(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 January 2014 (03.01.2014)

(10) International Publication Number
WO 2014/004271 A2

(51) International Patent Classification:
G06K 9/78 (2006.01) *H04N 7/18* (2006.01)

(21) International Application Number:
PCT/US2013/046971

(22) International Filing Date:
21 June 2013 (21.06.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/532,106 25 June 2012 (25.06.2012) US

(71) Applicant: **TANDENT VISION SCIENCE, INC.** [US/US]; 505 Montgomery Street, Suite 1100, San Francisco, CA 94111 (US).

(72) Inventors: **FRIEDHOFF, Richard, Mark**; 505 Montgomery Street, San Francisco, CA 94111 (US). **MAXWELL, Bruce, Allen**; 44 Denico Lane, Benton, ME 04901-3634 (US). **SMITH, Casey, Arthur**; 2013 Overlook Dr, Grand Junction, CO 81505 (US).

(74) Agents: **DAVIDSON, Clifford, M.** et al.; Davidson, Davidson & Kappel, LLC, 485 Seventh Avenue, 14th Floor, New York, NY 10018 (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, BN, 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, 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, PA, 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, 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,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR USE OF INTRINSIC IMAGES IN AN AUTOMOTIVE DRIVER-VEHICLE-ASSISTANCE DEVICE

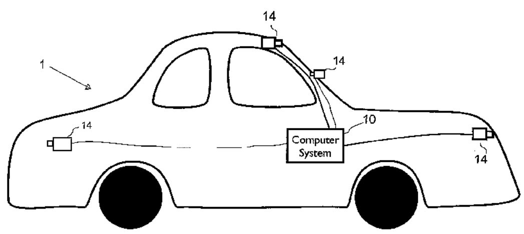


Fig 1a

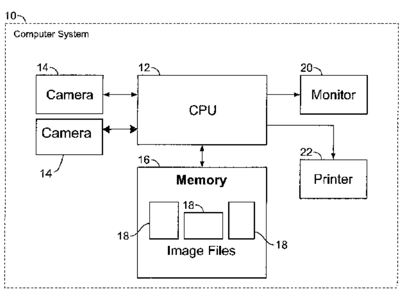


Fig. 1b

(57) **Abstract**: In a first 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 arranging a digital camera on a vehicle body, operating the digital camera to provide an image file depicting an image of a scene related to vehicle operation, in a computer memory, generating an intrinsic representation of the image and using the intrinsic representation to analyze the scene.

WO 2014/004271 A2

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

METHOD AND SYSTEM FOR USE OF INTRINSIC IMAGES IN AN AUTOMOTIVE DRIVER-VEHICLE-ASSISTANCE DEVICE

Background of the Invention

[0001] Many significant and commercially important uses of modern computer technology relate to images. These include image processing, image analysis and computer vision applications. In computer vision applications, such as, for example, object recognition and optical character recognition, it has been found that a separation of illumination and material aspects of an image can significantly improve the accuracy of computer performance. Significant pioneer inventions related to the illumination and material aspects of an image are disclosed in U. S. Patent No. 7,873,219 to Richard Mark Friedhoff, entitled Differentiation Of Illumination And Reflection Boundaries and U. S. Patent No. 7,672,530 to Richard Mark Friedhoff et al., entitled Method And System For Identifying Illumination Flux In An Image (hereinafter the Friedhoff Patents).

Summary of the Invention

[0002] The present invention provides an improvement and enhancement to the fundamental teachings of the Friedhoff Patents, and includes a method and system comprising image techniques that accurately and correctly generate intrinsic images for use in an automotive driver-vehicle-assistance device.

[0003] In a first 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 arranging a digital camera on a vehicle body, operating the digital camera to provide an image file depicting an image of a scene related to vehicle operation, in a computer memory, generating an intrinsic representation of the image and using the intrinsic representation to analyze the scene.

[0004] In a second exemplary embodiment of the present invention, a device is provided. The device comprises a digital camera adapted for mounting on a vehicle to record an image of a scene related to vehicle operation, a computer system adapted to be coupled to the digital camera , the computer system including a memory storing an image file containing the image of the scene and wherein the computer system is arranged and configured to execute a routine to generate an intrinsic representation of the image, and use the intrinsic representation to analyze the scene.

[0005] In a third exemplary embodiment of the present invention, a computer program product, disposed on a computer readable media is provided. The computer program product includes computer executable process steps operable to control a computer to: receive an image file depicting an image of a scene related to vehicle operation, in a computer memory, generate an intrinsic representation of the image and use the intrinsic representation to analyze the scene.

[0006] In a fourth exemplary embodiment of the present invention, a computer program product, disposed on a computer readable media is provided. The computer program product includes computer executable process steps operable to control a computer to: receive an image file depicting an image of a scene related to vehicle driver appearance, in a computer memory, generate an intrinsic representation of the image and use the intrinsic representation to analyze the vehicle driver appearance.

[0007] 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 methods described above. In accordance with other embodiments of the present invention, non-transitory 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.

Brief Description of the Drawings

[0008] Figure 1a is a schematic illustration of a motor vehicle, including a computer system arranged as a driver-assistance device, according to a feature of the present invention.

[0009] Figure 1b is a block diagram of the computer system of figure 1a, arranged and configured to perform operations related to images, according to a feature of the present invention.

[0010] Figure 2 shows an n X m pixel array image file for an image stored in the computer system of figure 1b.

[0011] Figure 3a is a flow chart for identifying Type C token regions in the image file of figure 2, according to a feature of the present invention.

[0012] Figure 3b is an original image used as an example in the identification of Type C tokens.

[0013] Figure 3c shows Type C token regions in the image of figure 3b.

[0014] Figure 3d shows Type B tokens, generated from the Type C tokens of figure 3c, according to a feature of the present invention.

[0015] Figure 4 is a flow chart for a routine to test Type C tokens identified by the routine of the flow chart of figure 3a, according to a feature of the present invention.

[0016] Figure 5 is a graphic representation of a log color space chromaticity plane according to a feature of the present invention.

[0017] Figure 6 is a flow chart for determining a list of colors depicted in an input image.

[0018] Figure 7 is a flow chart for determining an orientation for a log chromaticity space.

[0019] Figure 8 is a flow chart for determining log chromaticity coordinates for the colors of an input image, as determined through execution of the routine of figure 6.

[0020] Figure 9 is a flow chart for augmenting the log chromaticity coordinates, as determined through execution of the routine of figure 8.

[0021] Figure 10 is a flow chart for clustering the log chromaticity coordinates, according to a feature of the present invention.

[0022] Figure 10a is an illustration of a grid for a spatial hash, according to a feature of the present invention.

[0023] Figure 11 is a flow chart for assigning the log chromaticity coordinates to clusters determined through execution of the routine of figure 10.

[0024] Figure 12 is a flow chart for detecting regions of uniform reflectance based on the log chromaticity clustering.

[0025] Figure 13 is a representation of an [A] [x] = [b] matrix relationship used to identify and separate illumination and material aspects of an image, according to a same-material constraint, for generation of intrinsic images.

[0026] Figure 14 illustrates intrinsic images including an illumination image and a material image corresponding to the original image of figure 3b.

[0027] Figure 15 is a flow chart for an edge preserving blur post processing technique applied to the intrinsic images illustrated in figure 14, according to a feature of the present invention.

[0028] Figure 16 is a flow chart for an artifact reduction post processing technique applied to the intrinsic images illustrated in figure 14, according to a feature of the present invention.

[0029] Figure 17 is a flow chart for a BIDR model enforcement post processing technique applied to the intrinsic images illustrated in figure 14, according to a feature of the present invention.

[0030] Figure 18 is a graph in RGB color space showing colors for a material, from a fully shaded color value to a fully lit color value, as predicted by a bi-illuminant dichromatic reflection model.

[0031] Figure 19 is a flow chart for a driver assistance method according to a feature of the present invention.

Detailed Description of the Preferred Embodiments

[0032] Referring now to the drawings, and initially to figure 1a, there is shown a motor vehicle such as, for example, an automobile 1. A computer system 10 is mounted within the automobile 1, and is coupled to one or more cameras 14, arranged at various locations on the automobile 1. The arrangement of the cameras 14 is such that selected cameras 14 are focused on scenes relevant to vehicle operation, for example, the road scene in front of the vehicle, the driver, the road scene behind the vehicle, and so on. Accordingly, various scenes depicting views of the road and the driver can be input to the computer system 10. The various scenes, in addition to analysis for generating intrinsic images, as will be described, can be arranged to provide stereo pairs of the road scene images for use by the computer system 10 to calculate depth information.

[0033] Figure 1b shows a block diagram of the computer system 10 arranged and configured to perform operations related to images. A CPU 12 of the computer system 10 is coupled to each of the digital cameras 14 via, for example, a USB port. The digital cameras 14 operate to download images of the road scenes and/or the driver recorded and stored locally on the cameras 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 use in a vehicle driver-vehicle-assistance operation.

[0034] 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.

[0035] As shown in figure 2, each image file 18 comprises an n X 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.

[0036] In an image operation, the CPU 12 operates to analyze the RGB values of the pixels of a stored image file 18 to achieve various objectives, such as, for example, to identify regions of an image that correspond to a single material depicted in a scene recorded in the image file 18. 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. A method for detecting of one of these components, for example, material, provides a mechanism for distinguishing material or object geometry, such as object edges, from illumination and shadow boundaries.

[0037] Such a mechanism enables techniques that can be used to generate 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).

[0038] To advantage, the separation of illumination from material in the intrinsic images, according to the teachings of the present invention, provides the CPU 12 with images optimized for more effective and accurate further processing. For example, in the automobile 1, including the computer system 10, as shown in figure 1a, intrinsic material reflectance images of a road scene and/or of the driver, enable a more accurate and precise analysis by the CPU 12 of road features, such as, for example, lane markings and road signs, and driver identification, appearance and behavior, without the complexity in image appearance that can be caused by varying illumination present at the time the scene was recorded (for example, shadows).

[0039] Pursuant to a feature of the present invention, processing is performed at a token level. A token is a connected region of an image wherein the pixels of the region are related to one another in a manner relevant to identification of image features and characteristics such as an identification of materials and illumination. The pixels of a token can be related in terms of either homogeneous factors, such as, for example, close correlation of color among the pixels, or inhomogeneous factors, such as, for example, differing color values related geometrically in a color space such as RGB space, commonly referred to as a texture. The present invention utilizes spatio-spectral information relevant to contiguous pixels of an image depicted in an image file 18 to identify token regions. The spatio-spectral information includes spectral relationships among contiguous pixels, in terms of color bands, for example the RGB values of the pixels, and the spatial extent of the pixel spectral characteristics relevant to a single material.

[0040] According to one exemplary embodiment of the present invention, tokens are each classified as either a Type A token, a Type B token or a Type C token. A Type A token is a connected image region comprising contiguous pixels that represent the largest possible region of the image encompassing a single material in the scene (uniform reflectance). A Type B token is a connected image region comprising contiguous pixels that represent a region of the image encompassing a single material in the scene, though not necessarily the maximal region of uniform reflectance corresponding to that material. A Type B token can also be defined as a collection of one or more image regions or pixels, all of which have the same reflectance (material color) though not necessarily all pixels which correspond to that material color. A Type C token comprises a connected image region of similar image properties among the contiguous pixels of the token, where similarity is defined with respect to a noise model for the imaging system used to record the image.

[0041] Referring now to figure 3a, there is shown a flow chart for identifying Type C token regions in the scene depicted in the image file 18 of figure 2, according to a feature of the present invention. Type C tokens can be readily identified in an image, utilizing the steps of figure 3a, and then analyzed and processed to construct Type B tokens, according to a feature of the present invention.

[0042] A 1st order uniform, homogeneous Type C token comprises a single robust color measurement among contiguous pixels of the image. At the start of the identification routine, the CPU 12 sets up a region map in memory. In step 100, the CPU 12 clears the region map and assigns a region ID, which is initially set at 1. An iteration for the routine, corresponding to a pixel number, is set at $I = 0$, and a number for an N x N pixel array, for use as a seed to determine the token, is set an initial value, $N = N_{start}$. N_{start} can be any integer > 0, for example it can be set at set at 11 or 15 pixels.

[0043] At step 102, a seed test is begun. The CPU 12 selects a first pixel, $i = 1$, pixel (1, 1) for example (see figure 2), the pixel at the upper left corner of a first N x N sample of the image file 18. The pixel is then tested in decision block 104 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 x N array. The color values comparison can be with respect to multiple color band values (RGB in our example) of the pixel. If the comparison does not result in approximately equal values (within the noise levels of the recording device) for the pixels in the seed, the CPU 12 increments the value of i (step 106), for example, $i = 2$, pixel (1, 2), for a next N x N seed sample, and then tests to determine if $i = i_{max}$ (decision block 108).

[0044] If the pixel value is at i_{max}, a value selected as a threshold for deciding to reduce the seed size for improved results, the seed size, N, is reduced (step 110), for example, from N = 15 to N = 12. In an exemplary embodiment of the present invention, i_{max} can be set at a number of pixels in an image ending at pixel (n, m), as shown in figure 2. In this manner, the routine of figure 3a parses the entire image at a first value of N before repeating the routine for a reduced value of N.

[0045] After reduction of the seed size, the routine returns to step 102, and continues to test for token seeds. An N_{stop} value (for example, N = 2) is also checked in step 110 to determine if the analysis is complete. If the value of N is at N_{stop} , the CPU 12 has completed a survey of the image pixel arrays and exits the routine.

[0046] If the value of i is less than i_{max} , and N is greater than N_{stop}, the routine returns to step 102, and continues to test for token seeds.

[0047] When a good seed (an N x N array with approximately equal pixel values) is found (block 104), the token is grown from the seed. In step 112, the CPU 12 pushes the pixels from the seed onto a queue. All of the pixels in the queue are marked with the current region ID in the region map. The CPU 12 then inquires as to whether the queue is empty (decision block 114). If the queue is not empty, the routine proceeds to step 116.

[0048] In step 116, the CPU 12 pops the front pixel off the queue and proceeds to step 118. In step 118, the CPU 12 marks "good' neighbors around the subject pixel, that is neighbors approximately equal in color value to the subject pixel, with the current region ID. All of the marked good neighbors are placed in the region map and also pushed onto the queue. The CPU 12 then returns to the decision block 114. The routine of steps 114, 116, 118 is repeated until the queue is empty. At that time, all of the pixels forming a token in the current region will have been identified and marked in the region map as a Type C token.

[0049] When the queue is empty, the CPU 12 proceeds to step 120. At step 120, the CPU 12 increments the region ID for use with identification of a next token. The CPU 12 then returns to step 106 to repeat the routine in respect of the new current token region.

[0050] Upon arrival at $N = N_{stop}$, step 110 of the flow chart of figure 3a, or completion of a region map that coincides with the image, the routine will have completed the token building task. Figure 3b is an original image used as an example in the identification of tokens. The image shows areas of the color blue and the blue in shadow, and of the color teal and the teal in shadow. Figure 3c shows token regions corresponding to the region map, for example, as identified through execution of the routine of figure 3a (Type C tokens), in respect to the image of figure 3b. The token regions are color coded to illustrate the token makeup of the image of figure 3b, including penumbra regions between the full color blue and teal areas of the image and the shadow of the colored areas.

[0051] While each Type C token comprises a region of the image having a single robust color measurement among contiguous pixels of the image, the token may grow across material boundaries. Typically, different materials connect together in one Type C token via a neck region often located on shadow boundaries or in areas with varying illumination crossing different materials with similar hue but different intensities. A neck pixel can be identified by examining characteristics of adjacent pixels. When a pixel has two contiguous pixels on opposite sides that are not within the corresponding token, and two contiguous pixels on opposite sides that are within the corresponding token, the pixel is defined as a neck pixel.

[0052] Figure 4 shows a flow chart for a neck test for Type C tokens. In step 122, the CPU 12 examines each pixel of an identified token to determine whether any of the pixels under examination forms a neck. The routine of figure 4 can be executed as a subroutine directly after a particular token is identified during execution of the routine of figure 3a. All pixels identified as a neck are marked as "ungrowable." In decision block 124, the CPU 12 determines if any of the pixels were marked.

[0053] If no, the CPU 12 exits the routine of figure 4 and returns to the routine of figure 3a (step 126).

[0054] If yes, the CPU 12 proceeds to step 128 and operates to regrow the token from a seed location selected from among the unmarked pixels of the current token, as per the routine of figure 3a, without changing the counts for seed size and region ID. During the regrowth process, the CPU 12 does not include any pixel previously marked as ungrowable. After the token is regrown, the previously marked pixels are unmarked so that other tokens may grow into them.

[0055] Subsequent to the regrowth of the token without the previously marked pixels, the CPU 12 returns to step 122 to test the newly regrown token. Neck testing identifies Type C tokens that cross material boundaries, and regrows the identified tokens to provide single material Type C tokens suitable for use in creating Type B tokens.

[0056] Figure 3d shows Type B tokens generated from the Type C tokens of figure 3c, according to a feature of the present invention. The present invention provides a novel exemplary technique using log chromaticity clustering, for constructing Type B tokens for an image file 18. Log chromaticity is a technique for developing an illumination invariant chromaticity space.

[0057] A method and system for separating illumination and reflectance using a log chromaticity representation is disclosed in U. S. Patent No. 7,596,266, which is hereby expressly incorporated by reference. The techniques taught in U. S. Patent No. 7,596,266 can be used to provide illumination invariant log chromaticity representation values for each color of an image, for example, as represented by Type C tokens. 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 a bi-illuminant dichromatic reflection model (BIDR model), to provide a log chromaticity value for each pixel, as taught in U. S. Patent No. 7,596,266. The BIDR Model predicts that differing color measurement values fall within a cylinder 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.

[0058] Figure 5 is a graphic representation of a log color space, bi-illuminant chromaticity plane according to a feature of the invention disclosed in U. S. Patent No. 7,596,266. The alignment of the chromaticity plane is determined by a vector N, normal to the chromaticity plane, and defined as $N = \log(\text{Bright}_{\text{vector}}) - \log(\text{Dark}_{\text{vector}}) = \log(1 + 1/S_{\text{vector}})$. The co-ordinates of the plane, u, v can be defined by a projection of the green axis onto the chromaticity plane as the u axis, and the cross product of u and N being defined as the v axis. In our example, each log value for the materials A, B, C is projected onto the chromaticity plane, and will therefore have a corresponding u, v co-ordinate value in the plane that is a chromaticity value, as shown in figure 5.

[0059] Thus, according to the technique disclosed in U. S. Patent No. 7,596,266, the RGB values of each pixel in an image file 18 can be mapped by the CPU 12 from the image file value p(n, m, R, G, B) to a log value, then, through a projection to the chromaticity plane, to the corresponding u, v value, as shown in figure 5. Each pixel p(n, m, R, G, B) in the image file 18 is then replaced by the CPU 12 by a two dimensional chromaticity value: p(n, m, u, v), to provide a chromaticity representation of the original RGB image. In general, for an N band image, the N color values are replaced by N - 1 chromaticity values. The chromaticity representation is a truly accurate illumination invariant representation because the BIDR model upon which the representation is based, accurately and correctly represents the illumination flux that caused the original image.

[0060] According to the invention disclosed and claimed in related U.S. Patent Application Publication 2012/0114232 published on May 10, 2012, filed November 10, 2010, entitled System and Method for Identifying Complex Tokens in an Image (expressly incorporated by reference herein and hereinafter referred to as "related invention"), log chromaticity values are calculated for each color depicted in an image file 18 input to the CPU 12 for identification of regions of the uniform reflectance (Type B tokens). For example, each pixel of a Type C token will be of approximately the same color value, for example, in terms of RGB values, as all the other constituent pixels of the same Type C token, within the noise level of the equipment used to record the image. Thus, an average of the color values for the constituent pixels of each particular Type C token can be used to represent the color value for the respective Type C token in the log chromaticity analysis.

[0061] Figure 6 is a flow chart for determining a list of colors depicted in an input image, for example, an image file 18. In step 200, an input image file 18 is input to the CPU 12 for processing. In steps 202 and 204, the CPU 12 determines the colors depicted in the input image file 18. In step 202, the CPU 12 calculates an average color for each Type C token determined by the CPU 12 through execution of the routine of figure 3a, as described above, for a list of colors. The CPU 12 can be operated to optionally require a minimum token size, in terms of the number of constituent pixels of the token, or a minimum seed size (the N x N array) used to determine Type C tokens according to the routine of figure 3a, for the analysis. The minimum size requirements are implemented to assure that color measurements in the list of colors for the image are an accurate depiction of color in a scene depicted in the input image, and not an artifact of blend pixels.

[0062] 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. Thus, each blend pixel does not represent a true color of the image. If blend pixels are present, relatively small Type C tokens, consisting of blend pixels, can be identified for areas of an image between two differently colored regions. By requiring a size minimum, the CPU 12 can eliminate tokens consisting of blend pixel from the analysis.

[0063] In step 204, the CPU 12 can alternatively collect colors at the pixel level, that is, the RGB values of the pixels of the input image file 18, as shown in figure 2. The CPU 12 can be operated to optionally require each pixel of the image file 18 used in the analysis to have a minimum stability or local standard deviation via a filter output, for a more accurate list of colors. For example, second derivative energy can be used to indicate the stability of pixels of an image.

[0064] In this approach, the CPU 12 calculates a second derivative at each pixel, or a subset of pixels disbursed across the image to cover all illumination conditions of the image depicted in an input image file 18, using a Difference of Gaussians, Laplacian of Gaussian, or similar filter. The second derivative energy for each pixel examined can then be calculated by the CPU 12 as the average of the absolute value of the second derivative in each color band (or the absolute value of the single value in a grayscale image), the sum of squares of the values of the second derivatives in each color band (or the square of the single value in a grayscale image), the maximum squared second derivative value across the color bands (or the square of the single value in a grayscale image), or any similar method. Upon the calculation of the second derivative energy for each of the pixels, the CPU 12 analyzes the energy values of the pixels. There is an inverse relationship between second derivative energy and pixel stability, the higher the energy, the less stable the corresponding pixel.

[0065] In step 206, the CPU 12 outputs a list or lists of color (after executing one or both of steps 202 and/or 204). According to a feature of the related invention, all of the further processing can be executed using the list from either step 202 or 204, or vary the list used (one or the other of the lists from steps 202 or 204) at each subsequent step.

[0066] Figure 7 is a flow chart for determining an orientation for a log chromaticity representation, according to a feature of the related invention. For example, the CPU 12 determines an orientation for the normal N, for a log chromaticity plane, as shown in figure 5. In step 210, the CPU 12 receives a list of colors for an input file 18, such as a list output in step 206 of the routine of figure 6. In step 212, the CPU 12 determines an orientation for a log chromaticity space.

[0067] As taught in U. S. Patent No. 7,596,266, and as noted above, orientation of the chromaticity plane is represented by N, N being a vector normal to the chromaticity representation, for example, the chromaticity plane of figure 5. The orientation is estimated by the CPU 12 thorough execution of any one of several techniques. For example, the CPU 12 can determine estimates based upon entropy minimization, manual selection of lit/shadowed regions of a same material by a user or the use of a characteristic spectral ratio (which corresponds to the orientation N) for an image of an input image file 18, as fully disclosed in U. S. Patent No. 7,596,266.

[0068] For a higher dimensional set of colors, for example, an RYGB space (red, yellow, green, blue), the log chromaticity normal, N, defines a sub-space with one less dimension than the input space. Thus, in the four dimensional RYGB space, the normal N defines a three dimensional log chromaticity space. When the four dimensional RYGB values are projected into the three dimensional log chromaticity space, the projected values within the log chromaticity space are unaffected by illumination variation.

[0069] In step 214, the CPU 12 outputs an orientation for the normal N. As illustrated in the example of figure 5, the normal N defines an orientation for a u, v plane in a three dimensional RGB space.

[0070] Figure 8 is a flow chart for determining log chromaticity coordinates for the colors of an input image, as identified in steps 202 or 204 of the routine of figure 6. In step 220, a list of colors is input to the CPU 12. The list of colors can comprise either the list generated through execution of step 202 of the routine of figure 6, or the list generated through execution of step 204. In step 222, the log chromaticity orientation for the normal, N, determined through execution of the routine of figure 7, is also input to the CPU 12.

[0071] In step 224, the CPU 12 operates to calculate a log value for each color in the list of colors and plots the log values in a three dimensional log space at respective (log R, log G, log B) coordinates, as illustrated in figure 5. Materials A, B and C denote log values for specific colors from the list of colors input to the CPU 12 in step 220. A log chromaticity plane is also calculated by the CPU 12, in the three dimensional log space, with u, v coordinates and an orientation set by N, input to the CPU 12 in step 222. Each u, v coordinate in the log chromaticity plane can also be designated by a corresponding (log R, log G, log B) coordinate in the three dimensional log space.

[0072] According to a feature of the related invention, the CPU 12 then projects the log values for the colors A, B and C onto the log chromaticity plane to determine a u, v log chromaticity coordinate for each color. Each u, v log chromaticity coordinate can be expressed by the corresponding (log R, log G, log B) coordinate in the three dimensional log space. The CPU 12 outputs a list of the log chromaticity coordinates in step 226. The list cross-references each color to a u, v log chromaticity coordinate and to the pixels (or a Type C tokens) having the respective color (depending upon the list of colors used in the analysis (either step 202(tokens) or 204 (pixels))).

[0073] Figure 9 is a flow chart for optionally augmenting the log chromaticity coordinates for pixels or Type C tokens with extra dimensions, according to a feature of the related invention. In step 230, the list of log chromaticity coordinates, determined for the colors of the input image through execution of the routine of figure 8, is input to the CPU 12. In step 232, the CPU 12 accesses the input image file 18, for use in the augmentation.

[0074] In step 234, the CPU 12 optionally operates to augment each log chromaticity coordinate with a tone mapping intensity for each corresponding pixel (or Type C token). The tone mapping intensity is determined using any known tone mapping technique. An augmentation with tone mapping intensity information provides a basis for clustering pixels or tokens that are grouped according to both similar log chromaticity coordinates and similar tone mapping intensities. This improves the accuracy of a clustering step.

[0075] In step 236, the CPU 12 optionally operates to augment each log chromaticity coordinate with x, y coordinates for the corresponding pixel (or an average of the x, y coordinates for the constituent pixels of a Type C token) (see figure 2 showing a P (1,1) to P (N, M) pixel arrangement). Thus, a clustering step with x, y coordinate information will provide groups in a spatially limited arrangement, when that characteristic is desired.

[0076] In each of steps 234 and 236, the augmented information can, in each case, be weighted by a factor w_1 and w_2, w_3 respectively, to specify the relative importance and scale of the different dimensions in the augmented coordinates. The weight factors w_1 and w_2, w_3 are user-specified. Accordingly, the (log R, log G, log B) coordinates for a pixel or Type C token is augmented to (log R, log G, log B, $T*w_1$, $x*w_2$, $y*w_3$) where T, x and y are the tone mapped intensity, the x coordinate and the y coordinate, respectively.

[0077] In step 238, the CPU 12 outputs a list of the augmented coordinates. The augmented log chromaticity coordinates provide accurate illumination invariant representations of the pixels, or for a specified regional arrangement of an input image, such as, for example, Type C tokens. According to a feature of the related invention and the present invention, the illumination invariant characteristic of the log chromaticity coordinates is relied upon as a basis to identify regions of an image of a single material or reflectance, such as, for example, Type B tokens.

[0078] Figure 10 is a flow chart for clustering the log chromaticity coordinates, according to a feature of the present invention. In step 240, the list of augmented log chromaticity coordinates is input the CPU 12. In step 242, the CPU 12 operates to cluster the log chromaticity coordinates. According to the teachings of the related invention, the clustering step can be implemented via, for example, a known k-means clustering. Any known clustering technique can be used to cluster the log chromaticity coordinates to determine groups of similar log chromaticity coordinate values, according to the related invention. According to the teachings of each of the related invention and the present invention, the CPU 12 correlates each log chromaticity coordinate to the group to which the respective coordinate belongs.

[0079] According to a feature of the present invention, the clustering step 242 is implemented as a function of an index of the type used in database management, for example, a hash index, a spatial hash index, b-trees or any other known index commonly used in a database management system. By implementing the clustering step 242 as a function of an index, the number of comparisons required to identify a cluster group for each pixel or token of an image is minimized. Accordingly, the clustering step can be executed by the CPU 12 in a minimum amount of time, to expedite the entire image process.

[0080] Figure 10a is an illustration of a grid for a spatial hash, according to a feature of an exemplary embodiment of the present invention. As shown in figure 10a, a spatial hash divides an image being processed into a grid of buckets, each bucket being dimensioned to be spatialThresh x spatialThresh. The grid represents a histogram of the u,v log chromaticity values for the cluster groups. As each cluster is created, a reference to the cluster is placed in the appropriate bucket of the grid.

[0081] Each new pixel or token of the image being processed is placed in the grid, in the bucket it would occupy, as if the item (pixel or token) was a new group in the clustering process. The pixel or token is then examined relative to the clusters in, for example, a 3 x 3 grid of buckets surrounding the bucket occupied by the item being examined. The item is added to the cluster group within the 3 x 3 gird, for example, if the item is within a threshold for a clusterMean.

[0082] The CPU 12 also operates to calculate a center for each group identified in the clustering step. For example, the CPU 12 can determine a center for each group relative to a (log R, log G, log B, log T) space.

[0083] In step 244, the CPU 12 outputs a list of the cluster group memberships for the log chromaticity coordinates (cross referenced to either the corresponding pixels or Type C tokens) and/or a list of cluster group centers.

[0084] Pursuant to a further feature of the present invention, the list of cluster group memberships can be augmented with a user input of image characteristics. For example, a user can specify pixels or regions of the image that are of the same material reflectance. The CPU 12 operates to overlay the user specified pixels or regions of same reflectance onto the clustering group membership information.

[0085] As noted above, in the execution of the clustering method, the CPU 12 can use the list of colors from either the list generated through execution of step 202 of the routine of figure 6, or the list generated through execution of step 204. In applying the identified cluster groups to an input image, the CPU 12 can be operated to use the same set of colors as used in the clustering method (one of the list of colors corresponding to step 202 or to the list of colors corresponding to step 204), or apply a different set of colors (the other of the list of colors corresponding to step 202 or the list of colors corresponding to step 204). If a different set of colors is used, the CPU 12 proceeds to execute the routine of figure 11.

[0086] Figure 11 is a flow chart for assigning the log chromaticity coordinates to clusters determined through execution of the routine of figure 10, when a different list of colors is used after the identification of the cluster groups, according to a feature of the present invention. In step 250, the CPU 12 once again executes the routine of figure 8, this time in respect to the new list of colors. For example, if the list of colors generated in step 202 (colors based upon Type C tokens) was used to identify the cluster groups, and the CPU 12 then operates to classify log chromaticity coordinates relative to cluster groups based upon the list of colors generated in step 204 (colors based upon pixels), step 250 of the routine of figure 11 is executed to determine the log chromaticity coordinates for the colors of the pixels in the input image file 18.

[0087] In step 252, the list of cluster centers is input to the CPU 12. In step 254, the CPU 12 operates to classify each of the log chromaticity coordinates identified in step 250, according to the nearest cluster group center. In step 256, the CPU 12 outputs a list of the cluster group memberships for the log chromaticity coordinates based upon the new list of colors, with a cross reference to either corresponding pixels or Type C tokens, depending upon the list of colors used in step 250 (the list of colors generated in step 202 or the list of colors generated in step 204).

[0088] Figure 12 is a flow chart for detecting regions of uniform reflectance based on the log chromaticity clustering according to a feature of the present invention. In step 260, the input image file 18 is once again provided to the CPU 12. In step 262, one of the pixels or Type C tokens, depending upon the list of colors used in step 250, is input to the CPU 12. In step 264, the cluster membership information, form either steps 244 or 256, is input to the CPU 12.

[0089] In step 266, the CPU 12 operates to merge each of the pixels, or specified regions of an input image, such as, for example, Type C tokens, having a same cluster group membership into a single region of the image to represent a region of uniform reflectance (Type B token). The CPU 12 performs such a merge operation for all of the pixels or tokens, as the case may be, for the input image file 18. In step 268, the CPU 12 outputs a list of all regions of uniform reflectance (and also of similar tone mapping intensities and x, y coordinates, if the log chromaticity coordinates were augmented in steps 234 and/or 236). It should be noted that each region of uniform reflectance (Type B token) determined according to the features of the present invention, potentially has significant illumination variation across the region.

[0090] U. S. Patent No. 8,139,867 teaches a constraint/solver model for segregating illumination and material in an image, including an optimized solution based upon a same material constraint. A same material constraint, as taught in U. S. Patent No. 8,139,867, utilizes Type C tokens and Type B tokens, as can be determined according to the teachings of the present invention. The constraining relationship is that all Type C tokens that are part of the same Type B token are constrained to be of the same material. This constraint enforces the definition of a Type B token, that is, a connected image region comprising contiguous pixels that represent a region of the image encompassing a single material in the scene, though not necessarily the maximal region corresponding to that material. Thus, all Type C tokens that lie within the same Type B token are by the definition imposed upon Type B tokens, of the same material, though not necessarily of the same illumination. The Type C tokens are therefore constrained to correspond to observed differences in appearance that are caused by varying illumination.

[0091] Figure 13 is a representation of an [A] [x] = [b] matrix relationship used to identify and separate illumination and material aspects of an image, according to a same-material constraint, as taught in U. S. Patent No. 8,139,867. Based upon the basic equation I = ML (I = the recorded image value, as stored in an image file 18, M =

material reflectance, and L = illumination), log(I) = log (ML) = log (M) + log(L). This can be restated as $i = m + l$, wherein i represents log(I), m represents log(M) and l represents log(L). In the constraining relationship of a same material, in an example where three Type C tokens, a, b and c, (as shown in figure 13) are within a region of single reflectance, as defined by a corresponding Type B token defined by a, b and c, then $m_a = m_b = m_c$. For the purpose of this example, the I value for each Type C token is the average color value for the recorded color values of the constituent pixels of the token. The a, b and c, Type C tokens of the example can correspond to the blue Type B token illustrated in figure 3d.

[0092] Since: $m_a = i_a - l_a$, $m_b = i_b - l_b$, and $m_c = i_c - l_c$, these mathematical relationships can be expressed , in a same material constraint, as $(1)l_a + (-1)l_b + (0)l_c = (i_a - i_b)$, $(1)l_a + (0)l_b + (-1)l_c = (i_a - i_c)$ and $(0)l_a + (1)l_b + (-1)l_c = (i_b - i_c)$.

[0093] Thus, in the matrix equation of figure 13, the various values for the log (I) (i_a, i_b, i_c), in the [b] matrix, are known from the average recorded pixel color values for the constituent pixels of the adjacent Type C tokens a, b and c. The [A] matrix of 0's, 1's and -1's, is defined by the set of equations expressing the same material constraint, as described above. The number of rows in the [A] matrix, from top to bottom, corresponds to the number of actual constraints imposed on the tokens, in this case three, the same material constraint between the three adjacent Type C tokens a, b and c. The number of columns in the [A] matrix, from left to right, corresponds to the number of unknowns to be solved for, again, in this case, the three illumination values for the three tokens. Therefore, the values for the illumination components of each Type C token a, b and c, in the [x] matrix, can be solved for in the matrix equation, by the CPU 12. It should be noted that each value is either a vector of three values corresponding to the color bands (such as red, green, and blue) of our example or can be a single value, such as in a grayscale image.

[0094] Once the illumination values are known, the material color can be calculated by the CPU 12 using the I = ML equation. Intrinsic illumination and material images can be now be generated for the region defined by tokens *a*, *b* and *c*, by replacing each pixel in the original image by the calculated illumination values and material values, respectively. An example of an illumination image and material image, corresponding to the original image shown in figure 3b, is illustrated in figure 14.

[0095] Implementation of the constraint/solver model according to the techniques and teachings of U. S. Patent No. 8,139,867, utilizing the Type C tokens and Type B tokens obtained via a log chromaticity clustering technique according to the present invention, provides a highly effective and efficient method for generating intrinsic images corresponding to an original input image. The intrinsic images can be used to enhance the accuracy and efficiency of image processing, image analysis and computer vision applications implemented, for example, in the computer system 10 of the automobile 1.

[0096] However, the intrinsic images generated from the performance of the exemplary embodiments of the present invention can include artifacts that distort the appearance of a scene depicted in the image being processed. The artifacts can be introduced through execution of the intrinsic image generations methods of the present invention, or through user modifications such as the user input of image characteristics discussed above. Accordingly, according to a feature of the present invention, various post processing techniques can be implemented to reduce the artifacts.

[0097] Figure 15 is a flow chart for an edge preserving blur post processing technique applied to the intrinsic images illustrated in figure 14, according to a feature of the present invention, to improve the quality of the illumination and material reflectance aspects depicted in the intrinsic images. In step 300, the CPU 12 receives as an input an original image (an image file 18), and the corresponding intrinsic material reflectance and illumination images determined by the CPU 12 through solution of the matrix equation shown in figure 13, as described above.

[0098] In step 302, the CPU 12 operates to perform an edge-preserving blur of the illumination in the illumination image by applying an edge preserving smoothing filter. The edge preserving smoothing filter can be any one of the known filters such as, for example, a bilateral filter, a guided filter, a mean-shift filter, a median filter, anisotropic diffusion and so on. The filter can be applied one or more times to the illumination image. In an exemplary embodiment, a bilateral filter is applied to the illumination image twice. In addition, several different types of filters can be applied in succession, for example, a median filter followed by a bilateral filter.

[0099] In step 304, the CPU 12 recalculates the intrinsic material reflectance image based upon the I = ML equation, and using the original image of the image file 18 and the illumination image, as modified in step 302. In step 306, the CPU 12 outputs intrinsic material reflectance and illumination images, as modified by the CPU 12 through execution of the routine of figure 15.

[0100] A smoothing filter applied to the illumination image results in several improvements to the appearance of the intrinsic images when used in, for example, such applications as computer graphics. For example, in computer graphics, texture mapping is used to achieve certain special effects. Artists consider it desirable in the performance of texture mapping to have some fine scale texture form the illumination in the material reflectance image. By smoothing the illumination image, in step 302, the fine scale texture is moved to the material reflectance image upon a recalculation of the material image in step 304, as will be described below.

[0101] In addition, smoothing the illumination in step 302 places some of the shading illumination (illumination intensity variation due to curvature of a surface) back into the material reflectance image, giving the material image some expression of curvature.

That results in an improved material depiction more suitable for artistic rendering in a computer graphics application.

[0102] Moreover, small reflectance variation sometimes erroneously ends up in the illumination image. The smoothing in step 302 forces the reflectance variation back into the material image.

[0103] Figure 16 is a flow chart for an artifact reduction post processing technique applied to the intrinsic images illustrated in figure 14, according to a feature of the present invention, to improve the quality of the illumination and material reflectance aspects depicted in the intrinsic images. In step 400, the CPU 12 receives as an input an original image (an image file 18), and the corresponding intrinsic material reflectance and illumination images determined by the CPU 12 through solution of the matrix equation shown in figure 13, as described above. Optionally, the intrinsic images can be previously modified by the CPU 12 through execution of the routine of figure 15.

[0104] In step 402, the CPU 12 operates to calculate derivatives (the differences between adjacent pixels) for the pixels of each of the original image and the material reflectance image. Variations between adjacent pixels, in the horizontal and vertical directions, are caused by varying illumination and different materials in the scene depicted in the original image. When the CPU 12 operates to factor the original image into intrinsic illumination and material reflectance images, some of the variation ends up in the illumination image and some ends up in the material reflectance image. Ideally, all of the variation in the illumination image is attributable to varying illumination, and all of the variation in the material reflectance image is attributable to different materials.

[0105] Thus, by removing the illumination variation, variations in the material reflectance image should be strictly less than variations in the original image.

However, inaccuracies in the process for generating the intrinsic images can result in new edges appearing in the material reflectance image.

[0106] In step 404, the CPU 12 operates to identify the artifacts caused by the newly appearing edges by comparing the derivatives for the material reflectance image with the derivatives for the original image. The CPU 12 modifies the derivatives in the material reflectance image such that, for each derivative of the material reflectance image, the sign is preserved, but the magnitude is set at the minimum of the magnitude of the derivative in the original image and the material reflectance image. The modification can be expressed by the following equation:
derivativeReflectanceNew = min(abs(derivativeReflectanceOld), abs(derivativeOriginalimage)) * sign(derivativeReflectanceOld)

[0107] In step 406, the CPU integrates the modified derivatives to calculate a new material reflectance image. The new image is a material reflectance image without the newly appearing, artifact-causing edges. Any known technique can be implemented to perform the integration. For example, the CPU 12 can operate to perform numerical 2D integration by solving the 2D Poisson equation using discrete cosine transforms.

[0108] In step 408, the CPU 12 recalculates the intrinsic illumination image based upon the I = ML equation, and using the original image of the image file 18 and the material reflectance image, as modified in steps 404 and 406. In step 408, the CPU 12 outputs intrinsic material reflectance and illumination images, as modified by the CPU 12 through execution of the routine of figure 16.

[0109] Figure 17 is a flow chart for a BIDR model enforcement post processing technique applied to the intrinsic images illustrated in figure 14, according to a feature of the present invention, to improve the quality of the illumination and material reflectance aspects depicted in the intrinsic images.

[0110] As described above, the BIDR model predicts the correct color for a material, in a shadow penumbra, from full shadow to fully lit. As shown in figure 18, according to the prediction of the BIDR model, colors for a material, for example, in an RGB color space, from a fully shaded color value to a fully lit color value, generally form a line in the color space. In full shadow, the material is illuminated by an ambient illuminant, while when fully lit, the material is illuminated by the ambient illuminant and the direct or incident illuminant present in the scene at the time the digital image of an image file 18 was recorded.

[0111] According to the BIDR model, the illumination values in an image also define a line extending from the color of the ambient illuminant to the color of the combined ambient and direct illuminants. In log color space, the illumination line predicted by the BIDR model corresponds to the normal, N of the log color space chromaticity plane illustrated in figure 5.

[0112] Various inaccuracies in the generation of the illumination and material intrinsic images, as described above, can also result, for example, in illumination values in the generated intrinsic illumination image that diverge from the line for the illumination values predicted by the BIDR model. According to the present invention, the illumination line prediction of the BIDR model is used to correct such inaccuracies by modifying the illumination to be linear in log(RGB) space.

[0113] Referring once again to figure 17, in step 500, the CPU 12 receives as input a BIDR illumination orientation, corresponding to the normal N illustrated in figure 5. In the exemplary embodiment of the present invention, N is determined by the CPU 12 through execution of the routine of figure 7, as described above. In that case, the N determined through execution of the routine of figure 7 is used in both the clustering process described above, and in the BIDR model enforcement post processing technique illustrated in figure 17.

[0114] In the event the illumination and material reflectance images are generated via a method different from the log chromaticity clustering technique of the exemplary embodiment, the orientation N is determined by the CPU 12 in a separate step before the execution of the routine of figure 17, through execution of the routine of figure 7. When N is determined in a separate step, the CPU 12 can operate relative to either the original image or the illumination image. In addition, when the processing is based upon a user input, as described above, the user can make a selection from either the original image or the illumination image.

[0115] Moreover, in step 500, the CPU 12 also receives as input an original image (an image file 18), and the corresponding intrinsic material reflectance and illumination images determined by the CPU 12 through solution of the matrix equation shown in figure 13, also as described above. Optionally, the intrinsic images can be previously modified by the CPU 12 through execution of the routine(s) of either one, or both figures 15 and 16.

[0116] In step 502, the CPU 12 determines the full illumination color in the illumination image. The full illumination color (ambient + direct) can be the brightest color value depicted in the illumination image. However, the brightest value can be inaccurate due to noise in the image or other outliers. In a preferred exemplary embodiment of the present invention, a more accurate determination is made by finding all illumination color values in a preselected range of percentiles of the intensities, for example, the 87^{th} through 92^{nd} percentiles, and calculating an average of those values. The average is used as the full illumination color value. Such an approach provides a robust estimate of the bright end of the illumination variation in the intrinsic illumination image.

[0117] In step 504, the CPU 12 operates to modify all of the pixels of the illumination image by projecting all of the illumination colors depicted by the pixels in the illumination image to the nearest point on a line having the orientation N (input to the

CPU 12 in step 500) and passing through the full illumination color determined in step 302. Thus, the color of each pixel of the illumination image is modified to conform to the closest value required by the BIDR model prediction.

[0118] A special case exists for the pixels of the illumination image having an intensity that is greater than the full illumination color value, as calculated in step 502. The special case can be handled by the CPU 12 according to a number of different methods. In a first method, the modification is completed as with all the other pixels, by projecting each high intensity pixel to the nearest value on the illumination line. In a second method, each high intensity pixel is replaced by a pixel set at the full illumination color value. According to a third method, each high intensity pixel is kept at the color value as in the original image.

[0119] An additional method is implemented by using a weighted average for each high intensity pixel, of values determined according to the first and third methods, or of values determined according to the second and third methods. The weights would favor values calculated according to either the first or second methods when the values are similar to high intensity pixels that are not significantly brighter than the full illumination color value calculated in step 502. Values calculated via the third method are favored when values for high intensity pixels that are significantly brighter than the full illumination color value. Such a weighting scheme is useful when the I = ML equation for image characteristics is inaccurate, for example, in the presence of specular reflections.

[0120] Any known technique can be implemented to determine the relative weights for illumination values. In an exemplary embodiment of the present invention, a sigmoid function is constructed such that all the weight is on the value determined either according to the first or second methods, when the intensity of the high intensity pixel is at or near the full illumination color value, with a smooth transition to an equally weighted value, between a value determined either according to the first or second methods and a value determined according to the third method, as the intensity increases. That is followed by a further smooth transition to full weight on a value determined according to the third method, as the intensity increase significantly beyond the full illumination color value.

[0121] In step 506, the CPU 12 recalculates the intrinsic material reflectance image based upon the I = ML equation, and using the original image of the image file 18 and the illumination image, as modified in step 504. In step 508, the CPU 12 outputs intrinsic material reflectance and illumination images modified to strictly adhere to the predictions of the BIDR model.

[0122] For best results, the above-described post processing techniques can be executed in a log(RGB) space. Also, the various techniques can be executed in the order described above. Once one or more of the post processing techniques have been executed, the final modified material reflectance and illumination intrinsic images can be white balanced and/or scaled, as desired, and output by the CPU 12.

[0123] According to a further exemplary embodiment of the present invention, an approximate intrinsic estimate representation of an image is used as the intrinsic image for use by the computer system 10 of the automobile 1. The approximate intrinsic estimate representation according to a feature of the present invention utilizes illumination invariant log chromaticity plane coordinates and log color space distances to provide an illumination invariant image that represents an estimate of the intrinsic material aspects of an image for use in identifying road features such as lane markings.

[0124] As shown in figure 5, a log color value for each pixel in, for example, material C, is projected to an a, b coordinate on the u, v log chromaticity plane. Each pixel can also be designated by a distance T from the log chromaticity plane coordinates a, b for the respective pixel on the log chromaticity plane, to the log color space value for the respective pixel, as shown in the example of figure 5. Thus, each pixel of an image can be mapped to a corresponding Tab value.

[0125] To that end, the CPU 12 initially performs the routines of figures 6-8 to determine a list of log chromaticity coordinates a, b, for the colors of the pixels of the input image. The CPU 12 co-relates the a, b coordinate values to the respective pixels. The CPU 12 then determines a distance T for each pixel by calculating the log value of the color of each pixel, and then determining a signed Euclidian distance in the log color space for each pixel, from the log color value to the log chromaticity plane, as shown in the example of figure 5.

[0126] As a result of the a, b coordinate and T distance processing, the CPU 12 can map each pixel of the input image to corresponding T, ab values, to provide the approximate intrinsic material estimate representation of the input image.

[0127] Referring once again to figures 1a & b, the CPU 12 can store either the intrinsic material reflectance image or the Tab image for further processing, depending upon the embodiment of the present invention implemented in the computer system 10 of the automobile 1. The CPU 12 can then be operated to execute a known object recognition engine, face recognition engine and/or optical character recognition engine (OCR) relative to the intrinsic representation of the image. The outputs of the various recognition engines provide an identification of unique features in the scene, including objects such as, for example, lane markings and the text of street signs, the face of the driver, and road conditions such as, for example, surface reflectance properties (wet or dry) and material type (concrete, asphalt, etc.). The use of intrinsic representations of images of a scene enables a more accurate and precise analysis of image features by the selected engine, to identify the unique features, without the complexity in image appearance that can be caused by varying illumination present at the time the scene was recorded (for example, shadows).

[0128] For example, in an exemplary embodiment of the present invention, the intrinsic representation is applied as an input to a Mobileye brand automotive computer vision product. The use of an intrinsic representation enhances the functionality of the product in such applications as lane detection, radar vision fusion, pedestrian detection, head lamp control, surround view systems, vehicle detection, general object detection, traffic sign detection and rear camera applications.

[0129] In a further exemplary embodiment of the present invention, the intrinsic representation of images from a camera focused on the driver is applied as an input to a face recognition engine. The face recognition engine is operated to perform such functions as recognizing the identity of the driver, and analyzing driver appearance for indications of such conditions as drowsiness, intoxication and so on.

[0130] According to a further feature of the present invention, a segmentation of the intrinsic image based upon the identification of unique features in the scene is performed by the CPU 12 to assign a unique identification (ID) value to each image region corresponding to a uniquely identified object, as for example, road marking (for example, lane markings), curb, vehicle, tree, and/or road features and conditions, such as wet surfaces and so on. Further distinctions in ID values can be based upon such object attributes as region material color, texture properties, surface reflectance properties, material type, such as asphalt, concrete, moisture condition of surfaces, and so on.

[0131] A further refinement of the segmentation can include a binary map wherein each pixel is assigned to one of two categories of ID value: road marking and not road marking.

[0132] Various driver notifications can be generated as a function of vehicle position and ID value, for example outputs provided by the Mobileye brand automotive computer vision product.

[0133] Figure 19 is a flow chart for a driver assistance method according to a feature of the present invention. In step 600 an intrinsic representation of a scene recorded by one of the cameras 14 mounted in the automobile 1 is input to the CPU 12. The CPU 12 is then operated to optionally execute one or more engines, such as an object recognition engine (step 602), a face recognition engine (step 604) and/or an OCR engine (step 606).

[0134] All of the output information provided by the various engines operated by the CPU 12 are input to an output information processing step (step 608). As noted, the use of intrinsic representations improves the quality of the information that can be generated by the various known and available recognition engines due to the removal of the complexity in image appearance that can be caused by varying illumination present at the time the scene was recorded (for example, shadows). The improved and accurate output information can be analyzed and processed using, for example, additional input information (step 610) such as, for example, automobile operation status, such as speed, GPS, radar and/or sonar information, vehicle-to-vehicle communications, weather conditions, and so on, and/or depth information calculated from the radar input or from a stereo set of scenes provided by the cameras 14.

[0135] In the preceding specification, the invention has been described with reference to specific exemplary embodiments and examples thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the claims that follow. The specification and drawings are accordingly to be regarded in an illustrative manner rather than a restrictive sense.

What is claimed is:

1. For use in a vehicle, an automated, computerized method for processing an image, comprising the steps of:
arranging a digital camera on a vehicle body;
operating the digital camera to provide an image file depicting an image of a scene related to vehicle operation, in a computer memory;
generating an intrinsic representation of the image; and
using the intrinsic representation to analyze the scene.

2. The method of claim 1 wherein the intrinsic representation comprises an intrinsic material reflectance image.

3. The method of claim 1 wherein the intrinsic representation comprises an approximate intrinsic estimate representation.

4. The method of claim 3 wherein the approximate intrinsic estimate representation comprises a Tab representation of the image, wherein T is a log color space distance and ab are coordinates in a log chromaticity plane in the log color space.

5. The method of claim 1 wherein the step of using the intrinsic representation to analyze the scene is carried out by using a recognition engine to identify unique features in the scene.

6. The method of claim 5 including the further step of segmenting the image into regions as a function of the unique features identified by the recognition engine.

7. The method of claim 6 wherein the step of segmenting the image into regions is carried out by segmenting the image into a binary designation of the scene.

8. The method of claim 7 wherein the binary designation of the scene includes a designation of one of road marking and not road marking.

9. A device comprising:
a digital camera adapted for mounting on a vehicle to record an image of a scene related to vehicle operation;
a computer system adapted to be coupled to the digital camera; and
the computer system including a memory storing an image file containing the image of the scene;
the computer system being arranged and configured to execute a routine to generate an intrinsic representation of the image, and use the intrinsic representation to analyze the scene.

10. The device of claim 9 wherein the intrinsic representation comprises an intrinsic material reflectance image.

11. The device of claim 9 wherein the intrinsic representation comprises an approximate intrinsic estimate representation.

12. The device of claim 11 wherein the approximate intrinsic estimate representation comprises a Tab representation of the image, wherein T is a log color space distance and ab are coordinates in a log chromaticity plane in the log color space.

13. The device of claim 9 wherein the step of using the intrinsic representation to analyze the scene is carried out by using a recognition engine to identify unique features in the scene.

14. The device of claim 13 including the further step of segmenting the image into regions identified as a function of the unique features identified by the recognition engine.

15. The device of claim 14 wherein the step of segmenting the image into regions is carried out by segmenting the image into a binary designation of the scene.

16. The device of claim 15 wherein the binary designation of the scene includes a designation of one of road marking and not road marking.

17. A computer program product, disposed on a non-transitory computer readable media, the product including computer executable process steps operable to control a computer to: receive an image file depicting an image of a scene related to vehicle operation, in a computer memory, generate an intrinsic representation of the image and use the intrinsic representation to analyze the scene.

18. The computer program product of claim 17 wherein the intrinsic representation comprises an intrinsic material reflectance image.

19. The computer program product of claim 17 wherein the intrinsic representation comprises an approximate intrinsic estimate representation.

20. The computer program product of claim 19 wherein the approximate intrinsic estimate representation comprises a Tab representation of the image, wherein T is a log color space distance and ab are coordinates in a log chromaticity plane in the log color space.

21. The computer program product of claim 17 wherein the process step to use the intrinsic representation to analyze the scene is carried out by using a recognition engine to identify unique features in the scene.

22. The computer program product of claim 21 including the further process step of segmenting the image into regions identified as a function of the unique features identified by the recognition engine.

23. The computer program product of claim 22 wherein the process step of segmenting the image into regions is carried out by segmenting the image into a binary designation of the scene.

24. The computer program product of claim 23 wherein the binary designation of the scene includes a designation of one of road marking and not road marking.

25. The computer program product of claim 21 wherein the unique features include road markings and road signs.

26. The computer program product of claim 21 wherein the unique features include road conditions.

27. The computer program product of claim 22 wherein the unique features include uniquely identified objects.

28. The computer program product of claim 22 wherein the unique features include road conditions.

29. The computer program product of claim 22 wherein the unique features include object attributes.

30. The computer program product of claim 21 wherein the process step of using a recognition engine to identify unique features in the scene is carried out by using a recognition engine selected from the group consisting of an object recognition engine, a face recognition engine, an optical character recognition engine and combinations thereof.

31. A computer program product, disposed on a non-transitory computer readable media, the product including computer executable process steps operable to control a

computer to: receive an image file depicting an image of a scene related to vehicle driver appearance, in a computer memory, generate an intrinsic representation of the image and use the intrinsic representation to analyze the vehicle driver appearance.

32. The computer program product of claim 31 wherein the process step to use the intrinsic representation to analyze the vehicle driver appearance is carried out to identify the driver.

33. The computer program product of claim 31 wherein the process step to use the intrinsic representation to analyze the vehicle driver appearance is carried out to identify indications of vehicle driver condition.

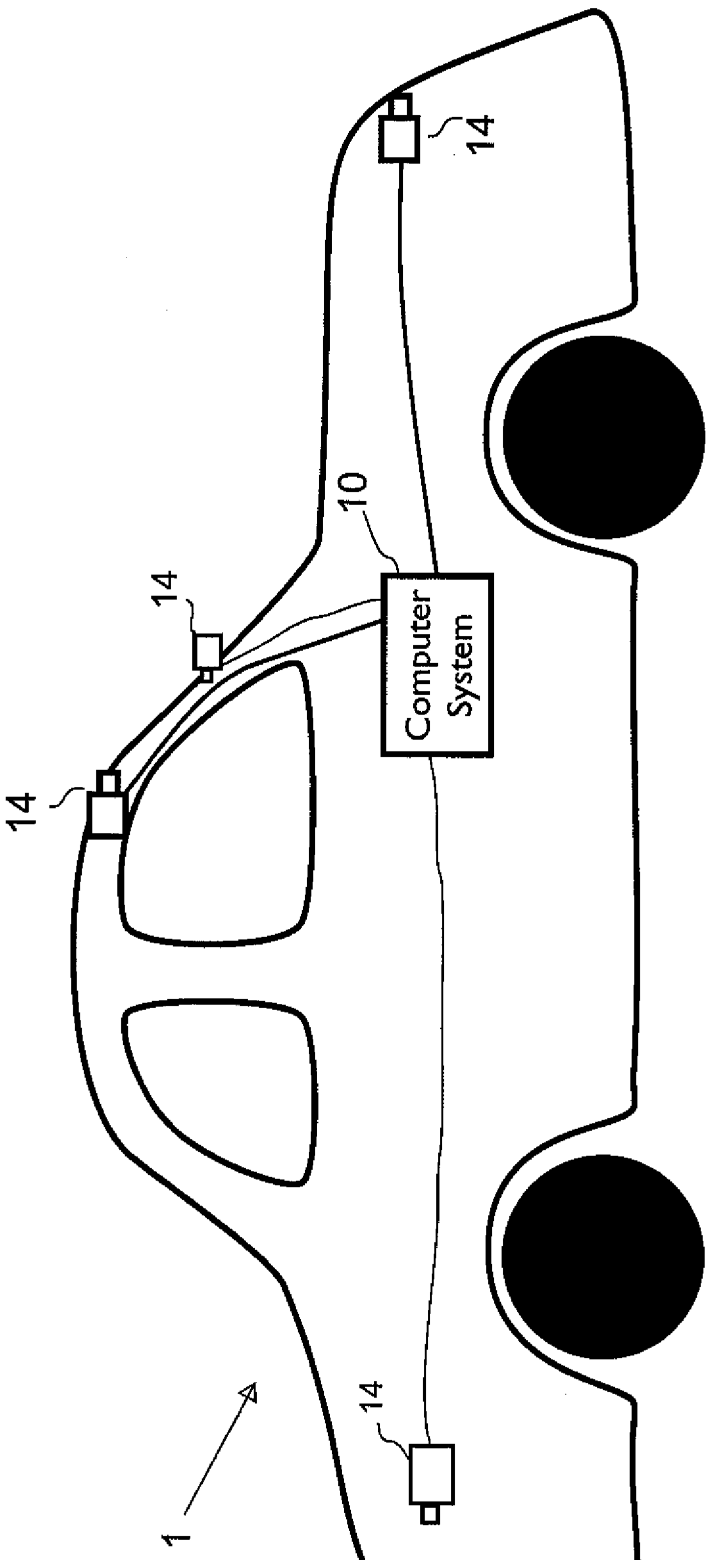


Fig 1a

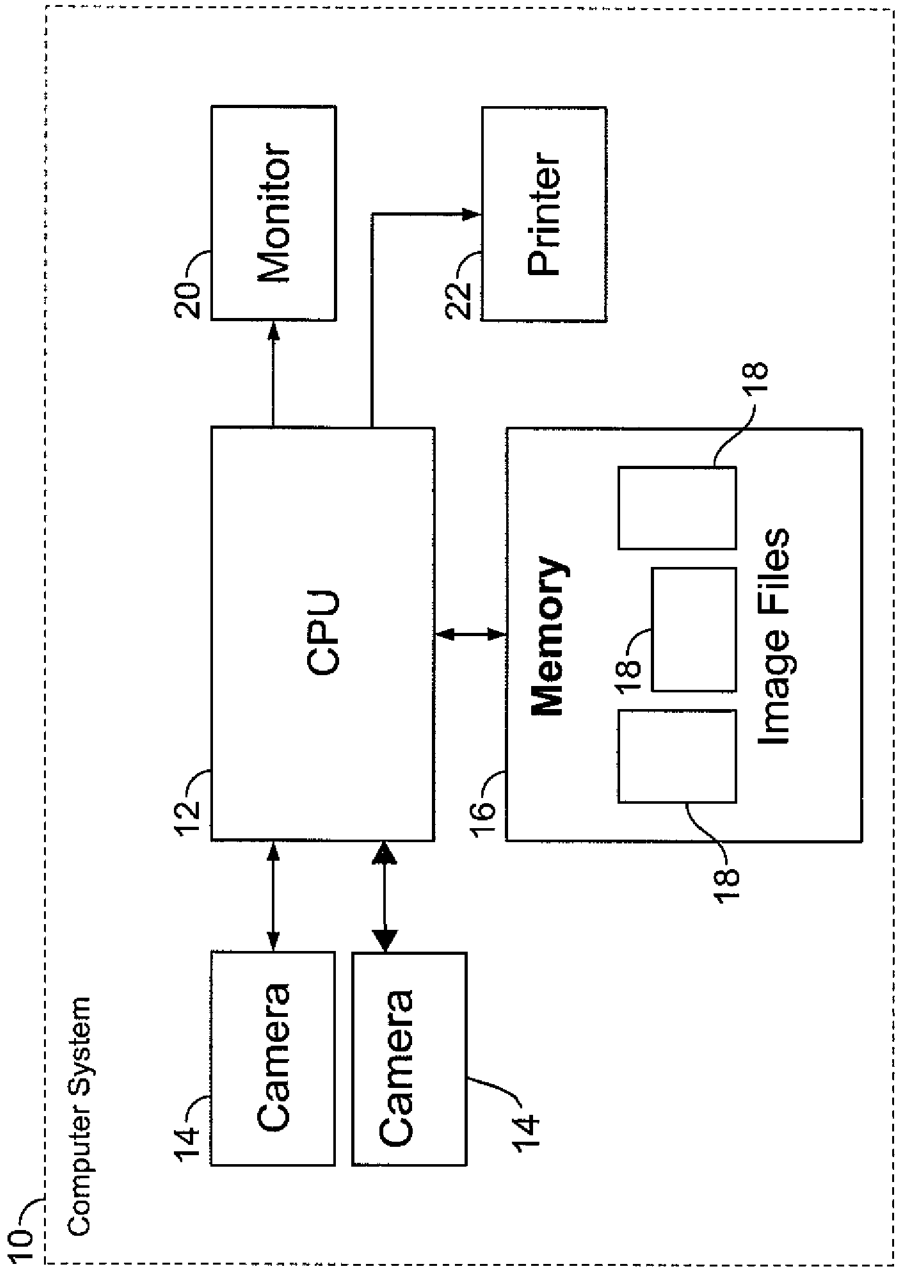

10
Computer System
14
Camera
14
Camera
12
CPU
20
Monitor
22
Printer
16
Memory
18
18
18
Image Files

Fig. 1b

Figure 2: Pixel Array for Storing Image Data

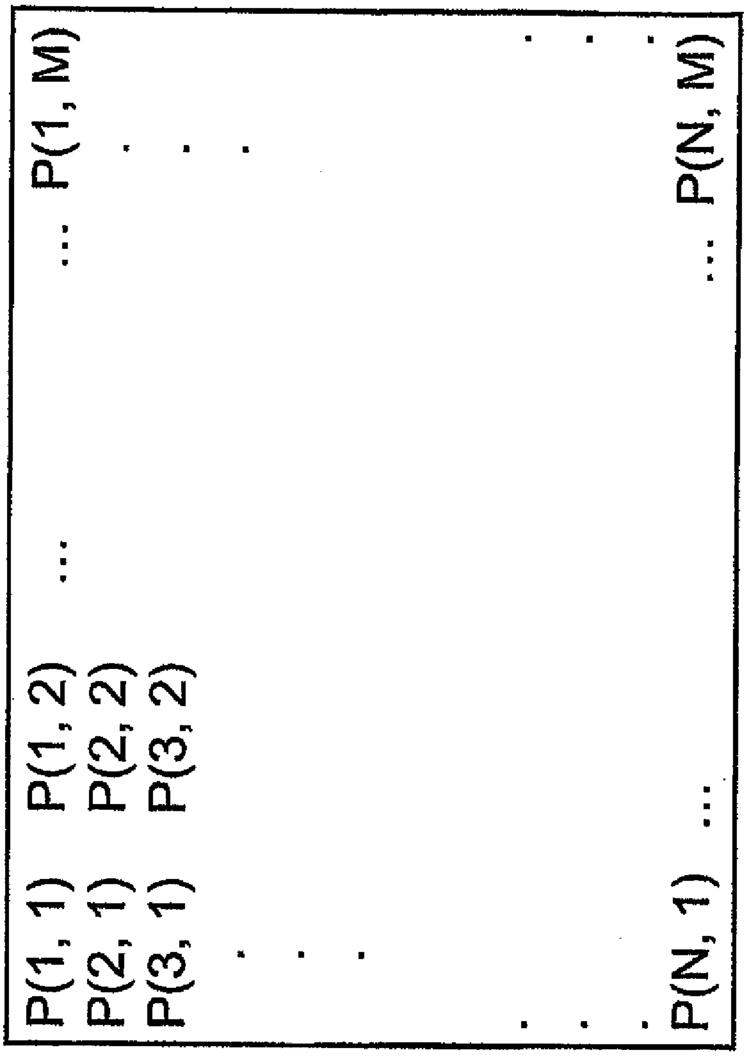


Image File 18

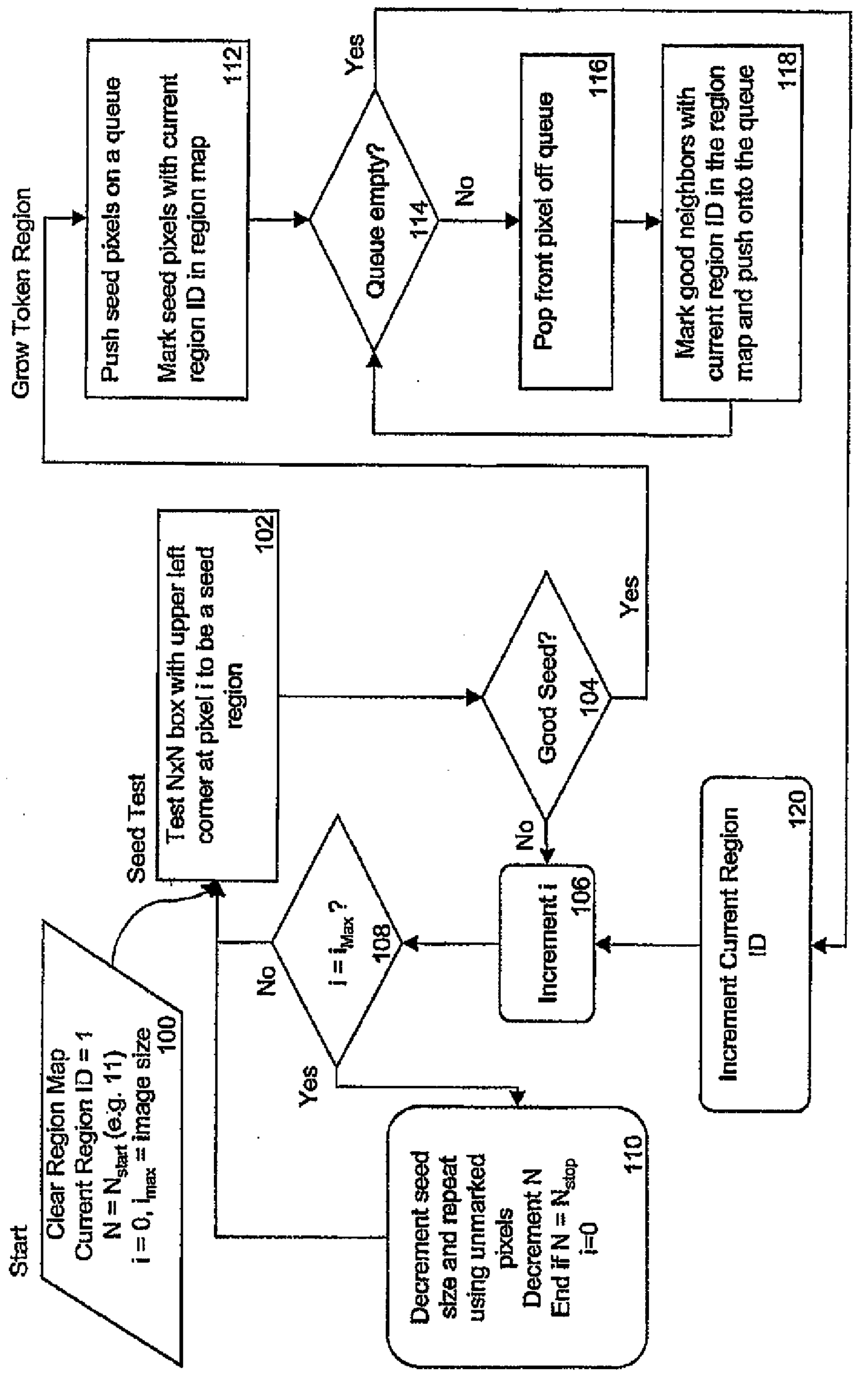


Figure 3A: Identifying Token Regions in an Image

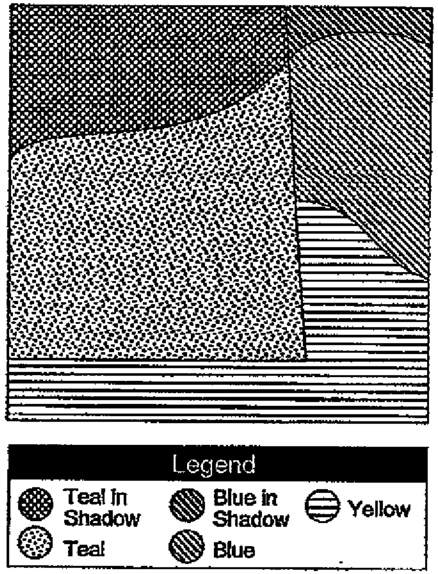


Figure 3B: Original Image

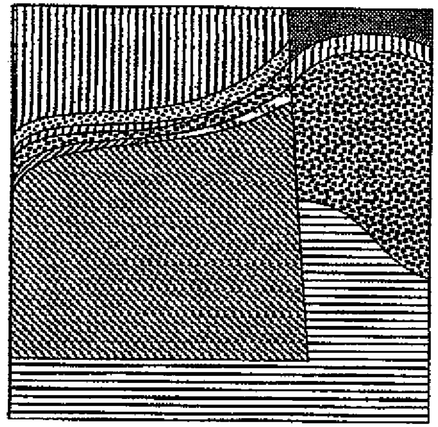


Figure 3C: Token Regions

Figure 3B, 3C: Examples of Identifying Token Regions in an Image

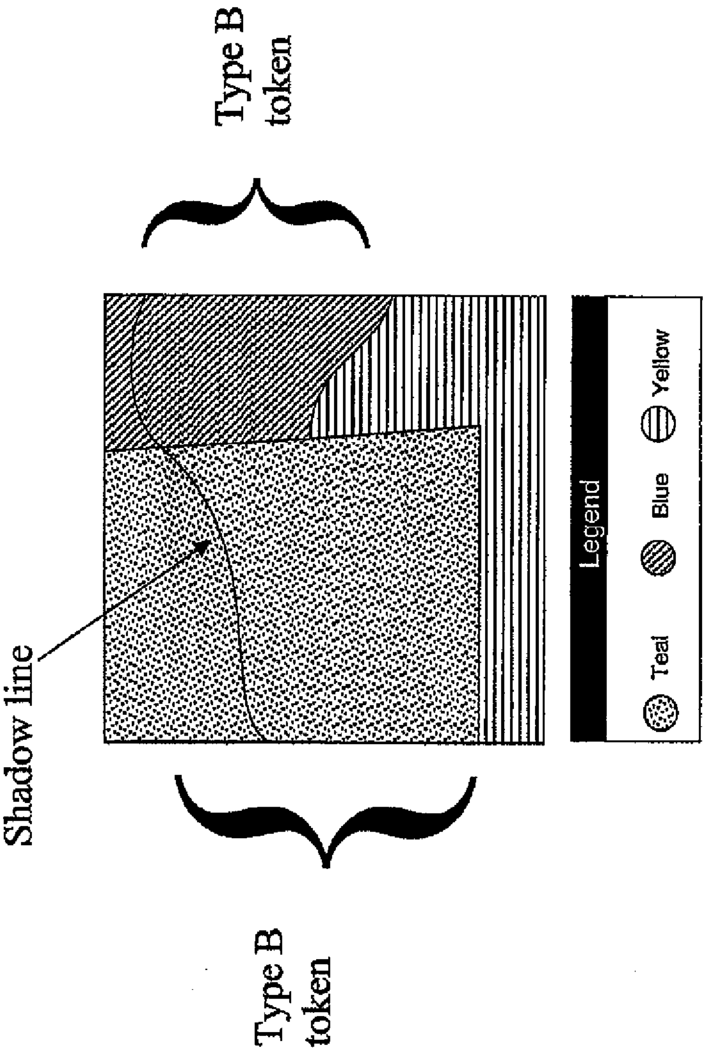


Figure 3D: Type B Tokens

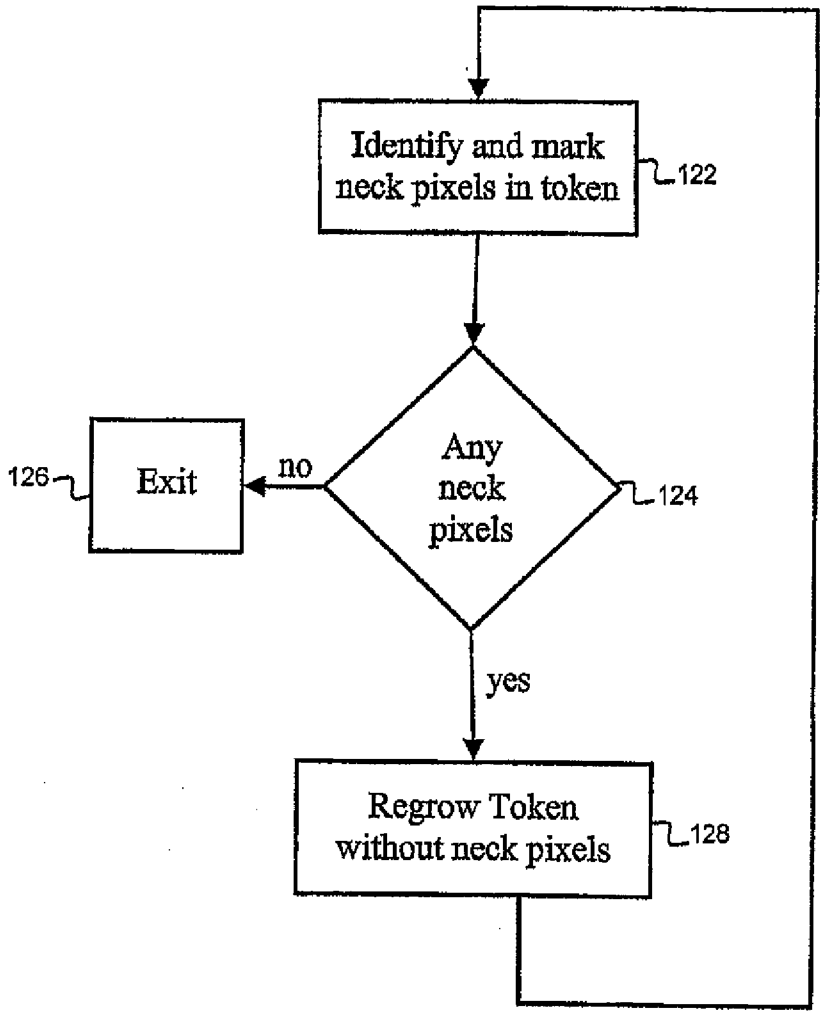
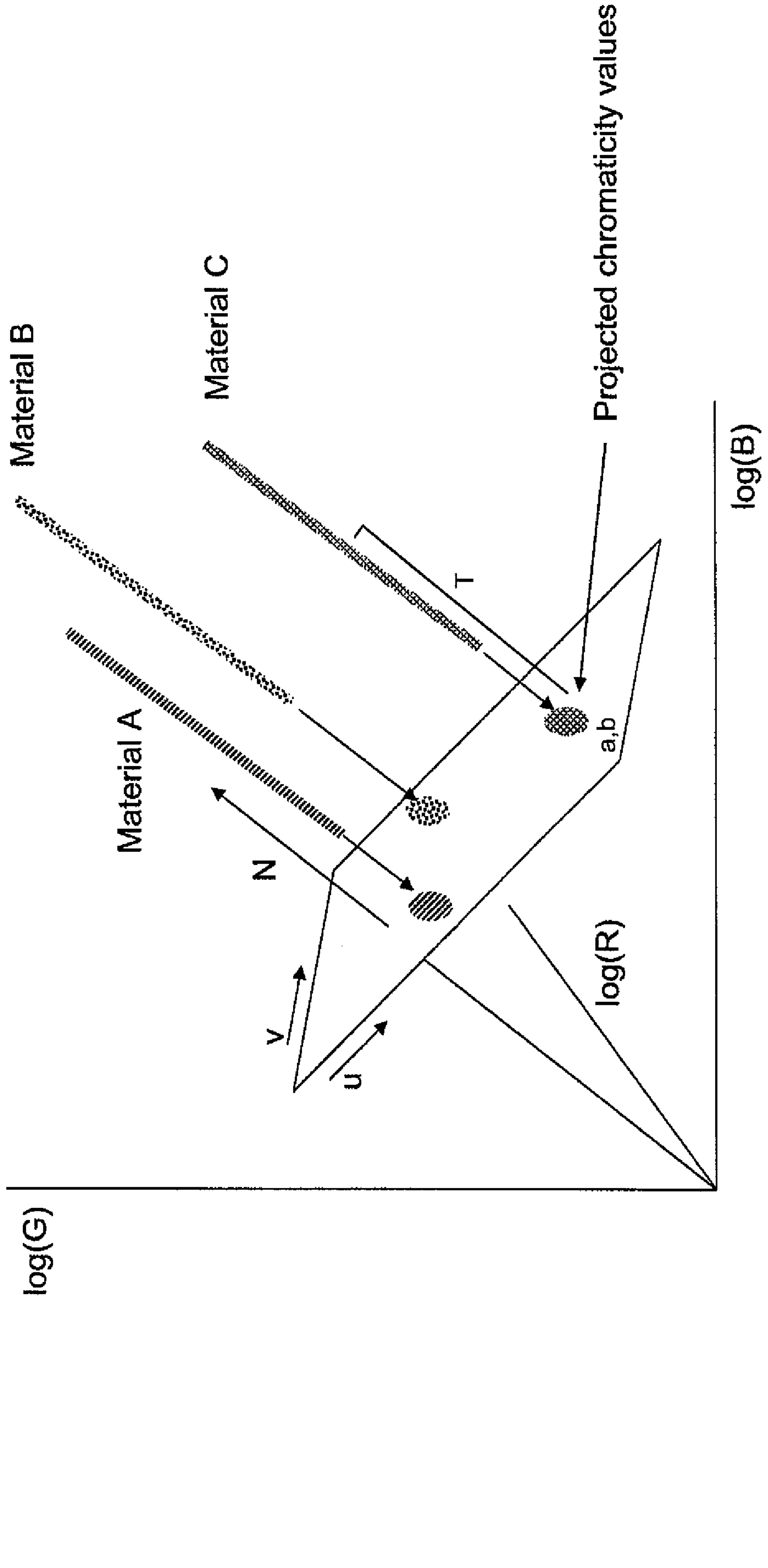

Identify and mark neck pixels in token
122
Any neck pixels
124
no
Exit
126
yes
Regrow Token without neck pixels
128

Figure 4



N = Log Space Chromaticity Plane Normal

Figure 5: Log Color Space Chromaticity Plane

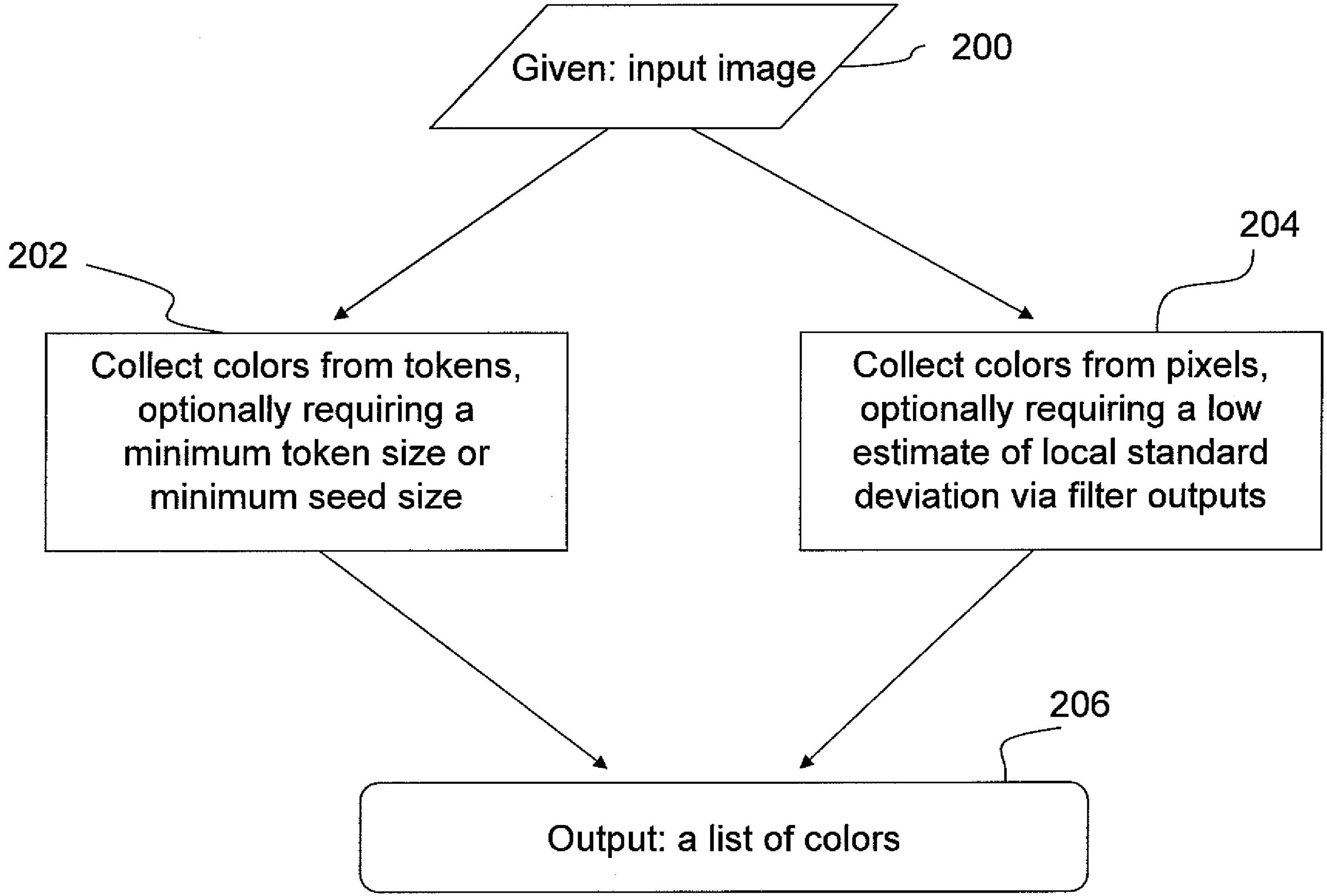


Figure 6: Selecting colors from an image

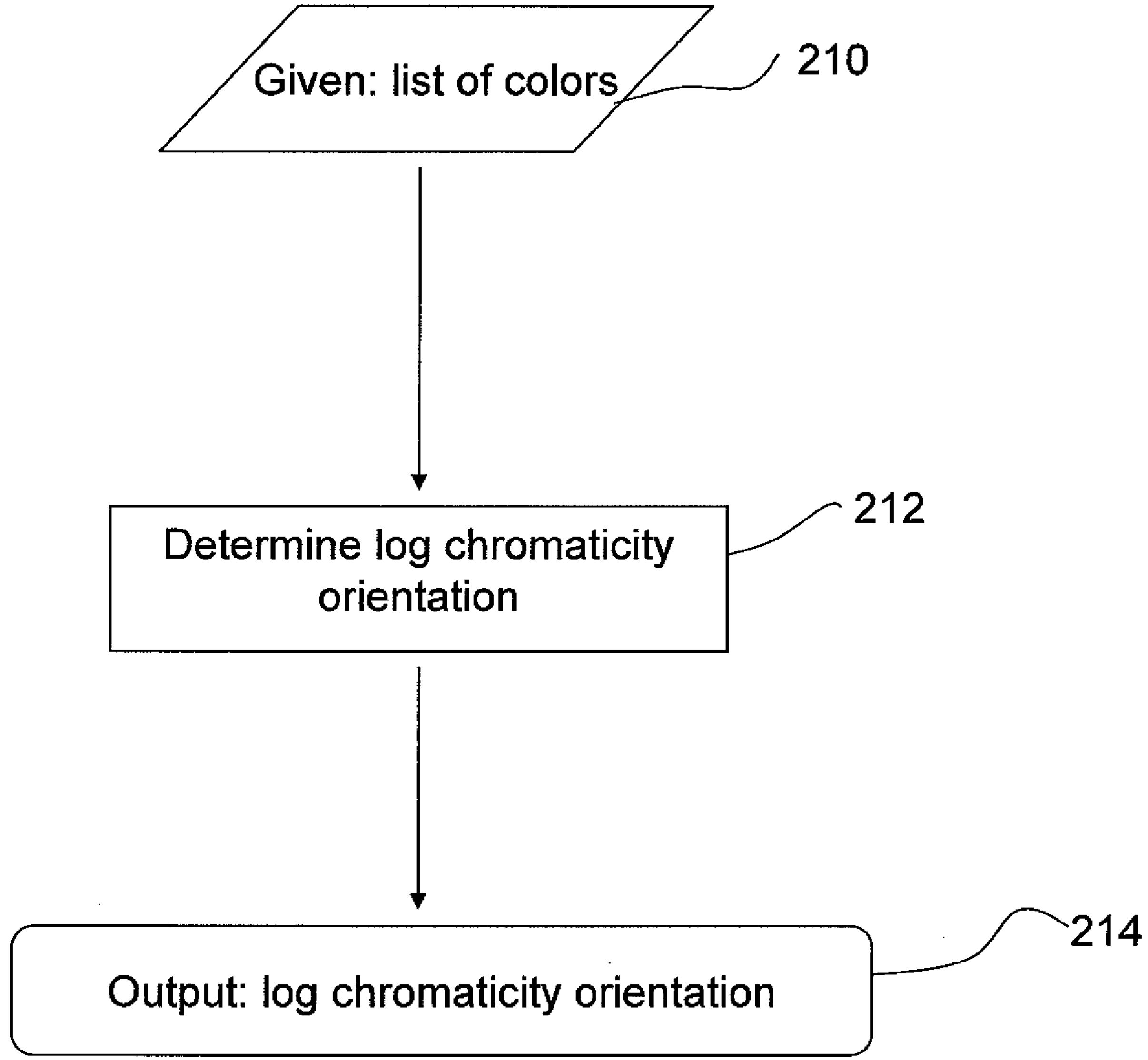


Figure 7: Determining the log chromaticity orientation

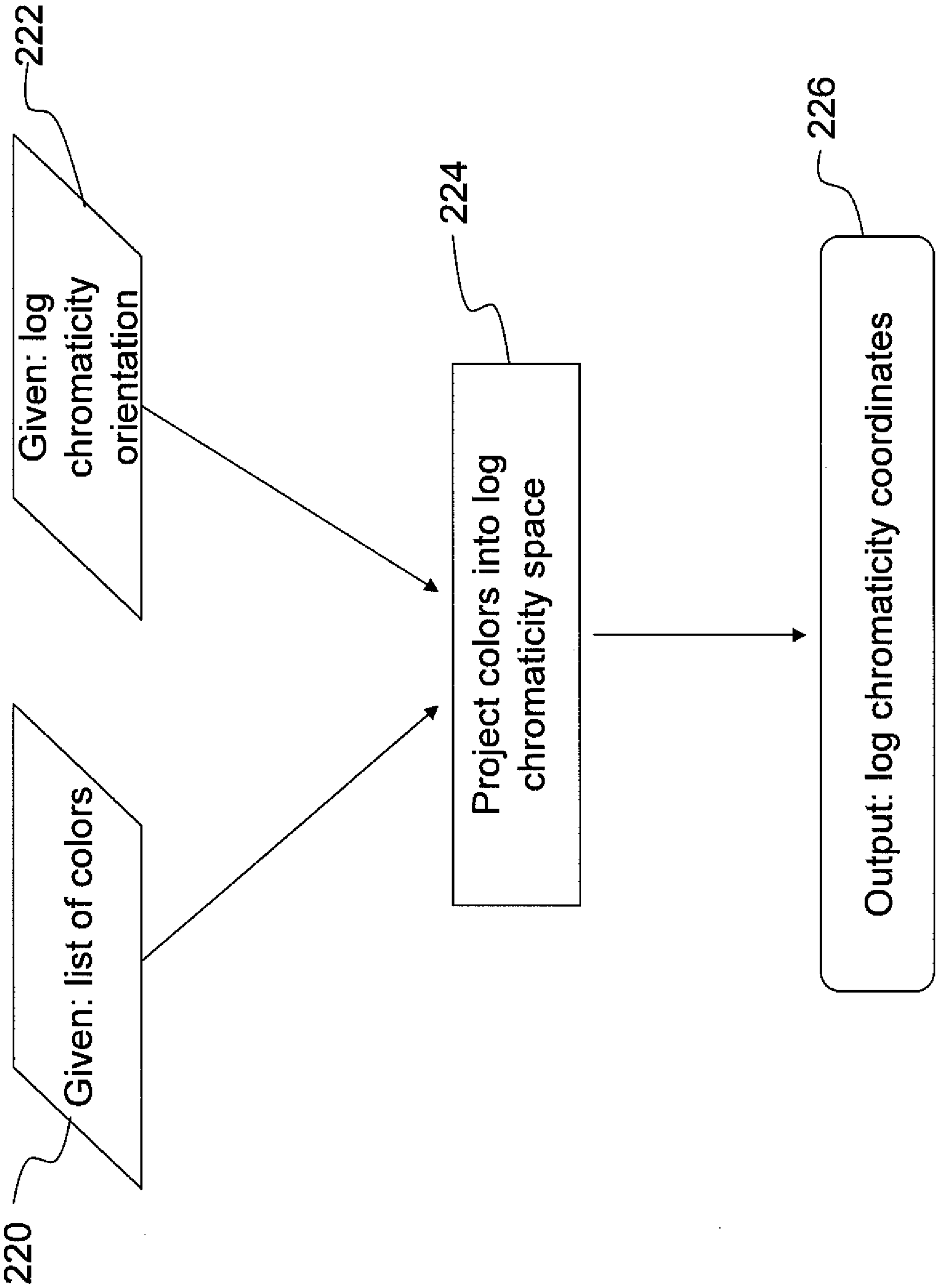


Figure 8: Determining log chromaticity coordinates

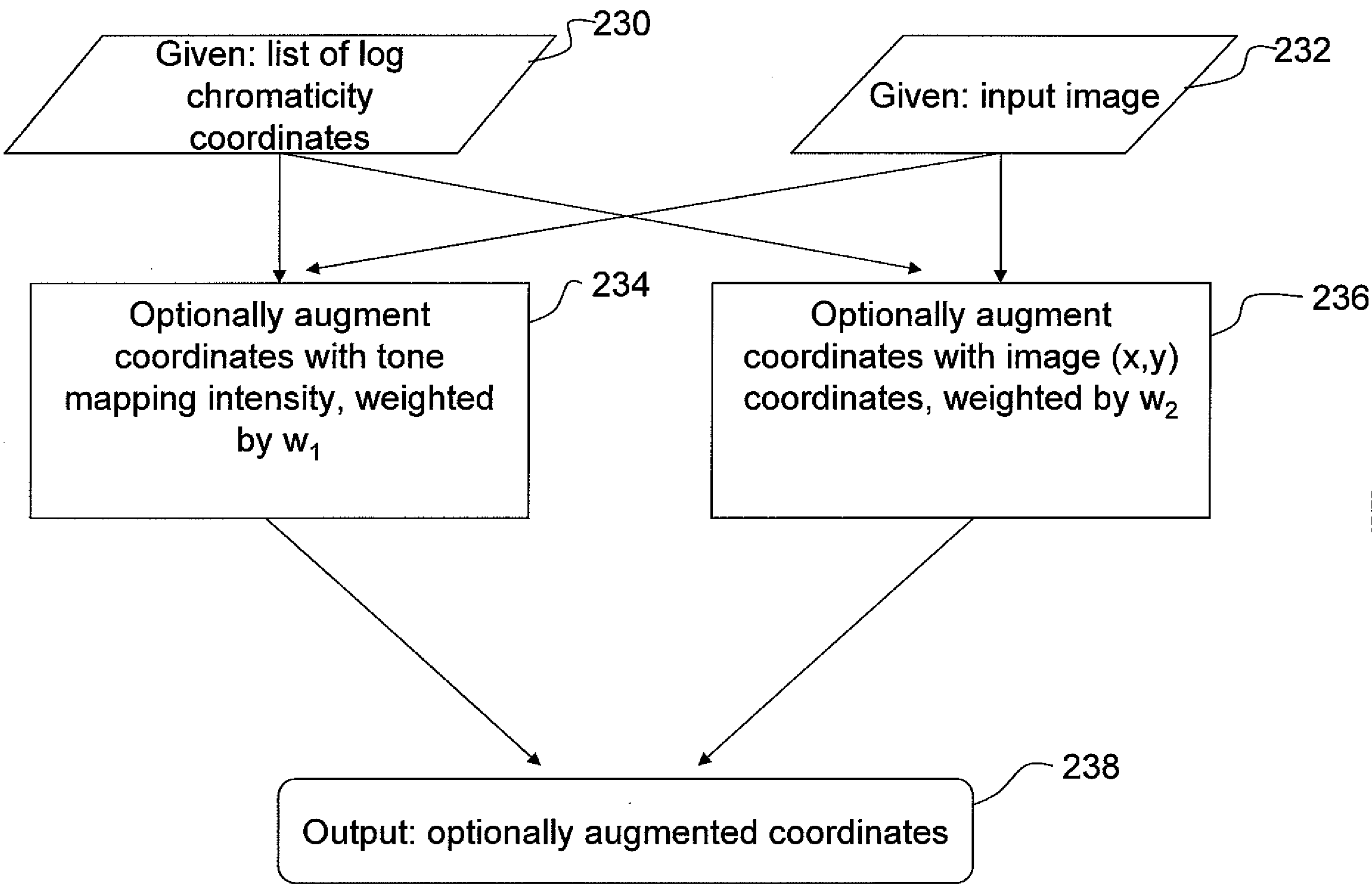


Figure 9: Optionally augmenting log chromaticity coordinates

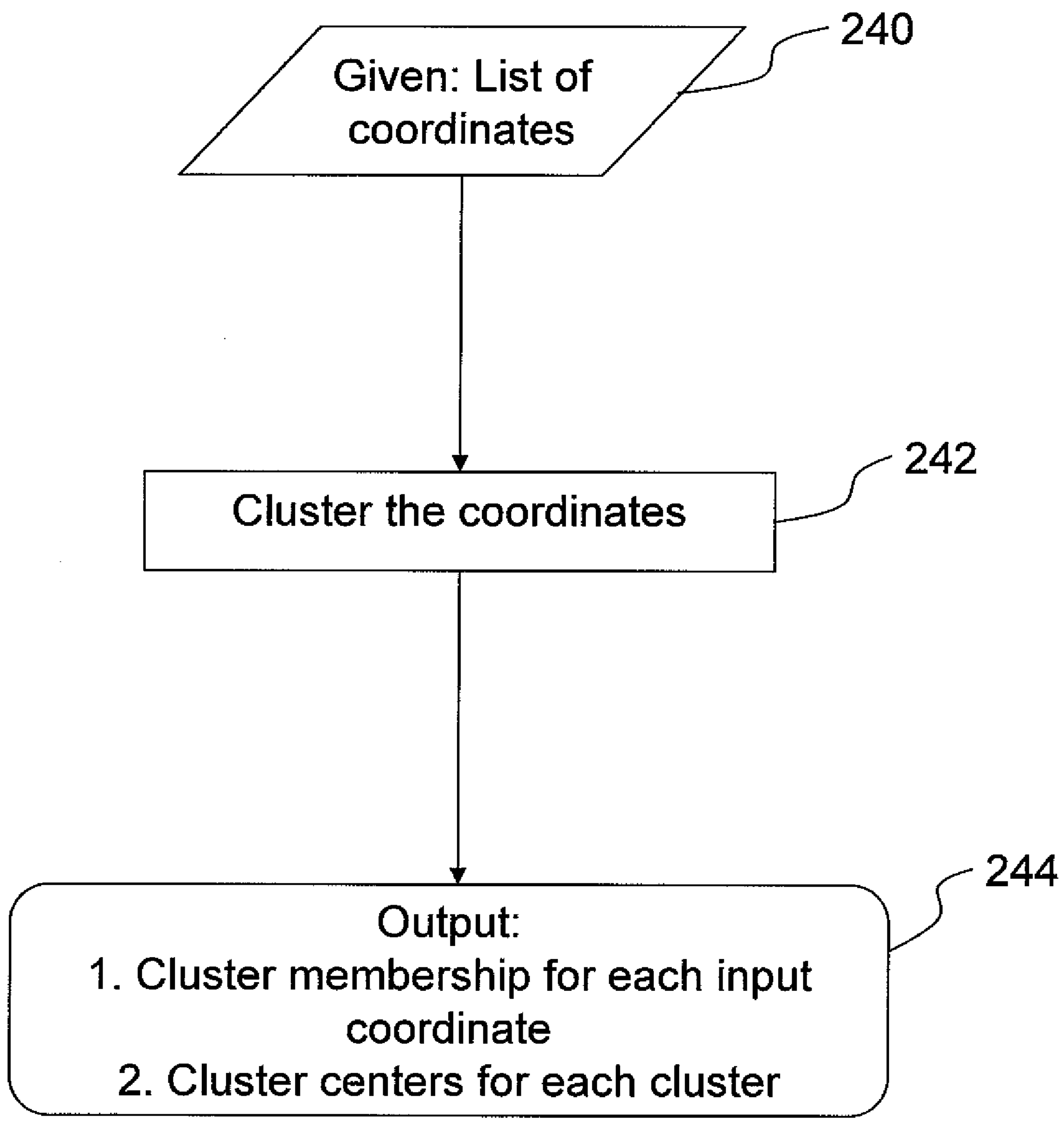


Figure 10: Clustering log chromaticity coordinates

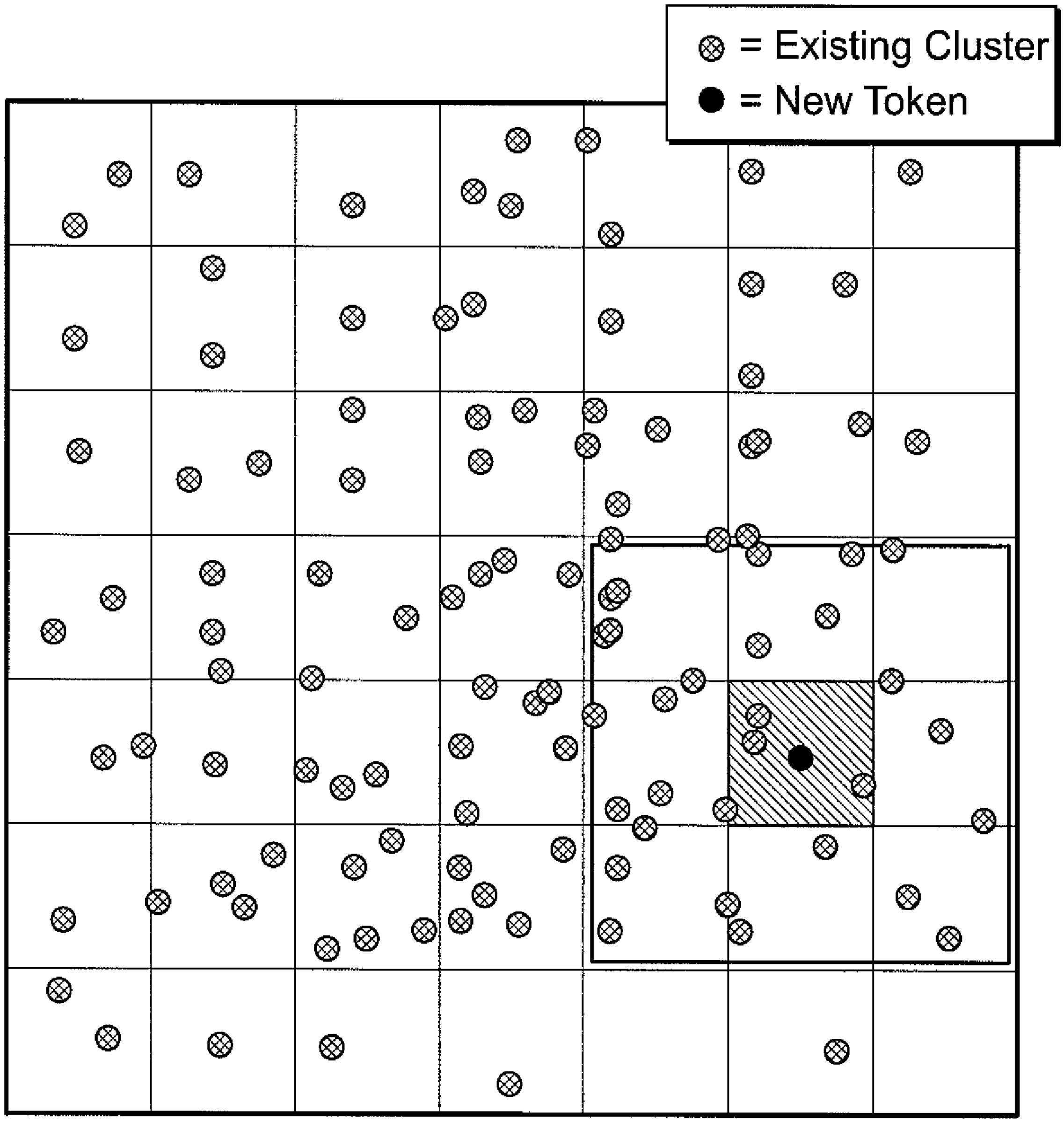


FIG. 10a

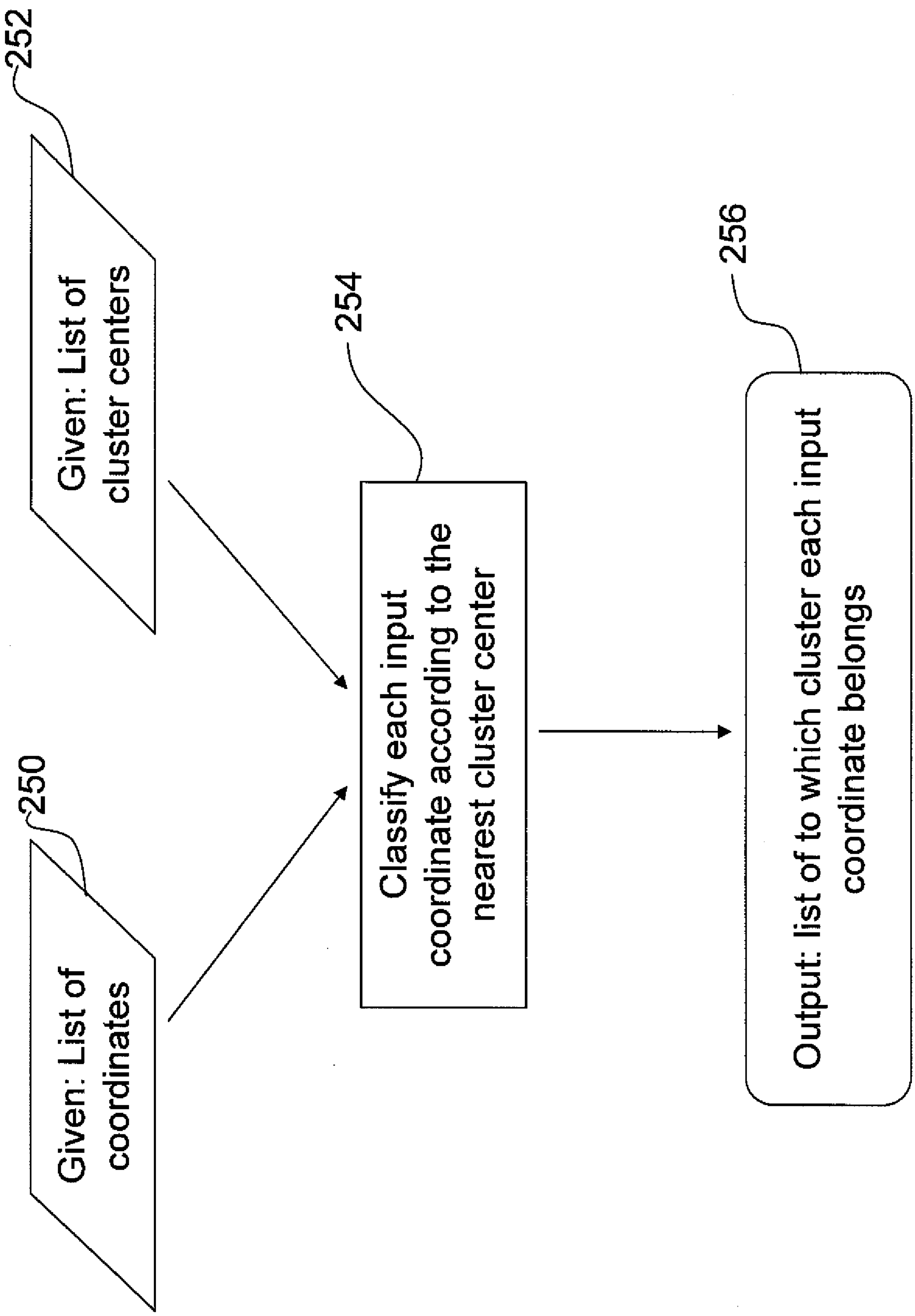


Figure 11: Assigning coordinates to clusters

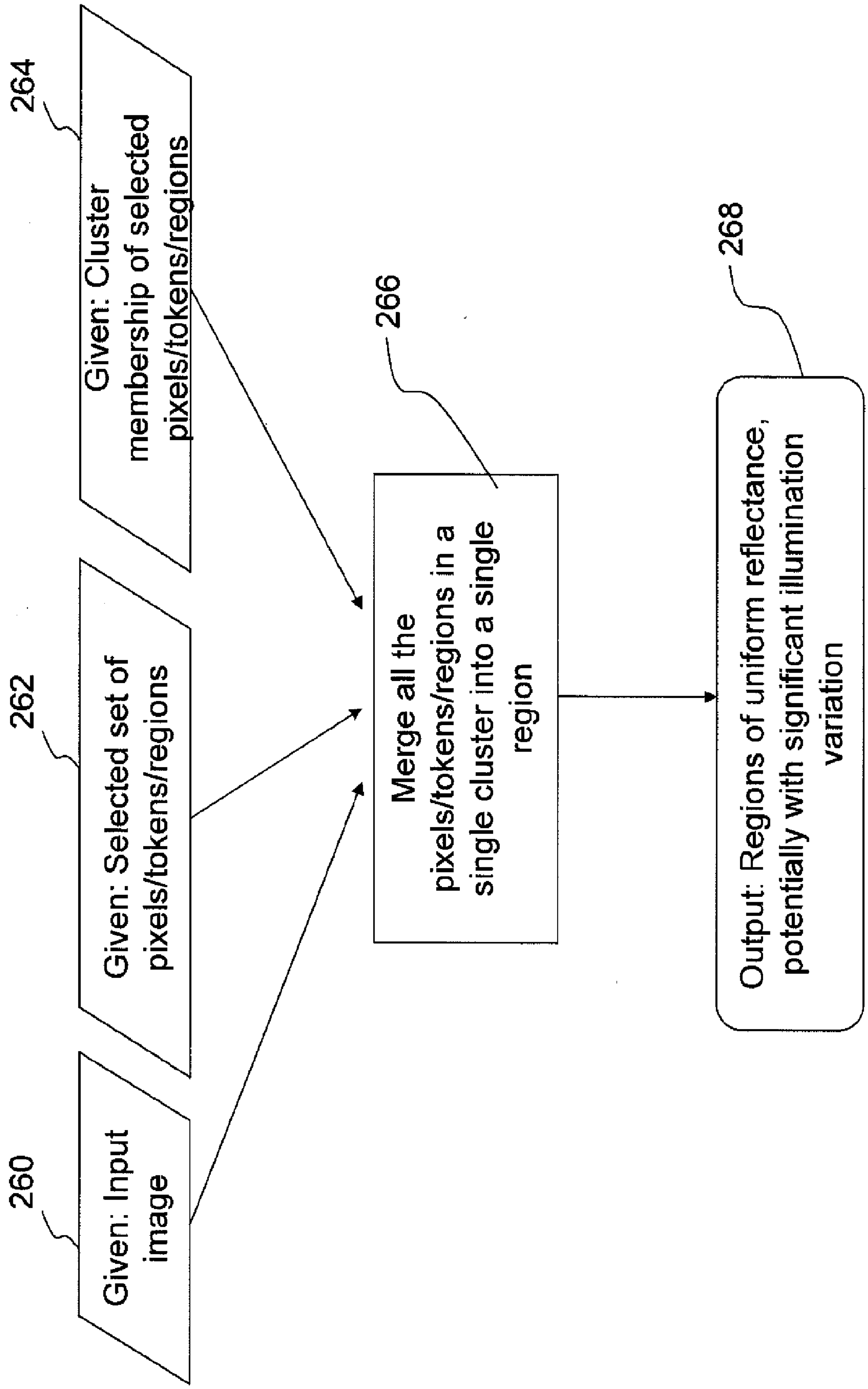
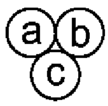


Figure 12: Detecting regions of uniform reflectance based on log chromaticity clustering



$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \begin{bmatrix} i_a - i_b \\ i_a - i_c \\ i_b - i_c \end{bmatrix}$$

$$[A] \qquad [x] = [b]$$

FIG. 13

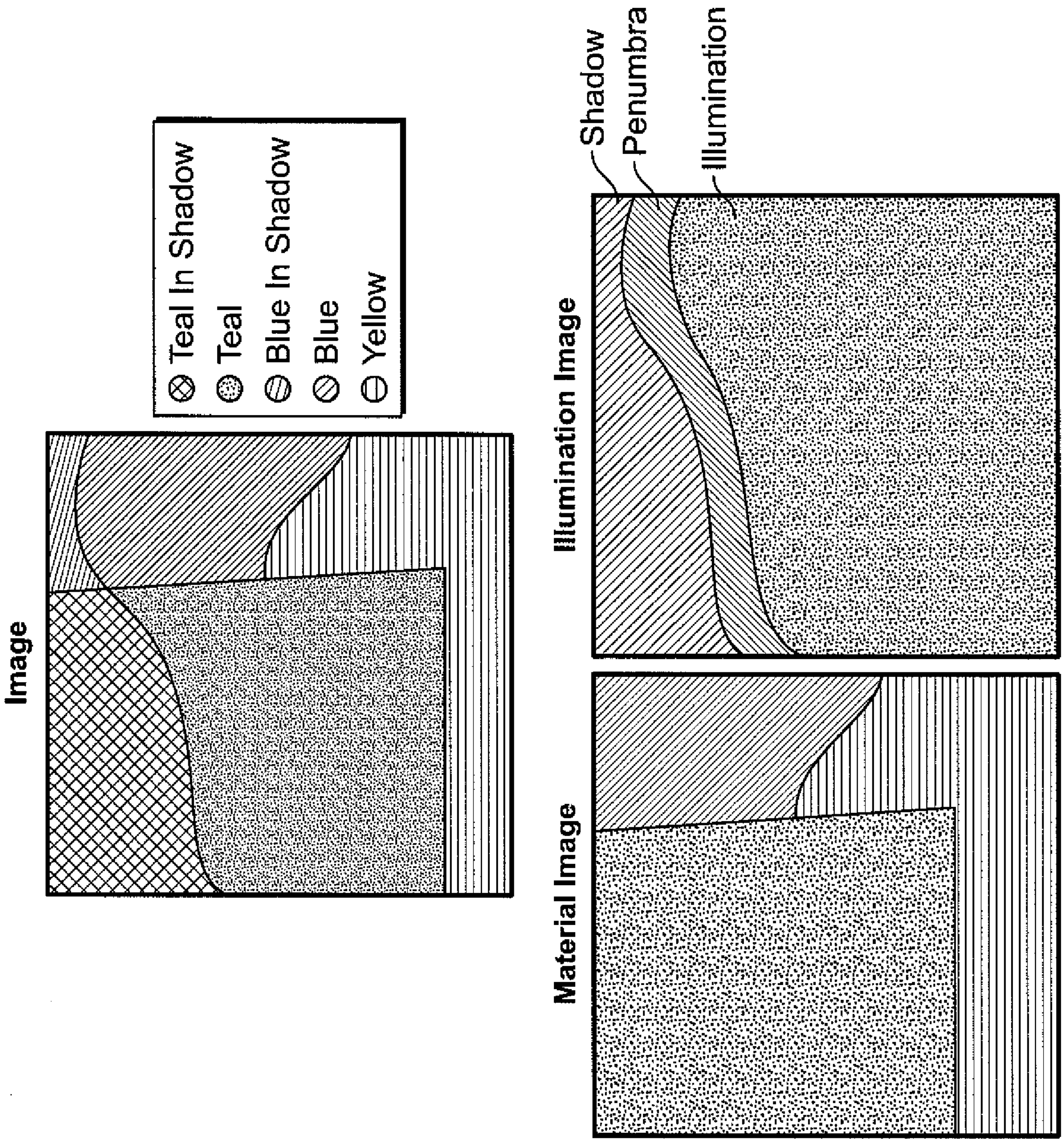


Image
Teal In Shadow
Teal
Blue In Shadow
Blue
Yellow
Illumination Image
Shadow
Penumbra
Illumination
Material Image

FIG. 14

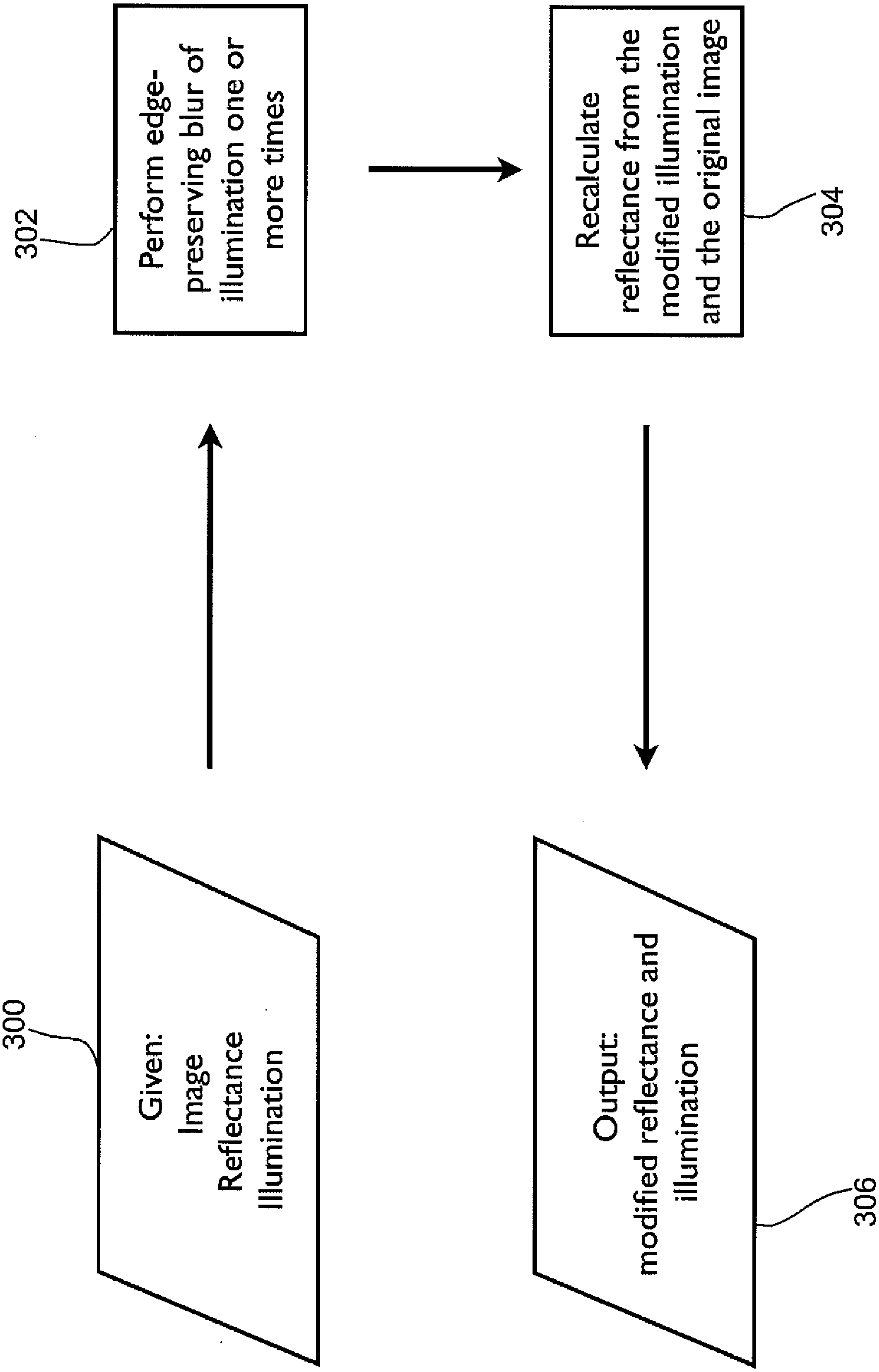

300
Given:
Image
Reflectance
Illumination
302
Perform edge-preserving blur of illumination one or more times
304
Recalculate reflectance from the modified illumination and the original image
306
Output:
modified reflectance and illumination

Figure 15

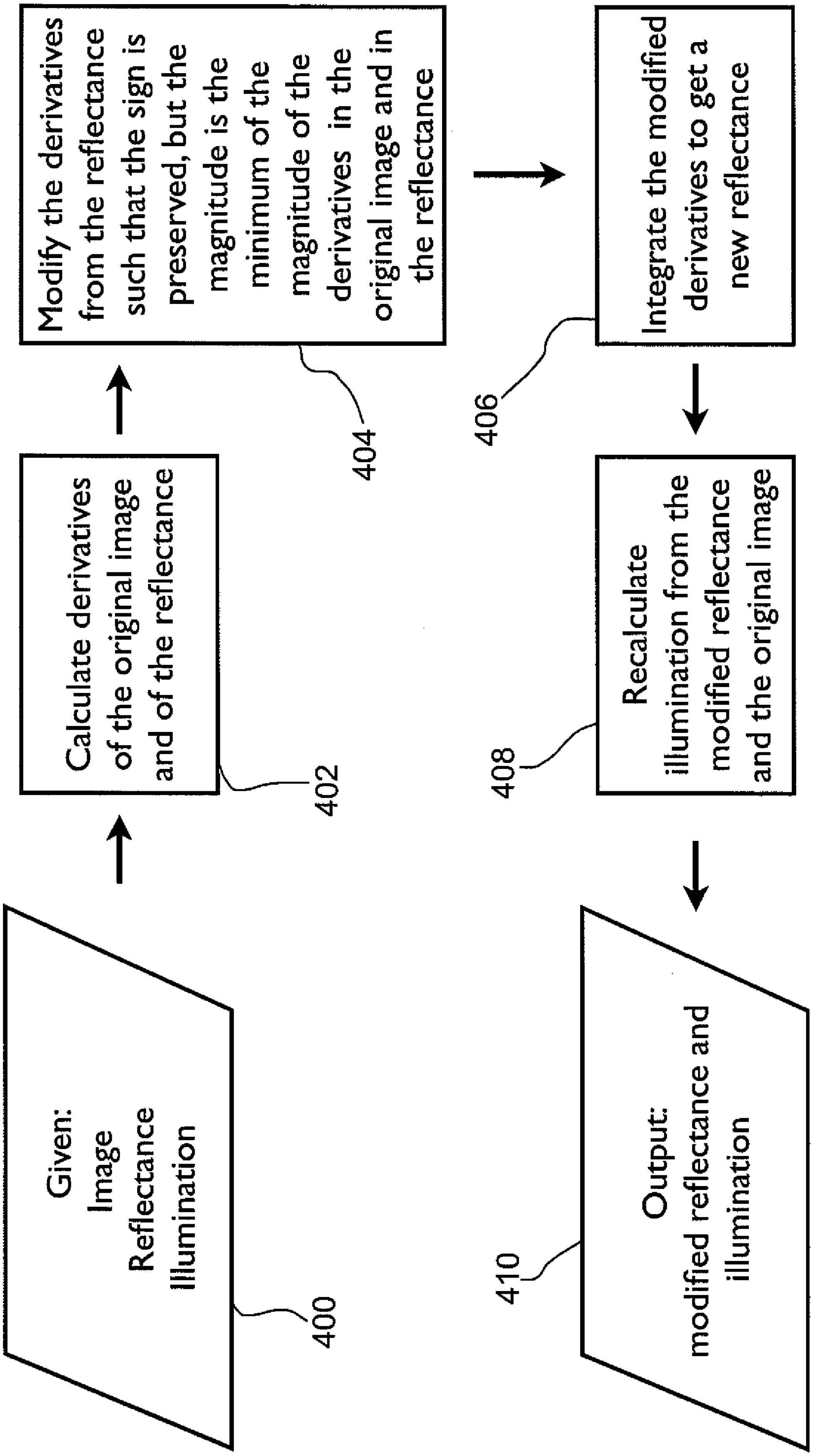

Given:
Image
Reflectance
Illumination
400
Calculate derivatives of the original image and of the reflectance
402
Modify the derivatives from the reflectance such that the sign is preserved, but the magnitude is the minimum of the magnitude of the derivatives in the original image and in the reflectance
404
Integrate the modified derivatives to get a new reflectance
406
Recalculate illumination from the modified reflectance and the original image
408
Output:
modified reflectance and illumination
410

Figure 16

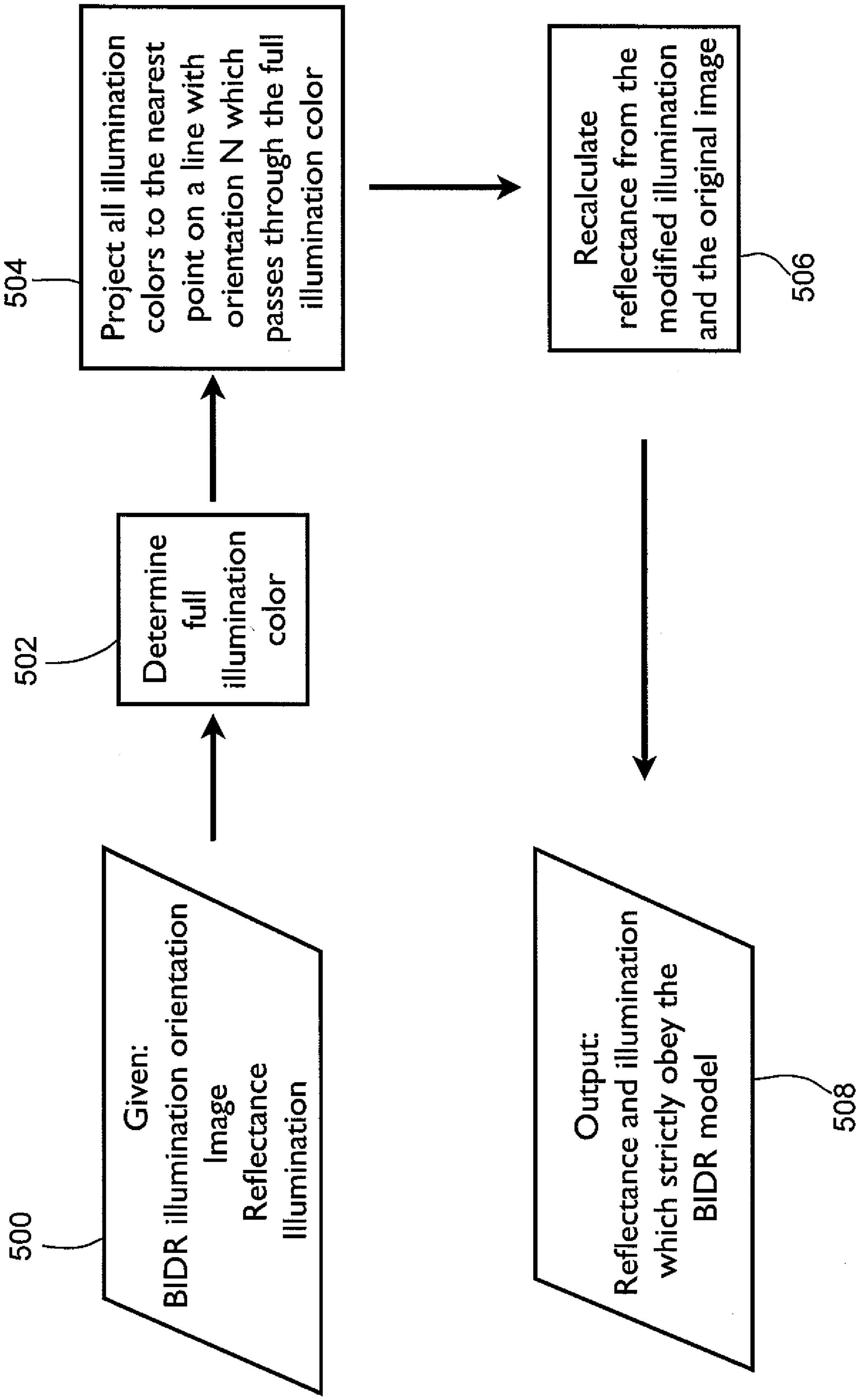

500
Given:
BIDR illumination orientation
Image
Reflectance
Illumination
502
Determine full illumination color
504
Project all illumination colors to the nearest point on a line with orientation N which passes through the full illumination color
506
Recalculate reflectance from the modified illumination and the original image
508
Output:
Reflectance and illumination which strictly obey the BIDR model

Figure 17

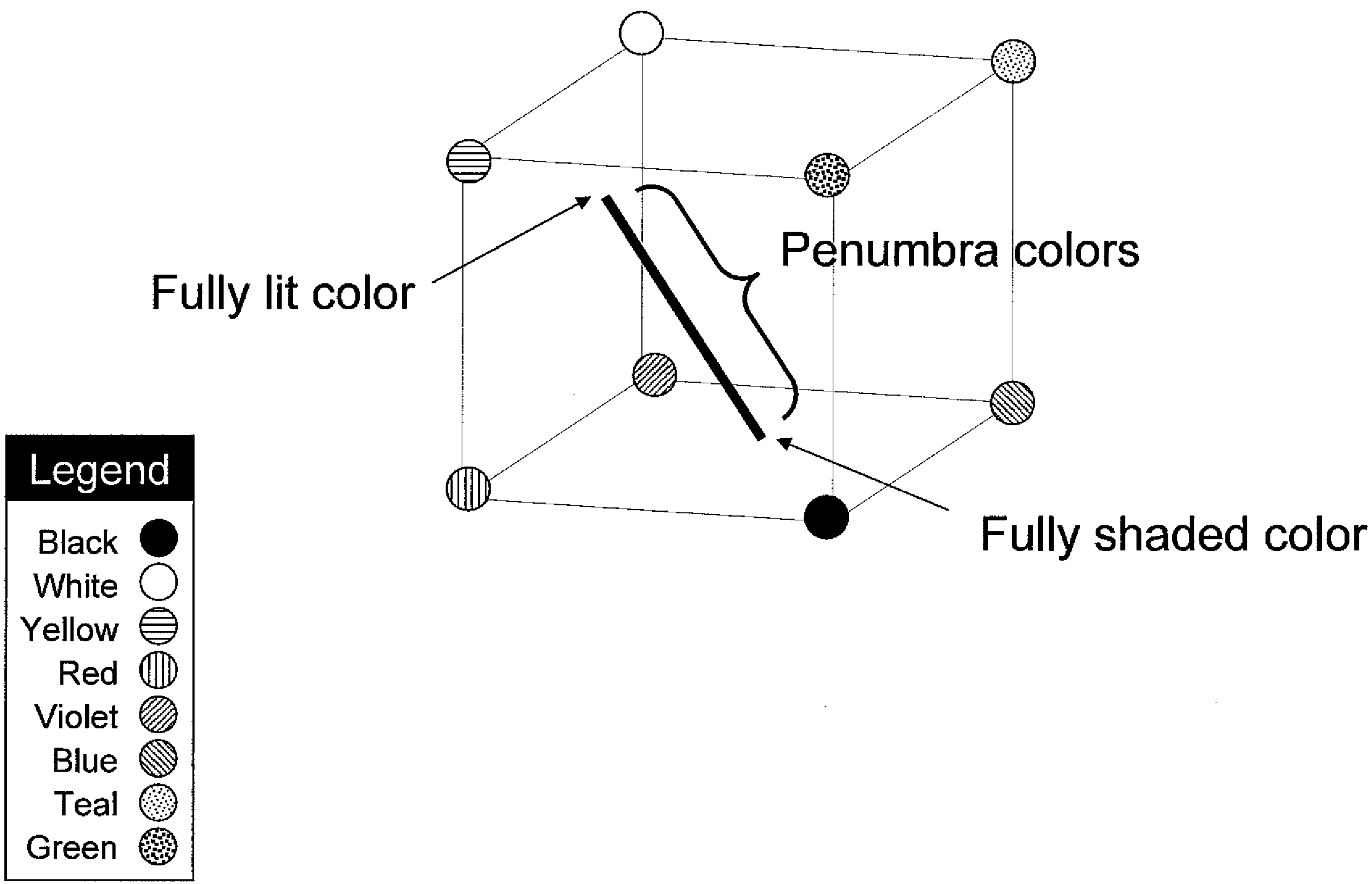


Figure 18: Representation of Body Reflection in RGB Space

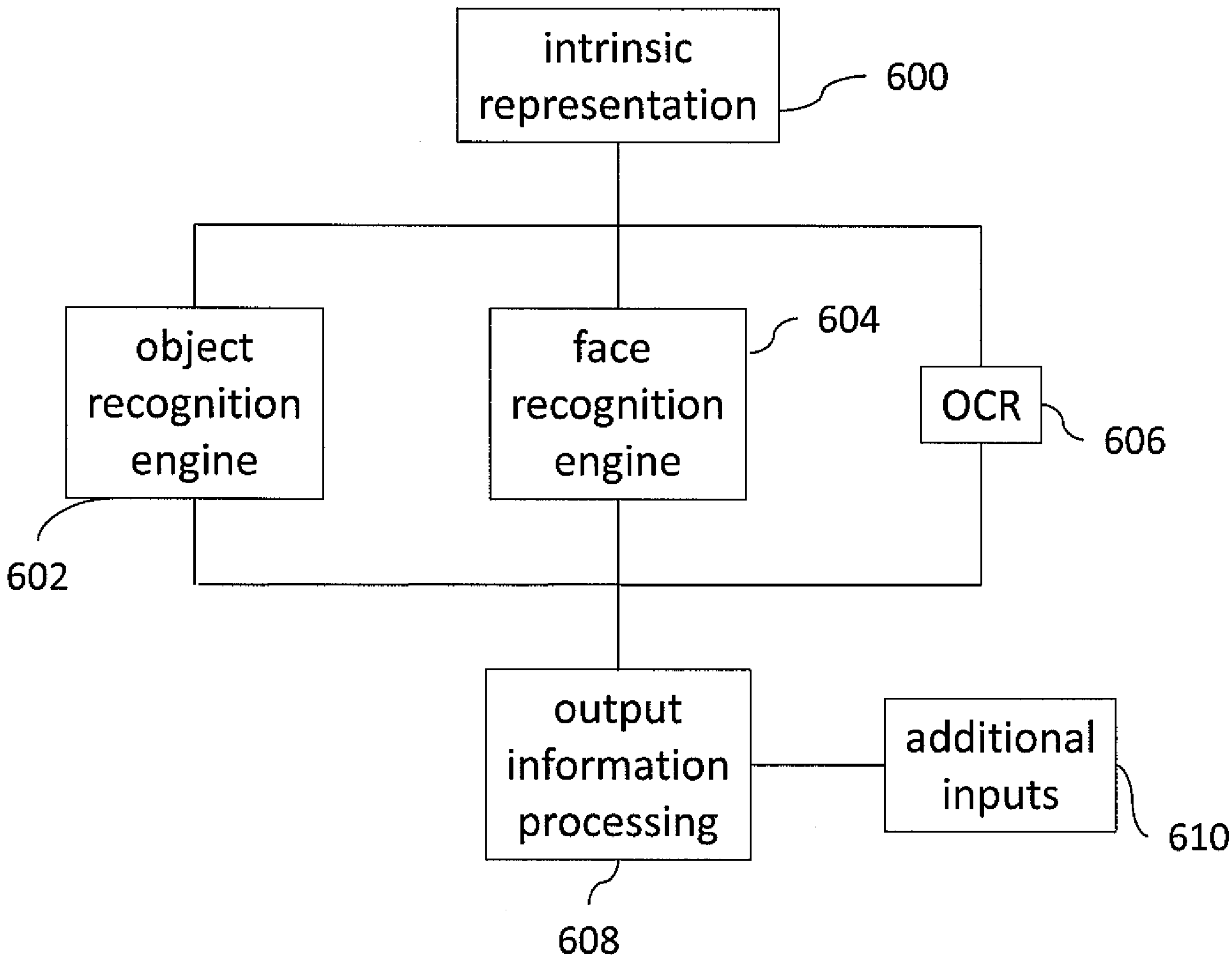

intrinsic representation
600
object recognition engine
602
face recognition engine
604
OCR
606
output information processing
608
additional inputs
610

Figure 19