateur ;
- en marquant le segment comme un segment spéculaire si un score sorti par le classificateur dépasse un seuil prédéfini, ou en marquant sinon le segment comme non spéculaire ;

30 et discriminer (ségréguer) l'image en images intrinsèques par rapport à la specularité détectée, les images intrinsèques incluant une carte d'éclairage et une carte de matériau.

6. Produit de programme pour ordinateur selon la revendication 5, dans lequel le vecteur de caractéristique est calculé à partir du segment par application de filtres de Gabor à des pixels du segment.
- 35 7. Produit de programme pour ordinateur selon la revendication 5 ou 6, dans lequel l'étape de discrimination de l'image en images intrinsèques par rapport à la specularité détectée est réalisée en éliminant la specularité détectée de l'image pour obtenir une image sans specularité.
- 40 8. Produit de programme pour ordinateur selon la revendication 5 ou 6, dans lequel l'étape de discrimination de l'image en images intrinsèques par rapport à la specularité détectée est exécutée en générant un masque spéculaire et en discriminant l'image en fonction du masque spéculaire.

45

50

55

Computer System Configured to Operate on Images

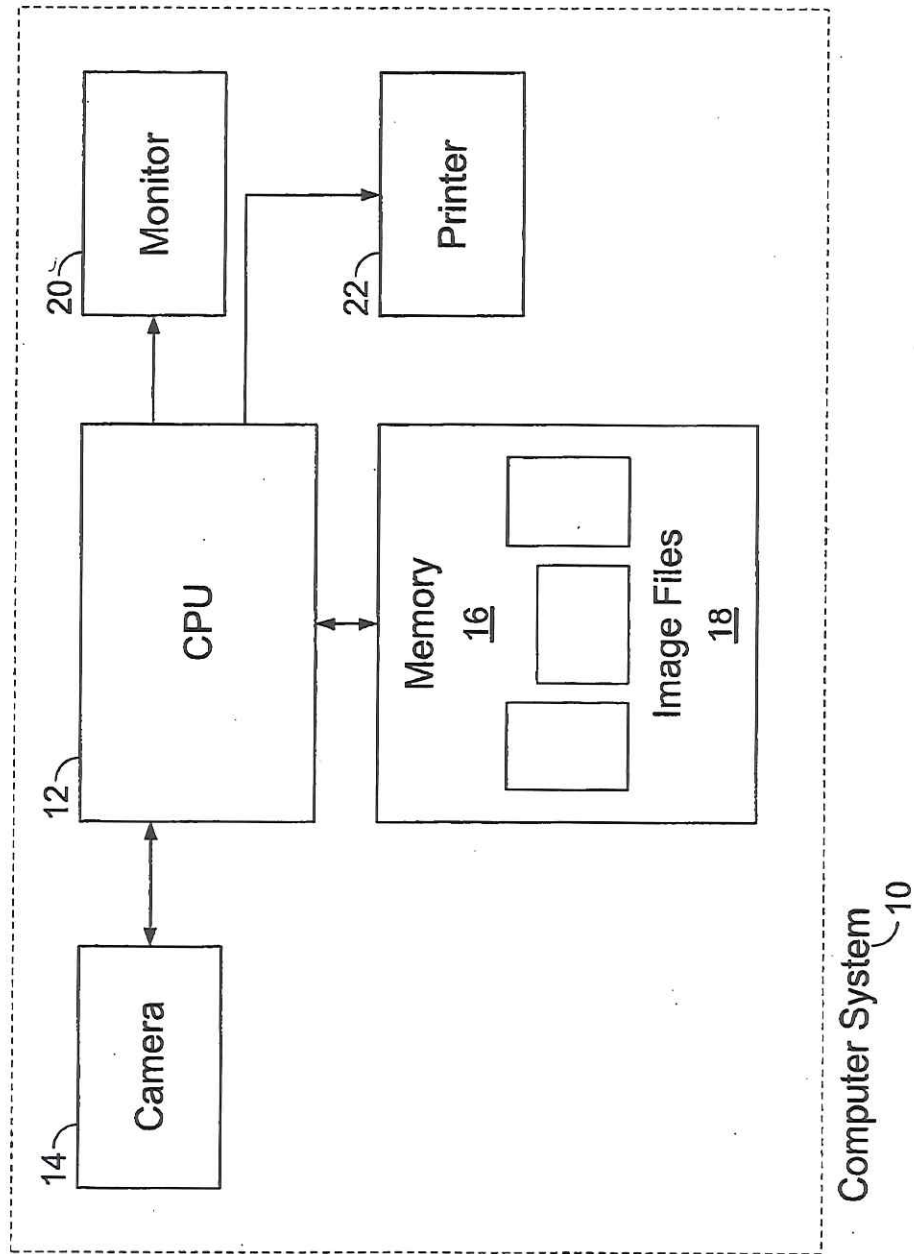


FIG. 1

Pixel Array for Storing Image Data

P(1, 1)	P(1, 2)	P(1, M)
P(2, 1)	P(2, 2)			⋮
P(3, 1)	P(3, 2)			⋮
⋮				
⋮				⋮
P(N, 1)	P(N, M)

Image File
18

FIG. 2

Integrating Specularity detector in Segregation Framework

Approach - 1

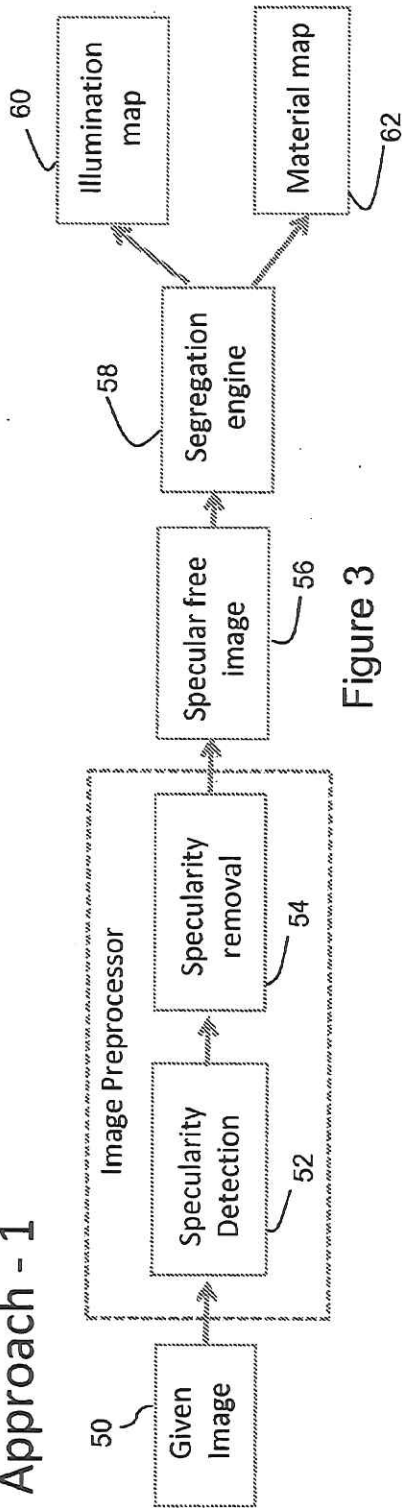


Figure 3

Approach - 2

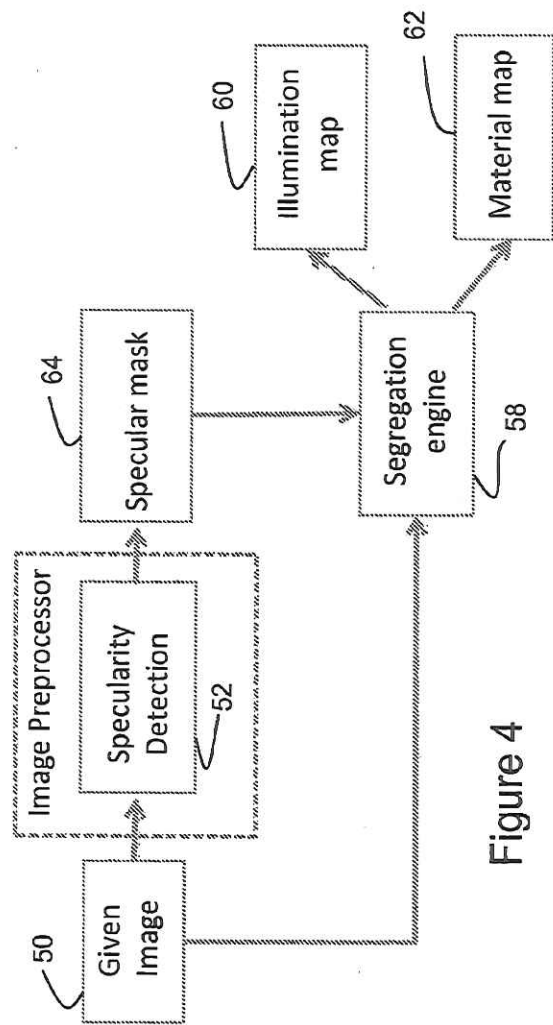


Figure 4

Specularity Detector

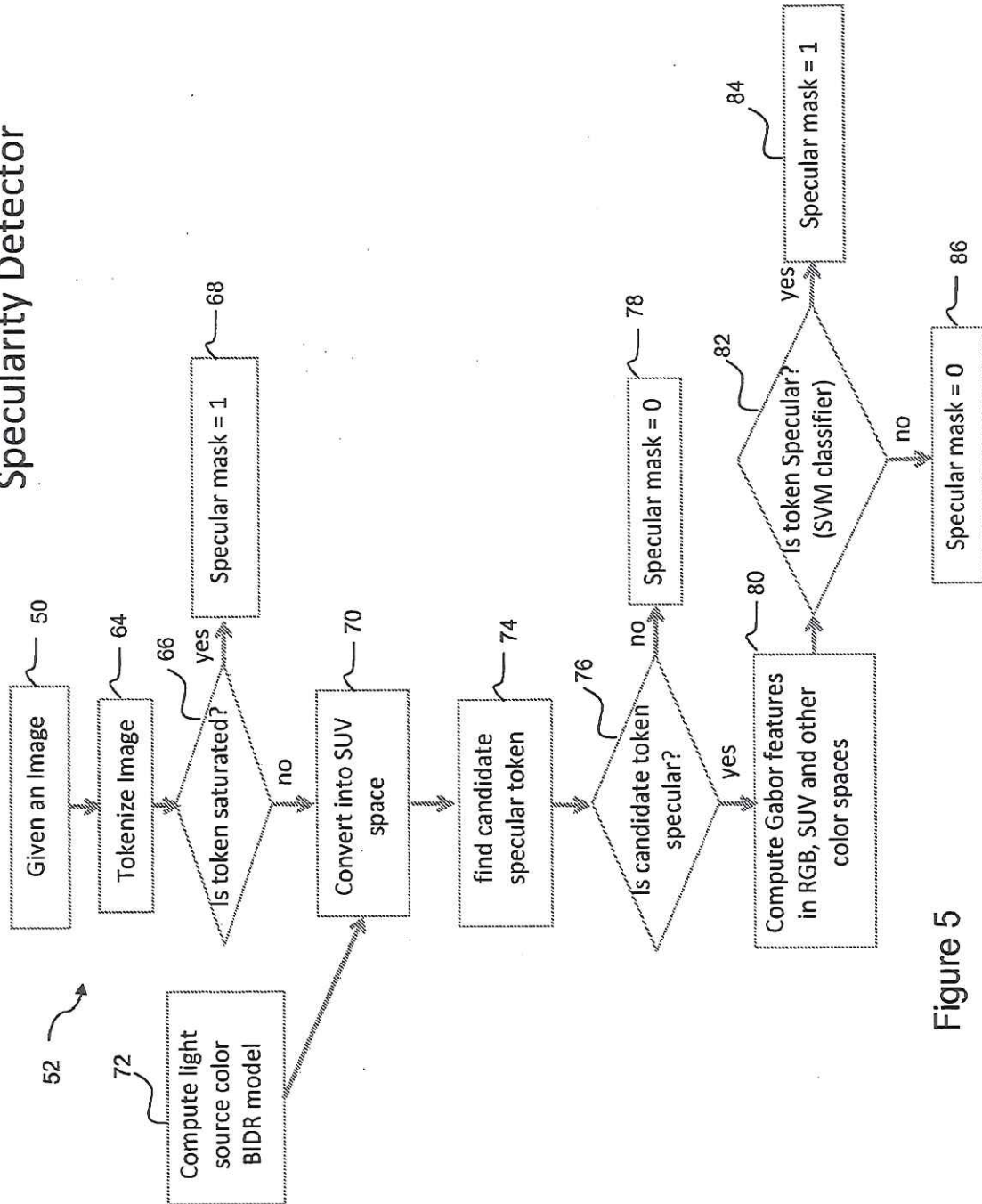


Figure 5

Training Framework for SVM Specularity Classifier

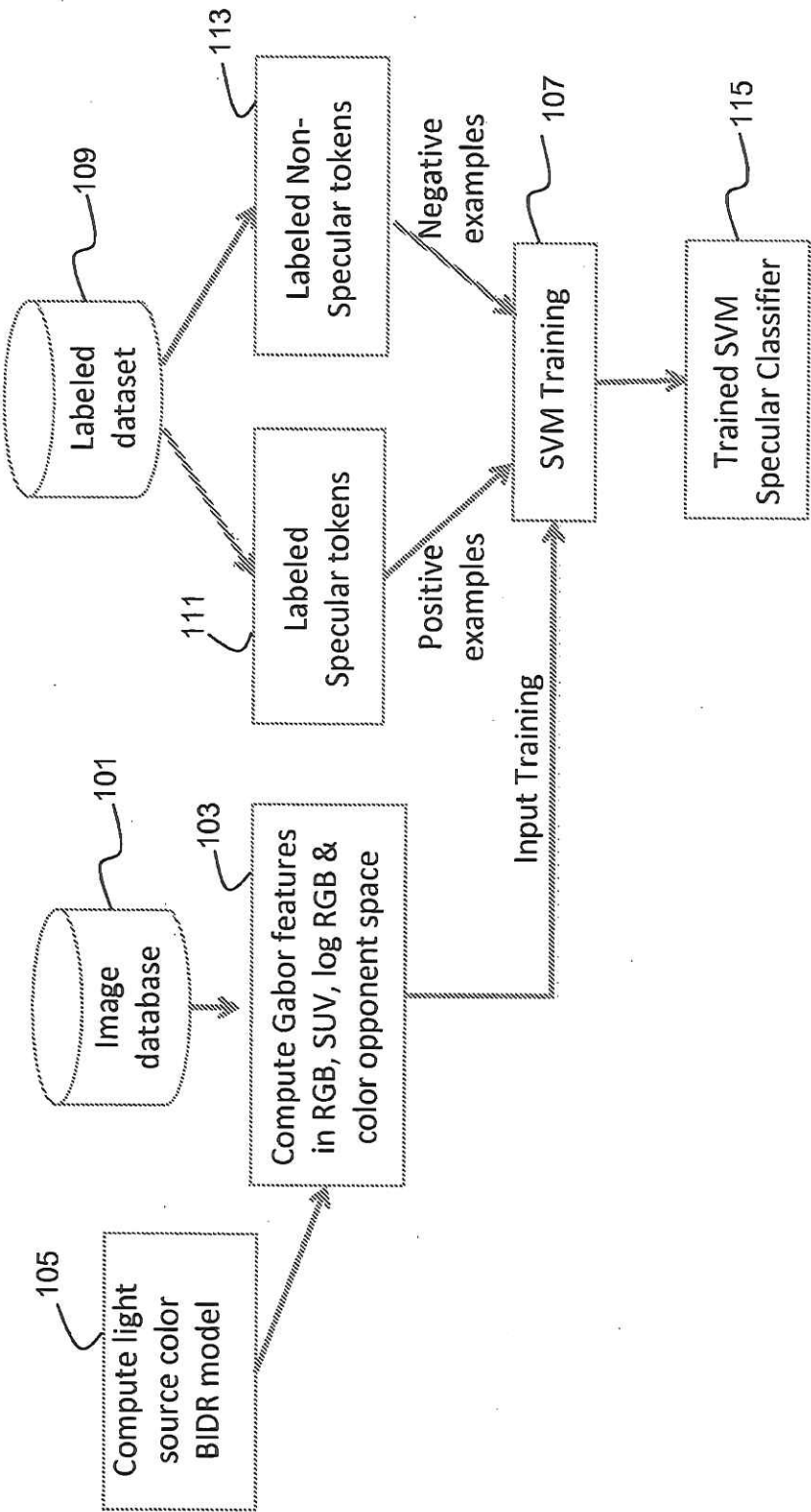


Figure 6

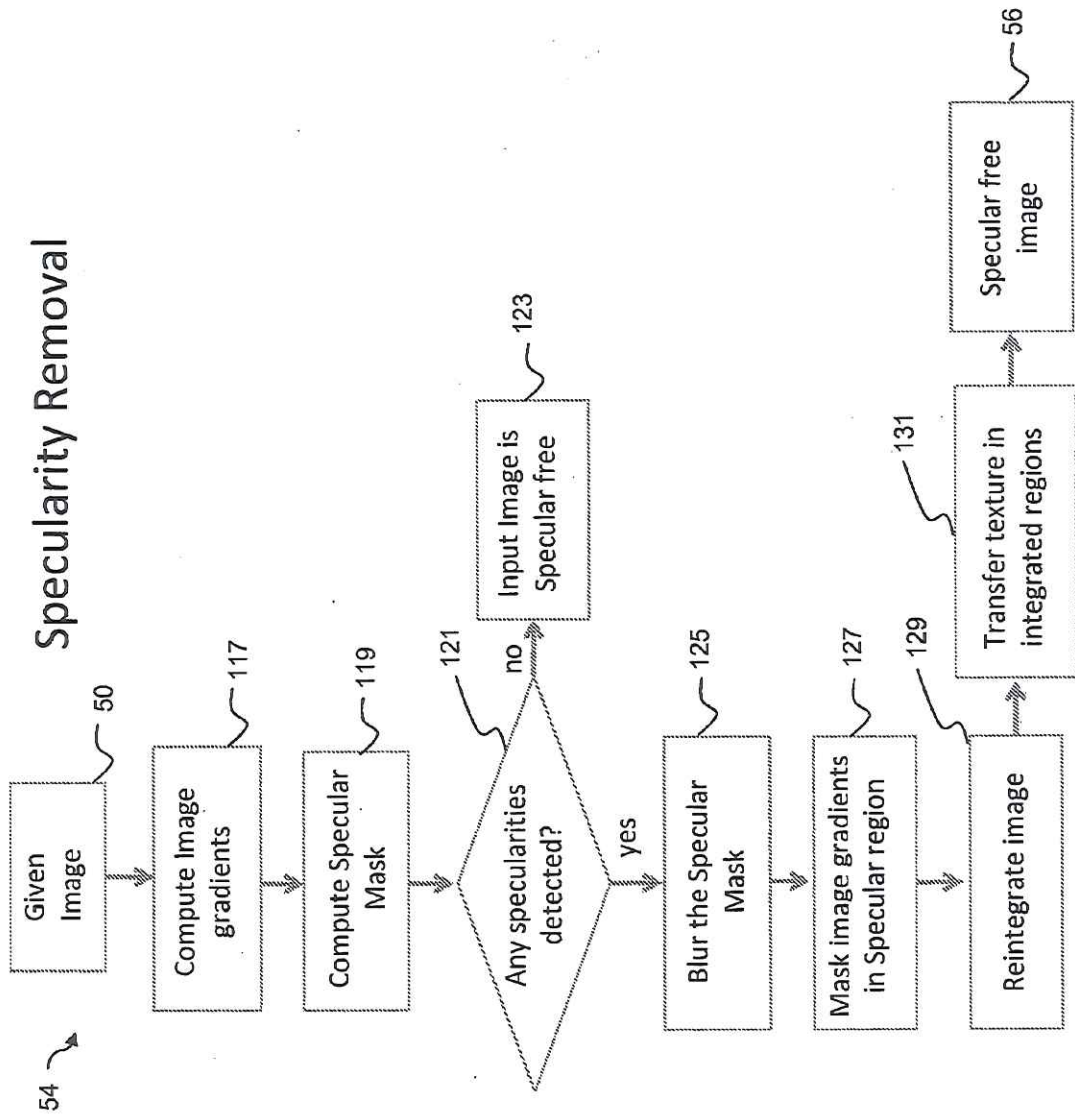


Figure 7

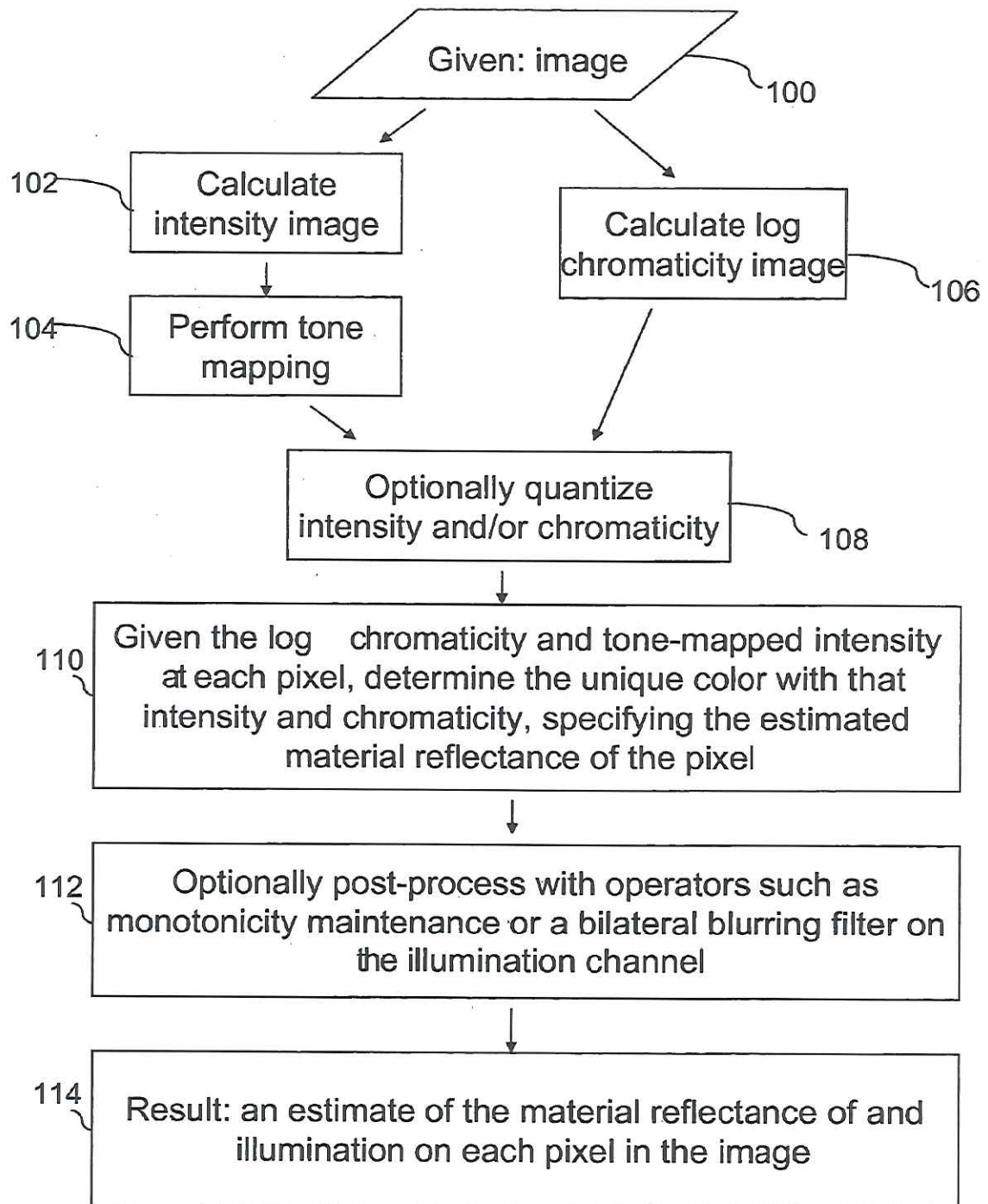


Figure 8

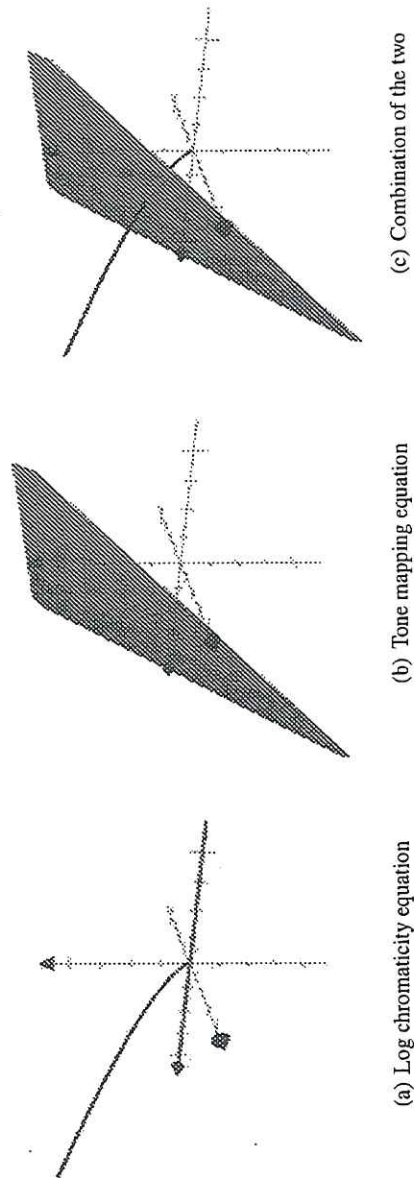


Figure 9

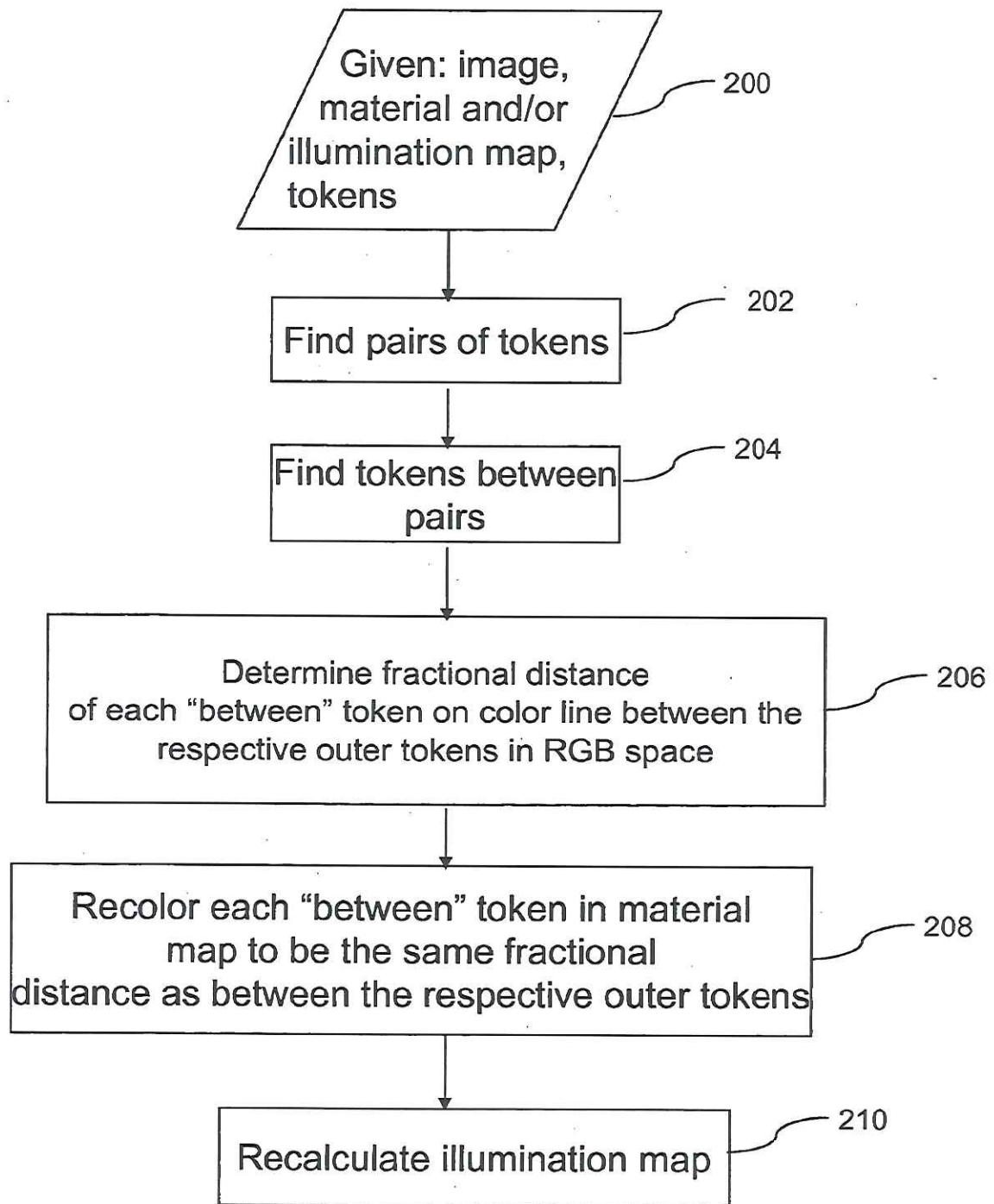


Figure 10

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 2008000015 W [0005]
- WO 2008085815 A1 [0005]
- US 20090034835 A [0020]
- US 20070176940 A [0026] [0057]
- US 20070242878 A [0056] [0068]
- US TIONNO20070242878 A [0071]

Non-patent literature cited in the description

- **M. F. TAPPEN et al.** Recovering Intrinsic Images from a Single Image. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, 1459-1472 [0002]
- **R. GROSSE et al.** Ground truth dataset and baseline evaluations for intrinsic image algorithms. *IEEE 12th Intern. Conference*, 2009, 2335-2342 [0003]
- **YUN-CHUNG CHUNG et al.** Intrinsic Image Extraction from a Single Image. *Journal of information science and engineering*, vol. 25, 1939-1953 [0004]
- **A. DELPOZO et al.** Detecting Specular Surfaces on Natural Images. *IEEE conference on computer vision and pattern recognition*, 2007, 1-8 [0006]
- **TODD ZICKLER et al.** Color Subspaces as Photometric Invariants. *International Journal of Computer Vision*, 13 August 2008, vol. 79 (1 [0024]
- A Bi-illuminant Dichromatic Reflection Model for Understanding Images. *CVPR08*, 2008 [0026] [0027]