

(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2016292908 B2**

(54) Title
Pixelwise illuminant estimation

(51) International Patent Classification(s)
G06K 9/00 (2006.01) **G06T 7/00** (2006.01)

(21) Application No: **2016292908** (22) Date of Filing: **2016.06.24**

(87) WIPO No: **WO17/008105**

(30) Priority Data

(31) Number	(32) Date	(33) Country
2015902732	2015.07.10	AU

(43) Publication Date: **2017.01.19**

(44) Accepted Journal Date: **2020.07.16**

(71) Applicant(s)
National ICT Australia Limited

(72) Inventor(s)
Robles-Kelly, Antonio;Mutimbu, Lawrence

(74) Agent / Attorney
FB Rice Pty Ltd, Level 23 44 Market Street, Sydney, NSW, 2000, AU

(56) Related Art
US 20160078317 A1
US 8253824 B2



(43) International Publication Date
19 January 2017 (19.01.2017)

- (51) International Patent Classification:
G06K 9/00 (2006.01) *G06T 7/00* (2006.01)
- (21) International Application Number:
PCT/AU2016/050539
- (22) International Filing Date:
24 June 2016 (24.06.2016)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
2015902732 10 July 2015 (10.07.2015) AU
- (71) Applicant: NATIONAL ICT AUSTRALIA LIMITED
[AU/AU]; Level 5, 13 Garden Street, Eveleigh, New South
Wales 2015 (AU).
- (72) Inventors: ROBLES-KELLY, Antonio; C/- Level 5, 13
Garden Street, Eveleigh, New South Wales 2015 (AU).
MUTIMBU, Lawrence; C/ - Level 5, 13 Garden Street,
Eveleigh, New South Wales 2015 (AU).
- (74) Agent: FB RICE; Level 23, 44 Market St, Sydney, New
South Wales 2000 (AU).

- (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, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, 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, SA, 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, ST, 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, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: PIXELWISE ILLUMINANT ESTIMATION

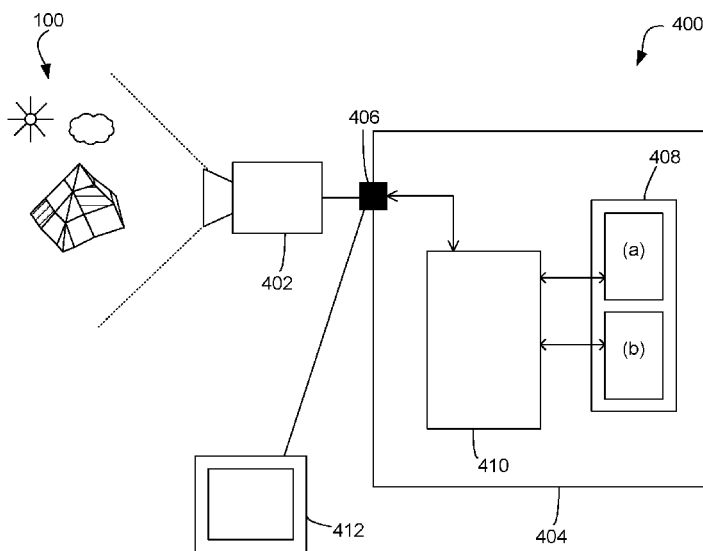


Fig. 4

(57) Abstract: The disclosure concerns estimating an illumination spectrum of a pixel of hyperspectral or multispectral image data. A processor determines an optimised illumination spectrum for a first point based on an initial illumination spectrum and based on associations between the first point and multiple second points of multiple instances of the image data, wherein each of the multiple instances is associated with a refined amount of image information. It is possible to determine the illumination spectrum per pixel of the original image data without clustering pixels or restrictive assumptions but instead by utilising the associations for the statistical inference on the illumination spectrum.

"Pixelwise illuminant estimation"

Cross-Reference to Related Applications

5 The present application claims priority from Australian Provisional Patent Application No 2015902732 filed on 10 July 2015, the content of which is incorporated herein by reference.

Technical Field

10 The disclosure concerns processing of electronic images, such as hyperspectral or multispectral images. In particular the invention concerns, but is not limited to, methods, software and computer systems for estimating an illumination spectrum of a pixel of image data.

Background Art

15 Since the appearance of an object in a scene depends greatly upon the illuminant colour, the recovery of the light power spectrum finds applications in recognition, surveillance and visual tracking. Despite its importance, the recovery and identification of illuminant colours in the scene has proven to be a difficult task in uncontrolled real world imagery. This is mainly due to the fact that the recovery of the region-wise
20 illuminant from a single image is an under-constrained problem. As a result, existing colour constancy algorithms, such as the grey-world, grey-edge and shades of grey method, assume uniform illumination or an achromatic scene. In other solutions, the colour constancy problem is tackled assuming that the statistical mean of the spatial derivative across the image is independent of the chromaticity or it is assumed the
25 reflectance distribution to be Gaussian.

In natural everyday scenes, however, these assumptions can be too restrictive. This arises because in practice, real-world imagery can comprise shadows, highlights and albedo variations arising from several light sources.
30

Any discussion of documents, acts, materials, devices, articles or the like which has been included in the present specification is not to be taken as an admission that any or all of these matters form part of the prior art base or were common general knowledge

in the field relevant to the present disclosure as it existed before the priority date of each claim of this application.

5 Throughout this specification the word "comprise", or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated element, integer or step, or group of elements, integers or steps, but not the exclusion of any other element, integer or step, or group of elements, integers or steps.

Disclosure of Invention

10 A method for determining an illumination spectrum for a first point of image data comprises:

determining the multiple instances of the image data by refining the amount of information associated with each of the multiple instances, wherein refining the amount of information comprises subsampling the image data;and

15 determining an optimised illumination spectrum for the first point based on an initial illumination spectrum and based on associations between the first point and multiple second points of the multiple instances of the image data, wherein each of the multiple instances is associated with the refined amount of image information.

20 Since the optimised illumination spectrum is determined based on associations between points of instances with a refined amount of image information, it is possible to determine the illumination spectrum per pixel of the original image data without clustering pixels or restrictive assumptions but instead by utilising the associations for the statistical inference on the illumination spectrum. This as an advantage over other
25 methods as the determined illumination spectrum is more accurate and less influenced by further input, which would require knowledge about the image data.

The method may further comprise determining the initial illumination spectrum for the first point and multiple initial illumination spectra for each of the multiple second
30 points based on multiple prototypes of illuminant spectra.

The method may further comprise determining the multiple prototypes of illuminant spectra based on the multiple instances of the image data.

35 Determining the multiple prototypes may comprise determining the multiple prototypes based on regions of the image data and the multiple instances of the image data.

The method may further comprise determining the associations between the first point and the multiple second points.

- 5 Determining the associations may comprise creating a relational structure between the multiple instances.

Creating the relational structure may comprise performing a triangulation.

- 10 The method may further comprise determining the multiple instances of the image data by refining the amount of information associated with each of the multiple instances.

Refining the amount of information may comprise applying a filtering operation to the image data.

15

Refining the amount of information may comprise subsampling the image data.

Refining the amount of information may comprise augmenting the image data.

- 20 Determining the optimised illumination spectrum may comprise performing message passing over the associations.

Determining the optimised illumination spectrum may comprise performing a statistical estimation of the illumination spectrum for the first point of the image data.

25

The image data may be one of:

- hyperspectral image data; and
- trichromatic image data.

- 30 The method may further comprise performing white balance adjustment based on the illumination spectrum.

The method may further comprise determining multiple initial illumination spectra for the first point, wherein determining the optimised illumination spectrum is based on the

35

multiple initial illumination spectra for the first point.

Each of the multiple illumination spectra may be associated with a respective subgraph.

Software, when installed on a computer, causes the computer to perform the above method.

5

A computer system for determining an illumination spectrum for a first point of image data comprises:

an input port to receive the image data; and

a processor to

10

determine the multiple instances of the image data by refining the amount of information associated with each of the multiple instances, wherein refining the amount of information comprises subsampling the image data; an

15

determine an optimised illumination spectrum for the first point based on an initial illumination spectrum and based on associations between the first point and multiple second points of the multiple instances of the image data, wherein each of the multiple instances is associated with the refined amount of image information.

Optional features described of any aspect, where appropriate, similarly apply to the other aspects also described here.

20

Brief Description of Drawings

Fig.1 illustrates an example scene (prior art).

Fig. 2 illustrates the example scene of Fig. 1 in more detail (prior art).

25

Fig. 3 illustrates the representation of radiance spectra in a spectral space (prior art)..

An example will be described with reference to

Fig. 4 illustrates a computer system for determining an illumination spectrum for a point of image data.

30

Fig. 5 illustrates a computer implemented method for estimating an illumination spectrum for a point of image data.

Fig. 6 illustrates a data structure for the multispectral image data.

35

Fig. 7a illustrates a scale space and the interaction of uniform grid patches across scales. This shows how a pyramid-like graph is induced by the coarse-to-fine relationship between pixels at different scales.

Fig. 7b illustrates a factor graph arising from the pyramid in Fig. 7a, where variables are shown as round markers whereas the factors are depicted as squares.

Fig. 8a illustrates a factor graph where interrelated variables (round nodes) are connected to factors (square nodes).

5 Fig. 8b illustrates incoming and outgoing messages along edges between variable and factor nodes within the same scale, and nodes in adjacent scales in the graph.

Figs. 9a and 9b illustrates respective example images.

Figs 10a and 11a illustrate respective real-world input images.

10 Figs. 10b and 11b illustrate respective illuminant maps yielded by the method of Fig. 5 when applied to the images of Figs. 10a and 11b.

Figs 10c and 11c illustrate respective corrected images.

Fig. 12 illustrates the relationships between subgraphs and pixels at different scales.

15 Fig. 13 illustrates a method for pixelwise illuminant recovery.

Fig. 14 illustrates a method for initialising an illuminant estimate.

Best Mode for Carrying Out the Invention

20 Fig. 4 illustrates a computer system 400 for estimating an illumination spectrum of an image of scene 100. Computer system 400 comprises a sensor 402 and a computer 404. In this example the sensor 402 is a hyperspectral or multispectral sensor that is able to capture an image of a scene 100 illuminated by three light sources as explained with reference to Fig. 1.

25 In one example, the computer system 400 is integrated into a handheld device such as a consumer camera and the scene 100 may be any scene on the earth, such as a tourist attraction or a person. The sensor 402 may have a number of bands that balances computational costs with accuracy. The sensor 402 may have as low as four bands and as high as hundreds. In one example, sensor 402 is a Fluxdata camera FD-1665-MS7.

30 The computer 404 receives images from the sensor 402 via a data port 406 and the images are stored in local memory 408(b) by the processor 410. The processor 410 uses software stored in memory 408(a) to perform the method shown in Fig. 5. The program memory 408(b) is a non-transitory computer readable medium, such as a hard drive, a solid state disk or CD-ROM.

35

The processor 410 performs the method of estimating an illumination spectrum of the image.

5 Processor 410 may perform a method for decomposing hyperspectral or multispectral image data as described in US Patent No. 8,670,620 "Decomposing hyperspectral or multispectral image data", which incorporated herein by reference. For storing the input spectra, illumination spectra or other spectra the computer may employ the method described in WO 2009/152583 "Compact Representation of a Reflectance Spectrum" which is incorporated herein by reference.

10

The software provides a user interface that can be presented to the user on a monitor 412. The user interface is able to accept input from the user (i.e. touch screen). The user input is provided to the input/out port 406 by the monitor 412. The image is stored in memory 408(b) by the processor 410. In this example the memory 408(b) is local to the compute r404, but alternatively could be remote to the computer 404.

15

The processor 410 may receive data, such as image data, from data memory 408(b) as well as from the communications port 406. In one example, the processor 410 receives image data from the sensor 402 via communications port 406, such as by using a Wi-Fi network according to IEEE 802.11. The Wi-Fi network may be a decentralised ad-hoc network, such that no dedicated management infrastructure, such as a router, is required or a centralised network with a router or access point managing the network.

20

In one example, the processor 410 receives and processes the image data in real time. This means that the processor 410 determines the illuminant spectrum every time the image data is received from sensor 402 and completes this calculation before the sensor 402 sends the next image data update.

25

Although communications port 406 is shown as single entity, it is to be understood that any kind of data port may be used to receive data, such as a network connection, a memory interface, a pin of the chip package of processor 410, or logical ports, such as IP sockets or parameters of functions stored on program memory 408(a) and executed by processor 410. These parameters may be stored on data memory 408(b) and may be handled by-value or by-reference, that is, as a pointer, in the source code.

30

35

The processor 410 may receive data through all these interfaces, which includes memory access of volatile memory, such as cache or RAM, or non-volatile memory, such as an optical disk drive, hard disk drive, storage server or cloud storage. The computer system 404 may further be implemented within a cloud computing environment, such as a managed group of interconnected servers hosting a dynamic number of virtual machines.

It is to be understood that any receiving step may be preceded by the processor 410 determining or computing the data that is later received. For example, the processor 410 determines the image data, such as by filtering the raw data from sensor 402, and stores the image data in data memory 408(b), such as RAM or a processor register. The processor 410 then requests the data from the data memory 408(b), such as by providing a read signal together with a memory address. The data memory 408(b) provides the data as a voltage signal on a physical bit line and the processor 410 receives the image data via a memory interface.

Fig. 5 illustrates a computer implemented method 500 for estimating an illumination spectrum for a point of an image as performed by processor 410. In other words, method 500 may serve as a blueprint or pseudo-code for software implemented in a particular programming language, such as C++, and stored on program memory 408(a) as compiled machine readable code. The image is comprised of points of wavelength indexed spectral data, such as multispectral image data or trichromatic data, such as RGB data.

Method 500 will be explained in broad terms first while a more detailed and more mathematical description follows afterwards.

Fig. 6 illustrates a data structure 600 for the multispectral image data. The data structure 600 comprises layers, one for each wavelength. Each layer represents the radiance values for one wavelength and all pixels and one example pixel 602 is highlighted. The values of pixel 602 for different wavelengths, that is the radiance values from lower layers at the same location as pixel 602, represent a radiance spectrum also referred to as the image spectrum or input spectrum. This input spectrum may be a mixture of multiple illumination spectra and the reflectance spectra of different materials present in the part of the scene that is covered by pixel 602.

In the following description, the term ‘pixel’ can be replaced by ‘point of the image’ to denote that the individually addressable image elements may be computed based on multiple pixels. For example, the image resolution may be reduced by combining pixels and the method 500 is performed on the low-resolution image having multiple points instead of pixels. Unless noted otherwise, if the word ‘pixel’ is used it may equally be applicable to a ‘point of the image’.

Method 500 in Fig. 5 is based on multiple instances of the image data and Fig. 7a illustrates an example of multiple instances 702, 704 and 706, respectively, forming a scale space 700. Each of the multiple instances is associated with a refined amount of image information. While the examples below relate to instances associated with a reduced amount of image information, other examples relate to augmented image information, such as after noise reduction. Therefore, a refined amount of image information may be a reduced or an augmented amount of image information.

In the example of reduced image information, the multiple instances may also be referred to as multiple scales and define the scale space 700. In one example, instances 702 and 704 are copies of the original. Copies 704 and 702 are subjected to a blurring filter to reduce the amount of information in those copies. The radius of the blurring filter is larger for copy 702 than for copy 704 such that the least amount of information is associated with copy 702. Fig. 7a further shows an example point 708 of the image data at the finest scale, that is, the instance 706 associated with the maximum amount of information, which may be the original image data. In the example of augmented image data for each instance, the filter may be a sharpening or de-speckle filter. Augmenting the image information may also involve using additional sensor information, such as dark noise, a library or user provided input.

It is noted that each instance 702, 704 or 706 may be represented by an image cube 600 shown in Fig. 6. However, the multiple layers for the different wavelengths are omitted for clarity in Figs. 7a and 7b.

As can be seen in Fig. 7a, the instances with reduced information 702 and 704 also comprise multiple points. In one example, the number of points in instances 702 and 704 is the same as the number of points in the original image data 706 and each point of instances 702 and 704 is the average of surrounding points. This average may be calculated for each wavelength index separately.

As a result, each point in the coarse instance 702 can be associated to the points of the middle instance 704 that contribute to the average of the point in the coarse instance 702. In other words, all points in middle instance 704 that are within the radius of the smoothing filter are associated with the resulting point in the coarse instance 702. Vice versa, points in the original image data instance 706 within the filter radius are associated with a point in the middle instance 704.

In another example, the points of the original image data 706 are subsampled and all points that are dropped in the neighbourhood of one point are associated with that point. Further below, these associations are formalised in the form of a factor graph 750 as shown in Fig. 7b.

Referring back to Fig. 5, processor 410 performs method 500 by determining an optimised illumination spectrum for a point 708 based on an initial illumination spectrum and based on associations between the first point and multiple second points of multiple instances 702 and 704 of the image data, wherein each of the multiple instances 702 and 704 is associated with a reduced amount of image information as described above.

Multiple instances may refer to making multiple copies of the image data on the data memory 408(b) or may refer to a list of pointers stored on data memory 408(b). In yet another example, determining multiple instances may refer to executing program code that comprises a loop statement over the image data where the loop variable is updated such that at each iteration the accessed image data has a reduced amount of information. For example, by updating the loop variable i by $i+=2$, every second pixel is addressed and the image data is subsampled to reduce the amount of information in the accessed instance of the image data.

Processor 410 may determine the initial illumination spectra for the point 708 of each of the multiple instances based on multiple prototypes that may also be determined by processor 410 based on irregular regions or regular regions, such as patches, of the original image instance 706 and the reduced image instances 702 and 704.

This disclosure presents a method as performed by processor 410 to recover the pixelwise illuminant colour for scenes lit by multiple lights according to method 500.

The following explanation starts from the image formation process and poses the illuminant recovery task in hand into an evidence combining setting. Processor 410 constructs a factor graph 750 making use of the scale space of the input image and a set of illuminant prototypes. It is noted that other relational structures, such as other
 5 graphs including metric graphs and hypergraphs may equally be used.

The computation of these prototypes is data driven and, hence, the method is devoid of libraries or user input. The use of a factor graph allows for the illuminant estimates at different scales, that is, instances with reduces amount of information, to be recovered making use of an inference process. Here, this process is statistical in nature.
 10 Moreover, the computation of the probability marginals used here is rendered as exact by constructing our factor graph making use of a Delaunay triangulation. For instance, consider the image of Fig. 1. The scene is being lit by two different light sources, that is, sun 104 and cloud 106. Processor 410 recovers pixelwise illuminants by making use of an inference process on a factor graph 750 defined across the scale space of the
 15 image under consideration. In some examples, the proposed method 500 does not assume smoothly varying illuminants across the scene. Secondly, some examples of the method 500 do not impose constraints on the reflectance or illumination gamuts for neither the input nor output imagery. Thirdly, processor 410 models the relationships between pixels and across scales using a factor graph [Kschischang01]. Moreover, the
 20 inference process used here is an exact one which hinges in the use of a variable elimination approach which does not change the topology of the graph.

Illuminant Estimation

As mentioned earlier, the explanation starts with the image formation process. Suppose that a location u in the scene is illuminated by a power spectrum $L(\lambda)$, where λ is the
 25 sample wavelength variable. Then the spectral radiance $I(u, \lambda)$ reflected from that location can be expressed as

$$I(u, \lambda) = L(u, \lambda)R(u, \lambda), \quad (1)$$

where $L(u, \lambda)$ is the power spectrum of the light impinging on the object surface and $R(u, \cdot)$ is the product of the mean scattered power in the direction of the viewer and the surface reflectance spectrum [Horn86][Kimmel03].

In the further analysis, processor 410 considers a set of illuminant prototypes Υ . We
 5 also introduce the scale space of the image and express the illuminant at pixel u and scale σ as a linear combination of the form

$$L_\sigma(u) = \sum_{\ell_i \in \Upsilon} \omega_{i,\sigma}(u) \ell_i, \quad (2)$$

where $\omega_{i,\sigma}(u)$ is the weight of the i^{th} illuminant $\ell_i \in \Upsilon$ at scale σ . In the equation above, and as a matter of convenience, we have omitted the wavelength variable λ .

10 Note that this can be done without any loss of generality and, hence, we do likewise throughout the remainder of the disclosure.

The use of the expression above allows processor 410 to treat the problem in hand in a manner akin to that used in labelling problems where the i^{th} label corresponds to $\ell_i \in \Upsilon$. It is worth noting that the introduction of the set Υ is not overly restrictive. Moreover,
 15 here, processor 410 adopts a data driven approach where the prototypes in Υ are computed from the input image. This implies that the method can be devoid of a pre-computed library or user-aided input for the recovery of the prototypes ℓ_i . More details on the computation of Υ is provided further below.

Factor Graphs

20 As mentioned earlier, processor 410 uses factor graphs to recover the pixelwise illuminant. The choice of factor graphs stems from the fact that these are a generalisation of probabilistic graphical models, which allow processor 410 to formulate the problem in an evidence combining setting where the pixelwise illuminant can be viewed as a mixture of the prototypes Υ across different scales. Processor 410

may use the max-product algorithm in factor graphs [Kschischang01] for purposes of inference.

Factor graphs generalise both Markov random fields and directed graphs such as Bayesian networks. They also model variable relationships that are not necessarily
5 probabilistic.

To construct the graph, processor 410 employs the scale space of the image. Figure 7a illustrates the interaction between three image scales 702, 704 and 706. This, in effect, is a combinatorial pyramid [Brun00] where there is an inherent relationship of many-to-one from finer to coarser scales. Across each scale, each pixel also interacts with its
10 neighbours.

Fig. 7b illustrates the factor graph 750 arising from the pyramid in the left-hand panel. In Fig. 7b, the factors are given by squared tokens whereby the variables in the graph are denoted by circles on the image lattice.

From a probabilistic viewpoint, in Figure 7b, interacting variables can be used to define
15 the nature of the conditional distributions within each scale. More formally, let the factor graph and the image scale be denoted by \mathcal{G} and σ , respectively. In one example, we can express the probability of the pixel-wise illuminants $L(u) \in L$ across the image as follows

$$P(L | \mathcal{G}) = \frac{1}{Z} \prod_{c \in C} \psi(L_c; \mathcal{G}) \quad (3)$$

$$= \frac{1}{Z} \prod_{u, v \in \mathcal{G}^{u \sim v}} P(I(u), I(v) | L(u), L(v))$$

$$\prod_{\sigma \in \Gamma} \prod_{\ell_i \in Y} P(L_\sigma(u) | \ell_i).$$

20

where Γ is the set of scales under consideration, C is the clique-set of the graph, $u \sim v$ implies the pixels u and v are adjacent to each other, $\psi(L_c; \mathcal{G})$ is the potential functions over the variable nodes in the graph for the clique c and Z is a partition function.

- 5 It is worth noting in passing that Equation (3) effectively conditions the illuminant distribution across the image to the potential functions for the variable nodes over the graph and the prototypes in Υ . This is as each of the potential functions is defined over a subset of variables L_c which comprises the illuminant estimates at each of the cliques $c \in C$. In this manner, every variable in a clique is connected to at least one other
- 10 variable through a factor node. The union of variables, factors and the edges that connect them constitute the factor graph \mathcal{G} .

- In processing Equation (3) processor 410 uses Bayes' rule to calculate the solution in terms of the likelihood and prior distributions with the partition function Z acting as a normalisation constant. Here, we have also assumed that illuminant priors $P(L_\sigma(u) | \ell_i)$
- 15 for the prototype illuminants across image scales $\sigma \in \Gamma$ are independent from each other.

- Note that, in some examples, the pixel neighbours on each scale can be used to compute cliques which, in turn, can be employed by processor 410 to define the likelihood function for the illuminant estimation process. This is reflected by the
- 20 probability $P(I(u), I(v) | L(u), L(v))$, which captures the likelihood of the observed image data at neighbouring pixel sites given their respective illuminant estimates. On the other hand, the priors $P(L_\sigma(u) | \ell_i)$ correspond to the likelihood of an illuminant for a given pixel u at scale σ up to the prototype set Υ .

- Moreover, the Bayesian formulation in problem (3) lends itself, by construction, to a
- 25 probability factorisation over the graph. This may be important since the inference process in hand can now be posed as a *maximum a posteriori* (MAP) process whose

aim of computation is marginal probabilities $P(L_\sigma(u))$. This can be viewed as the optimisation given by

$$\begin{aligned} & \underset{L}{\operatorname{argmax}} P(L | \mathcal{G}) \\ &= \underset{L}{\operatorname{argmax}} \left\{ \prod_{u,v \in \mathcal{G}u \sim v} P(I(u), I(v) | L(u), L(v)) \right. \\ & \quad \left. \prod_{\sigma \in \Gamma} \prod_{\ell_i \in \Upsilon} P(L_\sigma(u) | \ell_i) \right\} \end{aligned} \quad (4)$$

- 5 where we have removed the partition function Z from further consideration since it is a constant that does not affect the maximisation above.

It is noted that instead of a MAP process, a maximum likelihood estimation or a maximum description length method may be used by processor 410.

- 10 For the computation of the illuminant prior conditional probability, processor 410 uses the angular error between the pixelwise illuminant and each of the prototypes in Υ to write

$$P(L_\sigma(u) | \ell_i) = \frac{1}{Y} \exp \left(- \left(1 - \frac{\langle L_\sigma(u), \ell_i \rangle}{\|L_\sigma(u)\| \|\ell_i\|} \right) \right), \quad (5)$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $\|\cdot\|$ is the vector norm and Y is a normalisation constant given by

$$15 \quad Y = \sum_{\ell_i \in \Upsilon} \exp \left(- \left(1 - \frac{\langle L_\sigma(u), \ell_i \rangle}{\|L_\sigma(u)\| \|\ell_i\|} \right) \right) \quad (6)$$

Similarly, for the likelihood, neighbouring pixels may be likely to have similar illuminant colour if their reflectance and shape are close to one another. Recall that the $R(u, \lambda)$ variable in Equation (1) is the product of the reflectance and the mean scattered power. Moreover we note that the variable $R(u, \lambda)$ can be written in compact
 5 form as follows

$$R(u) = L(u)^{-T} \mathbf{I}(u) \quad (7)$$

where \mathbf{I}_n is an $n \times n$ the identity matrix.

Thus, we write

$$P(I(u), I(v) | L(u), L(v)) = \frac{1}{\Omega} K(L(u), L(v)) \times \exp(-\|R(u) - R(v)\|^2), \quad (8)$$

where Ω is a normalisation constant given by

$$\Omega = \sum_{\substack{u, v \in \mathcal{G} \\ u \sim v}} K(L(u), L(v)) \exp(-\|R(u) - R(v)\|^2) \quad (9)$$

and $K(L(u), L(v))$ is a kernel function.

For the kernel function, and similarly to Equation (5), processor 410 uses the
 15 exponential function of the Euclidean angle between the illuminant for adjacent pixels and write

$$K(L(u), L(v)) = \exp\left(-\left(1 - \frac{\langle L(u), L(v) \rangle}{\|L(u)\| \|L(v)\|}\right)\right). \quad (10)$$

Inference Process

Note that Equations (5) and (8) provide the ingredients necessary to solve the MAP problem in Equation (4). The aim hence becomes that of inferring the marginal probabilities $P(L_\sigma(u))$ of the illuminants at the finest scale. This is achieved through the use of a sum/max-product algorithm [27] which is a message passing strategy.

- 5 The max-product algorithm computes exact probability marginals of variables in a tree [31, 2]. It is worth noting that the developments below also apply to the max-product. This is since both, the max-product and sum methods are equivalent up to a log-space mapping that does not affect the maximum of the likelihood. In order to take advantage of the exact inference property of the max-product algorithm, processor 410 partitions
- 10 the factor graph and converts the sub-graphs into a collection of local tree structures. Performing exact inference on factor graphs with cycles, such as the grid-graph arising from the lattice on the imagery by converting them into trees is a procedure may be effected via variable elimination [42, 12]. Variable elimination algorithms, however, introduce fill-edges for cycles larger than three. As a result, they often change the
- 15 topology of graphs under study [Barber12]. This is an undesirable effect since it invariably alters the inference problem in hand.

In some examples, processor 410 performs variable elimination using a convex hull algorithm for Delaunay triangulations [Watson81] since Delaunay triangulations are chordal graphs, *i.e.* every cycle comprising four or more vertices has a chord between

20 them, that do not introduce any fill edges in a sum(max)-product variable elimination ordering [Koller09]. Further, Delaunay triangulations have low tree-width, *i.e.* the number of variables in a clique, reducing computational complexity during message passing.

Figs. 8a and 8b show a simplified realisation of the factor graph for the MAP problem

25 in Equation (4) as stored on program memory 408(a). Fig. 8a shows the joint interaction between variables across different scales. This is, effectively, a realisation of the pyramid-like structure in Fig. 7b. Fig. 8b illustrates the message passing process between variables and factors in a close-up diagram. In the figure, and for the sake of

clarity, we have adopted the following notation. For the illuminant variables at different scales, we have used x and y , respectively. We have denoted the factors as f and g and used μ to indicate a message flowing in the direction of the sub-indexed arrow.

Note that, from the figure, we can observe that the variable node x "passes on" the corresponding probability to a factor node f , *i.e.*

$$\mu_{x \rightarrow f} = \prod_{\substack{g \in \mathcal{N}_x \\ g \neq f}} \mu_{g \rightarrow x}, \quad (11)$$

where \mathcal{N}_x is the set of factor nodes neighbouring the variable node x .

Moreover, the factor-to-variable interactions can be expressed explicitly as follows

$$\mu_{f \rightarrow x} = \prod_{\substack{y \in \mathcal{N}_f \\ y \neq x}} \mu_{y \rightarrow f}, \quad (12)$$

where \mathcal{N}_f is the set of all variable nodes in the neighbourhood of the factor f . The process of message passing as performed by processor 410 entails computing probability distributions and marginalisation.

The definition of the distributions at factors is determined by the nature of the node, *i.e.* either intra-scale likelihood or within-scale factor node, respectively. The "message" of a given variable is then computed by using a sum- or maximum-rule to marginalise posteriors over all other variables, except the one being queried.

After the messages have been sent in both directions on all edges in the graph, each factor will contain the joint posterior probability distributions of the variables adjacent to the factor under consideration. As a result, in order to obtain the marginal distribution for any one variable in the clique processor 410 marginalises with respect to the adjacent variables making use of the joint distribution.

As a result of this treatment, processor 410 can compute the optimal illuminant estimate $L_\sigma(u)$ at each pixel u and scale σ using the marginals as the weights in Equation (2). Thus, making use of the notation above and Bayes rule:

$$P(\ell_i | L_\sigma(u)) = \frac{P(L_\sigma(u) | \ell_i)P(\ell_i)}{P(u)} \quad (13)$$

- 5 Note that the probability $P(\ell_i | L_\sigma(u))$ is, in practice, the posterior probability of a pixel being illuminated by the illuminant $\ell_i \in \Upsilon$. For all pixels, this is equivalent to the distribution $P(L | \mathcal{G})$.

Similarly, the pixelwise illuminant estimate at a scale σ can be viewed as the expected value of the illuminants across scales, *i.e.*

$$10 \quad L_\sigma(u) = \frac{\sum_{\ell_i \in \Upsilon} P(\ell_i | L_\sigma(u)) \ell_i}{\sum_{\ell_i \in \Upsilon} P(\ell_i | L_\sigma(u))}, \quad (14)$$

such processor 410 can approximate that the overall pixelwise illuminant by the geometric mean,

$$L(u) = \left[\prod_{\sigma \in \Gamma} L_\sigma(u) \right]^{-|\Gamma|}, \quad (15)$$

where Γ is, as before, the set of scales under consideration.

15 Implementation

The method may commence by processor 410 computing the scale space 700 for the input image. Processor 410 may use a Gaussian scale space over an octave with three different intermediate scales. As described above, this yields a factor graph 750 which

can be viewed as a three-level pyramid and represents the associations between the points of the scale space 700.

Once the scale space 700 is in hand, processor 410 proceeds to compute the set of prototypes Υ and to initialise the illuminant variables $L_\sigma(u)$. This is done as follows.

- 5 Processor 410 commences by computing a local estimate of the illuminant at each scale by dividing the scaled image into small patches (in one example these are 10x10 pixels at full resolution). At every patch, processor 410 applies an existing colour constancy algorithm. This can be done making use of a number of methods, such as grey world [Buchsbaum80], white patch [Land71] and the 1st and 2nd order grey edge methods
10 [Weijer07].

- As a result, the initial estimate of $L_\sigma(u)$ may be uniform across those pixels in the corresponding patch. It is worth noting, however, that the choice of patch size is motivated by computational complexity. Nonetheless, processor 410 may compute a pixel-wise estimate by taking neighbourhoods about each pixel under consideration at
15 each scale.

- To compute the set Υ , processor 410 may apply the mode seeking method in [Vedaldi08] to the initial estimates of $L_\sigma(u)$. This follows the intuition that the prototypes ℓ_i are expected to correspond to the modes of the distribution of initial estimates across the scale space of the image. This also has the advantage that the
20 cardinality of Υ , *i.e.* the number of prototypes ℓ_i , is data driven and determined by the distribution of $L_\sigma(u)$. It is worth noting in passing that other methods, such as k-means or maximum likelihood estimation (MLE) can be equally used to compute these estimates.

- Once the initialisation of $L_\sigma(u)$ is effected and the set of prototypes Υ is stored on
25 data memory 408(b), processor 410 proceeds to compute the graph using a Delaunay triangulation across the three-level pyramid comprised by the image lattice at the three

different scales under consideration. With the factor graph in data memory 408(b), processor 410 applies the message passing algorithm described above so as to solve the MAP problem in Equation (4).

Finally, processor 410 uses the posterior probability $P(\ell_i | I(u))$ to compute the pixel-wise illuminant using Equations (13) and (14) by setting $\frac{P(\ell_i)}{P(u)}$ as a constant. This follows the intuition that the probability of each prototype and pixel-site are uniform across the image, *i.e.* each prototype and pixel in the image are equally likely. With the posteriors in hand, the pixelwise illuminant is given by Equation (15).

Experiments

Figs. 9a and 9b show example images that may be processed using method 500.

For performing method 500 processor 410 can employ a variety of existing colour constancy methods. Prominent examples are the White Patch (WP) [Land71], Grey World (GW) [Buchsbaum80] and first and second order Grey Edge (1st GE and 2nd GE) [39] algorithms.

In Figs. 10a to 10c and 11a to 11c, we show two input scenes, the illuminant map recovered by our method and the colour corrected image, which can be computed using Equation (7). Note that the colour corrected images in Figs. 10c and 11c are devoid of object chromaticity changes induced by the illuminant in the input images.

For instance, in Fig. 10a the background is dominated by skylight whereas the foreground is being lit by a tungsten lamp. In Fig. 10 c a "warm" feeling of the tungsten upon the tower and a "cool" feeling of the skylight have both been removed. In a similar manner, changes in hue induced by the scattering underwater on the image in Fig. 11a has been removed via colour correction using the illuminant map in Fig. 11b. The illuminant maps are in good accordance with the per pixel changes expected from the top row imagery.

For the above calculation, processor 410 may obtain different scales by smoothing the image data using the Gaussian kernel

$$G(u; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u^2}{2\sigma^2}}, \quad (16)$$

where the variance σ represents the scale.

- 5 In one exmple, processor 410 processes multiple initialisation spectra for a single point leading to multiple subgraphs. In particular, processor 410 processes the pixelwise illuminant estimates $L_{\sigma,k}(u)$ at each of these subgraphs as variables, *i.e.*

$\mathcal{V} = \bigcup_{u,\sigma,k} L_{\sigma,k}(u)$. The factors, on the other hand, can be processed as the potential functions $\psi(L_c)$ which define the relationship between a subset of connected variables, *i.e.* over a clique c . Note that in coding theory and low-density parity-check codes these potentials are multivariate functions of Fourier transforms. Here, the factorisations are probabilities governed by the image data and the illuminant prototype set Υ_k .

10

Fig. 12 illustrates a diagram 1200 of the factor graph used here showing the relationships between subgraphs and pixels at different scales. From a probabilistic standpoint, processor 410 may process the interrelated variables of the graphical structure in Fig. 12 as conditional distributions within and across scales of each subgraph. In this setting, if \mathcal{G}_k denotes a factor subgraph for the k^{th} colour constancy method employed at initalisation, the complete graph \mathcal{G} can be fully written as the union over the set of subgraphs \mathcal{K} given by

20

$$\mathcal{G} = \bigcup_{\mathcal{G}_k \in \mathcal{K}} \mathcal{G}_k. \quad (17)$$

This is important since the probability of the pixelwise illuminants across the image L given the graph \mathcal{G} can be viewed as the product of the potentials over the clique set C

of the graph. This, in turn, can be expressed in terms of the set of subgraphs \mathcal{K} and the pixelwise illuminant $L_k(u)$ for the k^{th} subgraph as follows

$$\begin{aligned}
 P(L|\mathcal{G}) &= \frac{1}{Z} \prod_{c \in C} \psi(L_c; \bigcup_{\mathcal{G}_k \in \mathcal{K}} \mathcal{G}_k) \\
 &= \frac{1}{Z} \prod_{\mathcal{G}_k \in \mathcal{K}} \prod_{u, v \in \mathcal{G}_k, u \sim v} P(I(u), I(v) | L_k(u), L_k(v)) \\
 &\quad \prod_{\sigma \in \Gamma} \prod_{\ell_i \in Y_k} P(L_{\sigma, k}(u) | \ell_i), \tag{18}
 \end{aligned}$$

- 5 where Γ is the set of scales under consideration, C is the clique-set of all factors in the graph, $u \sim v$ means that pixels u and v are neighbouring each other, $\psi(L_c; \mathcal{G}_k)$ is the potential function over connected variables in the clique $c \in C$ and Z is a partition function.

It is worth mentioning that the relationship between subgraphs can be captured using a
 10 variety of distributions for the probability $P(L_k(u) | L_{k-1}(u))$. Here, as mentioned earlier, we have assumed a one-to-one correspondence across subgraphs and, for the sake of simplicity, employ a uniform distribution for $P(L_k(u) | L_{k-1}(u))$. This, in effect, permits us to remove $P(L_k(u) | L_{k-1}(u))$ from further consideration and, hence, we focus our attention on the Bayesian formulation in Equation (17). This allows a
 15 probability factorisation over the graph. This is important since the inference process can now be posed as *maximum a posteriori* (MAP) inference over the subgraphs. The objective hence becomes, to compute marginal probabilities $P(L_{\sigma, k}(u))$ for the variable set in the graph. This can be expressed in an optimisation setting as follows

$$\underset{L}{\operatorname{argmax}} P(L | \mathcal{G}_k)$$

$$= \operatorname{argmax}_L \left\{ \prod_{\mathcal{G}_k \in \mathcal{K}_u, v \in \mathcal{G}_k} P(I(u), I(v) | L_k(u), L_k(v)) \prod_{\sigma \in \Gamma} \prod_{\ell_i \in Y_k} P(L_{\sigma,k}(u) | \ell_i) \right\} \quad (19)$$

where the partition function Z has been removed from further consideration since it is a constant that does not affect the maximisation above.

The following documents are incorporated herein by reference:

- 5 [Brun00] L. Brun and Walter Kropatsch. The Construction of Pyramids with Combinatorial Maps. Technical report, 63, Institute of Computer Aided Design, Vienna University of Technology, Istr. 3/1832,A-1040 Vienna AUSTRIA, 2000.

[Barber12] Barber, David. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA, 2012.

- 10 [Buchsbaum80] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1):1--26, 1980.

[Dechter99] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41-85, 1999.

- 15 [Horn86] B. K. P. Horn and M. J. Brooks. The Variational Approach to Shape from Shading. *CVGIP*, 33(2):174-208, 1986.

[Judea88] Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

- 20 [Kimmel03] R. Kimmel and M. Elad and D. Shaked and R. Keshet and I. Sobel. A Variational Framework for Retinex. *Int. Journal of Computer Vision*, 52(1):7-23, 2003.

- [Koller09] Koller, Daphne and Friedman, Nir. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
- [Kschischang01] Frank Kschischang and Brendan J. Frey and Hans-andrea Loeliger.
5 Factor Graphs and the Sum-Product Algorithm. *IEEE Trans. Inf. Theory*, 47:498--519, 2001.
- [Land71] Edwin H. Land and John J. McCann. Lightness and Retinex Theory. *J. Opt. Soc. Am.*, 61(1):1--11, 1971.
- [Zhang94] Zhang, Nevin L and Poole, David. A simple approach to Bayesian network
10 computations. *Proc. of the Tenth Canadian Conf. on Artificial Intelligence*, 1994.
- [Vedaldi08] A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. *European Conference on Computer Vision*, pages 705-718, 2008.
- [Watson81] D. F. Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167--172, 1981.
- 15 [Weijer07] van de Weijer, J. and Gevers, T. and Gijzenij, A. Edge-Based Color Constancy. *Image Processing, IEEE Transactions on*, 16(9):2207-2214, 2007.

CLAIMS:

1. A method for determining an illumination spectrum for a first point of image data, the method comprising:
 - 5 determining the multiple instances of the image data by refining the amount of information associated with each of the multiple instances, wherein refining the amount of information comprises subsampling the image data; and
 - determining an optimised illumination spectrum for the first point based on an initial illumination spectrum and based on associations between the first point and
10 multiple second points of the multiple instances of the image data, wherein each of the multiple instances is associated with the refined amount of image information.
2. The method of claim 1, further comprising determining the initial illumination spectrum for the first point and multiple initial illumination spectra for each of the
15 multiple second points based on multiple prototypes of illuminant spectra.
3. The method of claim 2, further comprising determining the multiple prototypes of illuminant spectra based on the multiple instances of the image data.
- 20 4. The method of claim 3, wherein determining the multiple prototypes comprises determining the multiple prototypes based on regions of the image data and the multiple instances of the image data.
5. The method of any one of the preceding claims, further comprising determining
25 the associations between the first point and the multiple second points.
6. The method of claim 5, wherein determining the associations comprises creating a relational structure between the multiple instances.
- 30 7. The method of claim 6, wherein creating the relational structure comprises performing a triangulation.
8. The method of any one of the preceding claims, wherein refining the amount of information comprises applying a filtering operation to the image data.

9. The method of any one of the preceding claims, wherein refining the amount of information comprises augmenting the image data.
10. The method of any one of the preceding claims, wherein determining the optimised illumination spectrum comprises performing message passing over the associations.
11. The method of any one of the preceding claims, wherein determining the optimised illumination spectrum comprises performing a statistical estimation of the illumination spectrum for the first point of the image data.
12. The method of any one of the preceding claims, wherein the image data is one of:
hyperspectral image data; and
trichromatic image data.
13. The method of any one of the preceding claims, further comprising performing white balance adjustment based on the illumination spectrum.
14. The method of any one of the preceding claims, further comprising determining multiple initial illumination spectra for the first point, wherein determining the optimised illumination spectrum is based on the multiple initial illumination spectra for the first point.
15. The method of claim 14, wherein each of the multiple illumination spectra is associated with a respective subgraph.
16. Software that, when installed on a computer, causes the computer to perform the method of any one of the preceding claims.
17. The method of any one of the preceding claims, wherein the multiple instances form a scale space and a factor graph characterises the associations between the first point and the multiple second points of multiple instances of the scale space of the image data.

18. The method of any one of the preceding claims, wherein determining the optimised illumination spectrum comprises:

constructing the factor graph making use of the scale space and the multiple prototypes of illuminant spectra; and

5 performing message passing to compute an exact probability marginal of variables in the factor graph.

19. A computer system for determining an illumination spectrum for a first point of image data, the computer system comprising:

10 an input port to receive the image data; and

a processor to

determine the multiple instances of the image data by refining the amount of information associated with each of the multiple instances, wherein refining the amount of information comprises subsampling the image data; and

15 determine an optimised illumination spectrum for the first point based on an initial illumination spectrum and based on associations between the first point and multiple second points of the multiple instances of the image data, wherein each of the multiple instances is associated with the refined amount of image information.

20

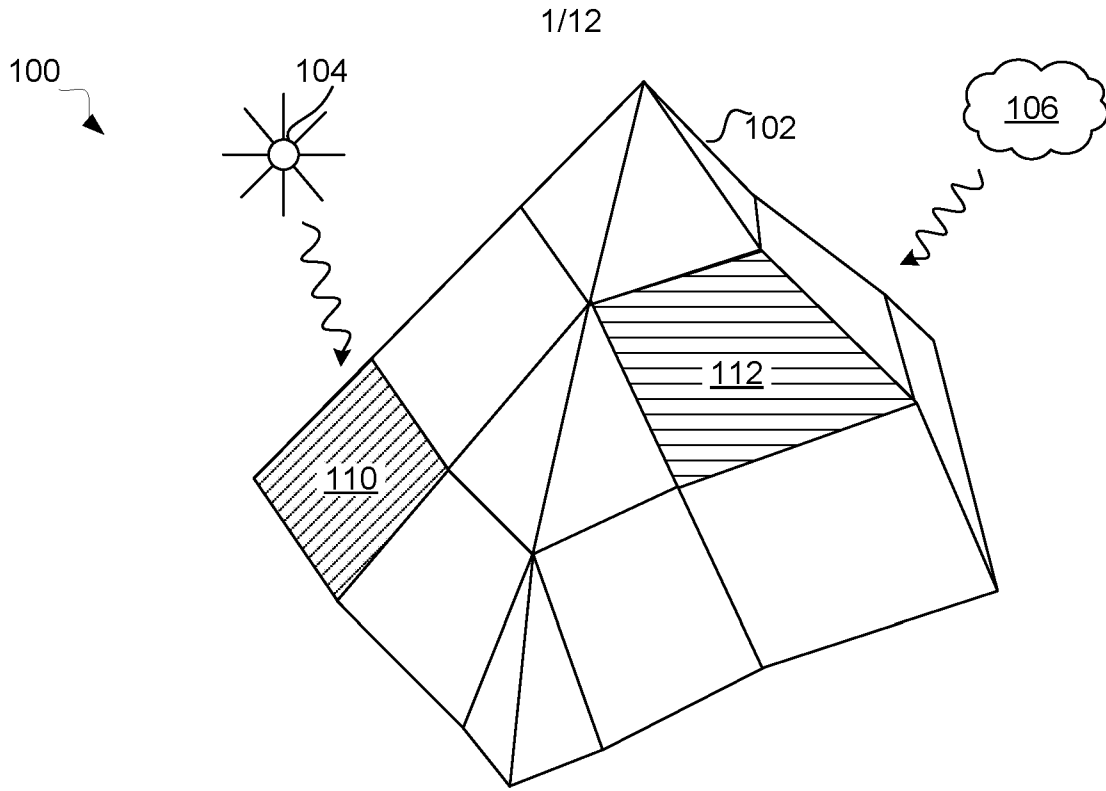


Fig. 1

PRIOR ART

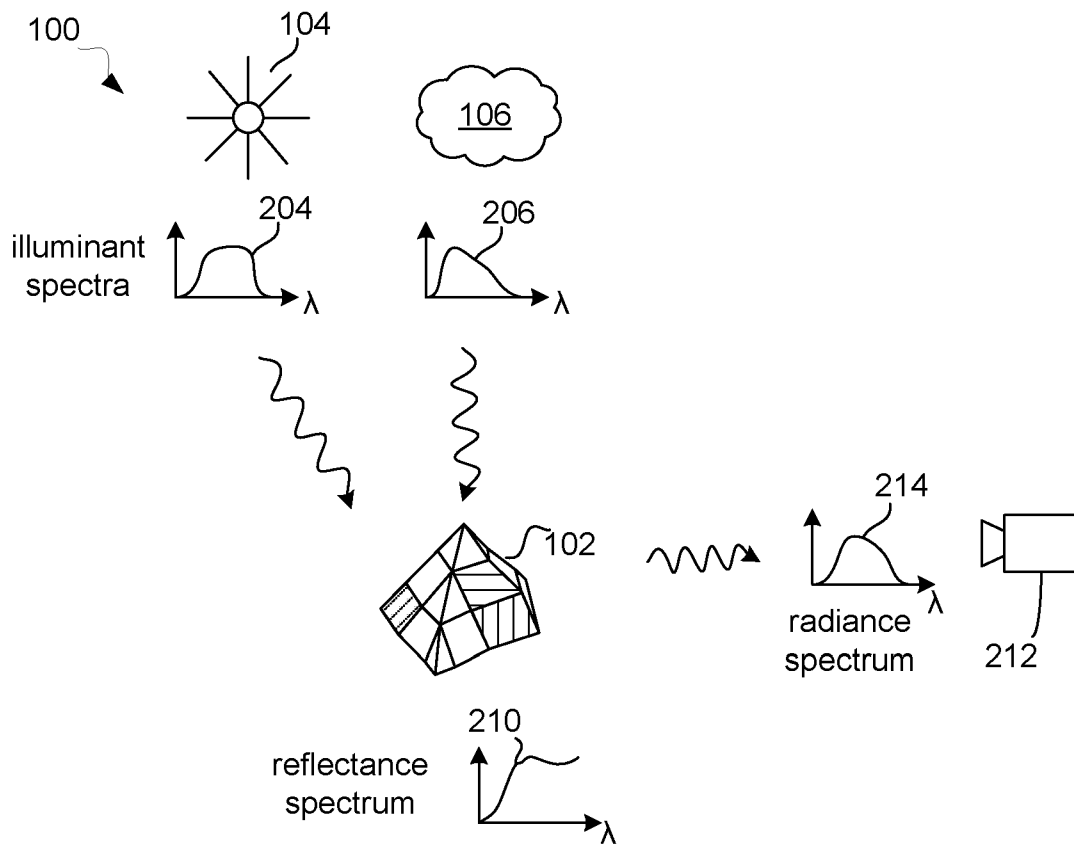


Fig. 2

PRIOR ART

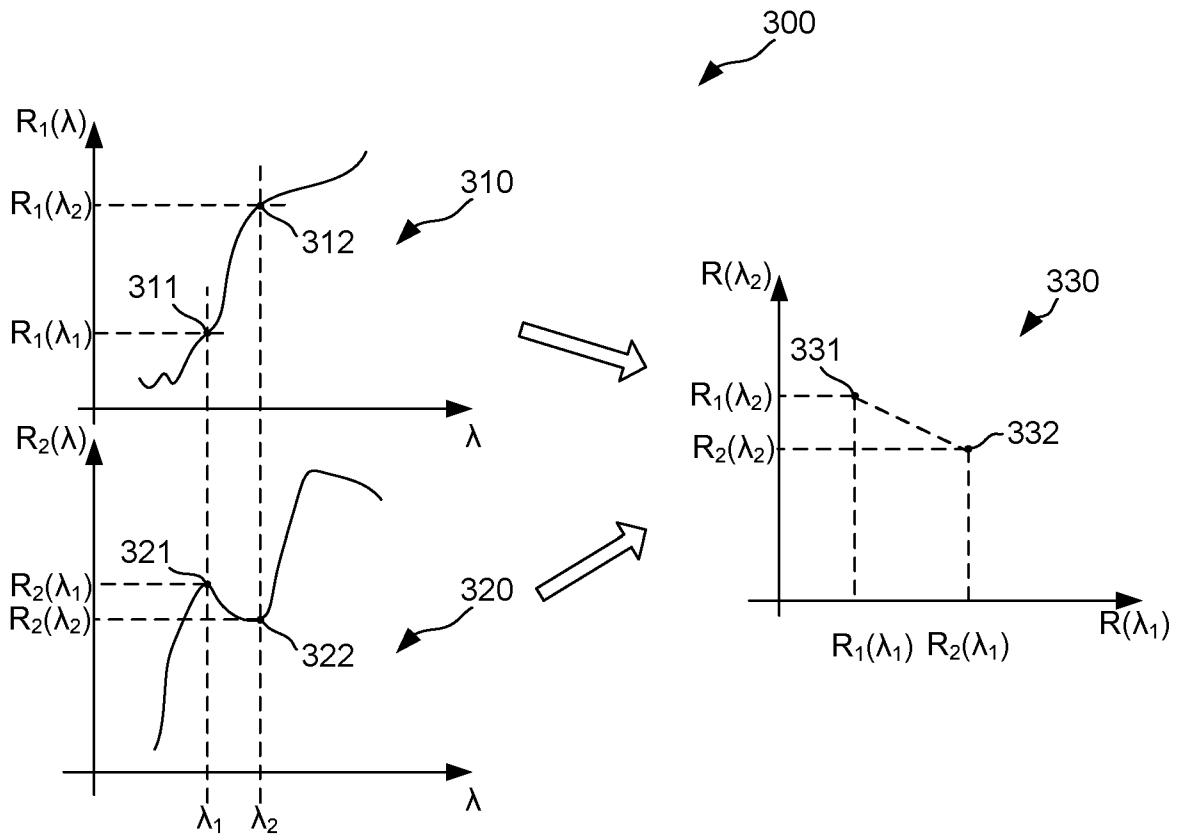


Fig. 3

PRIOR ART

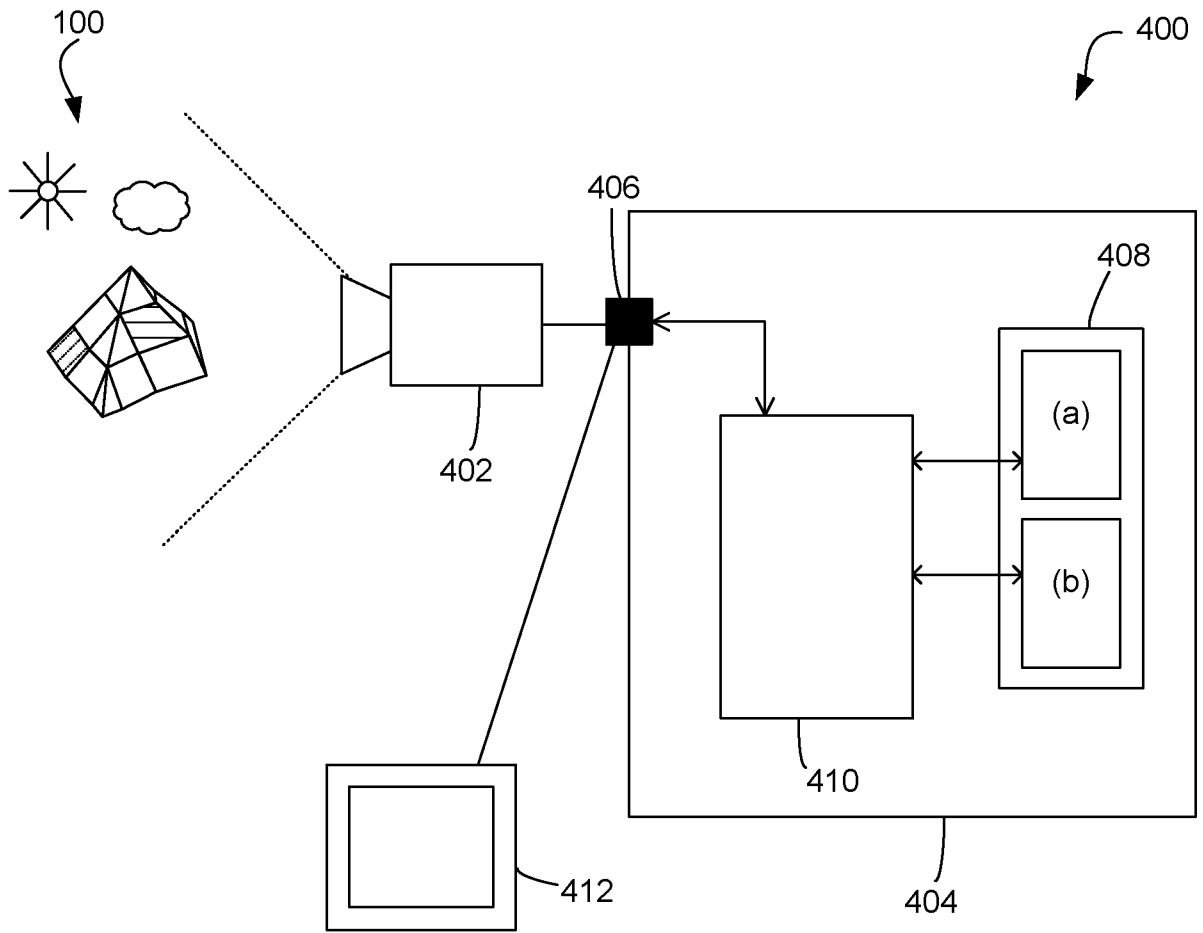


Fig. 4

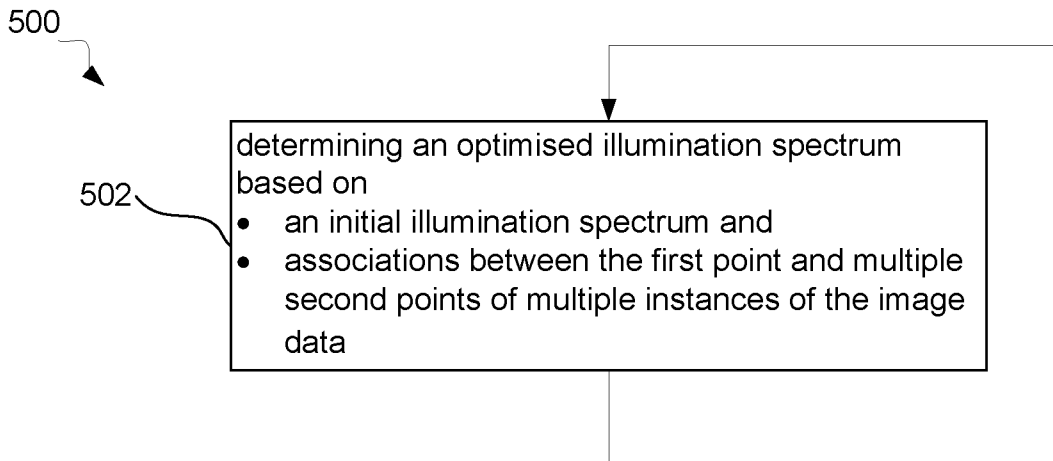


Fig. 5

5/12

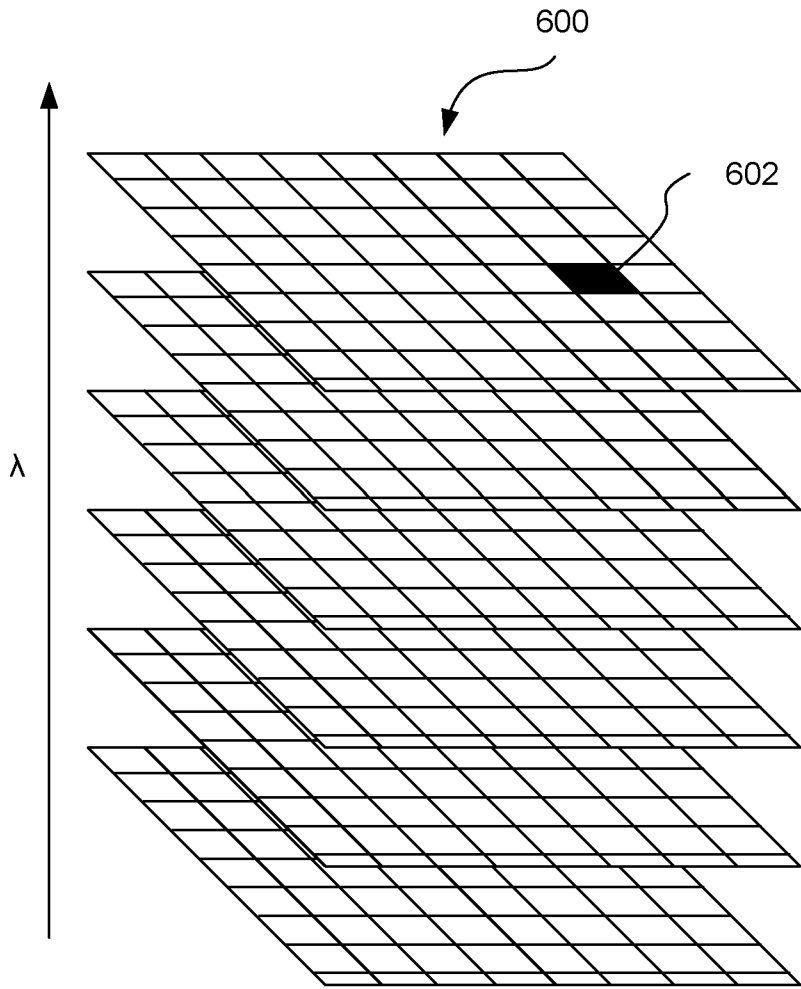
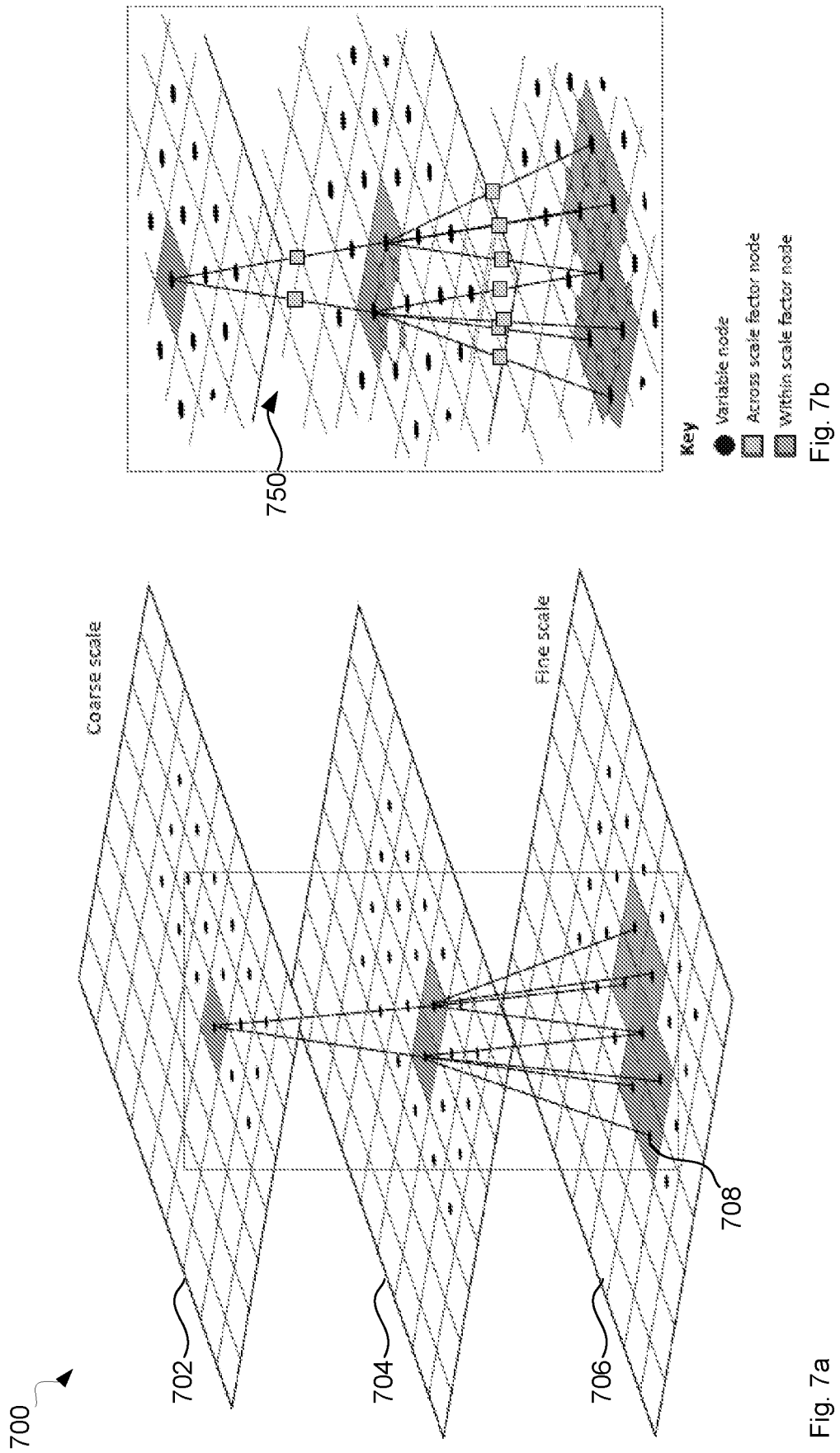


Fig. 6



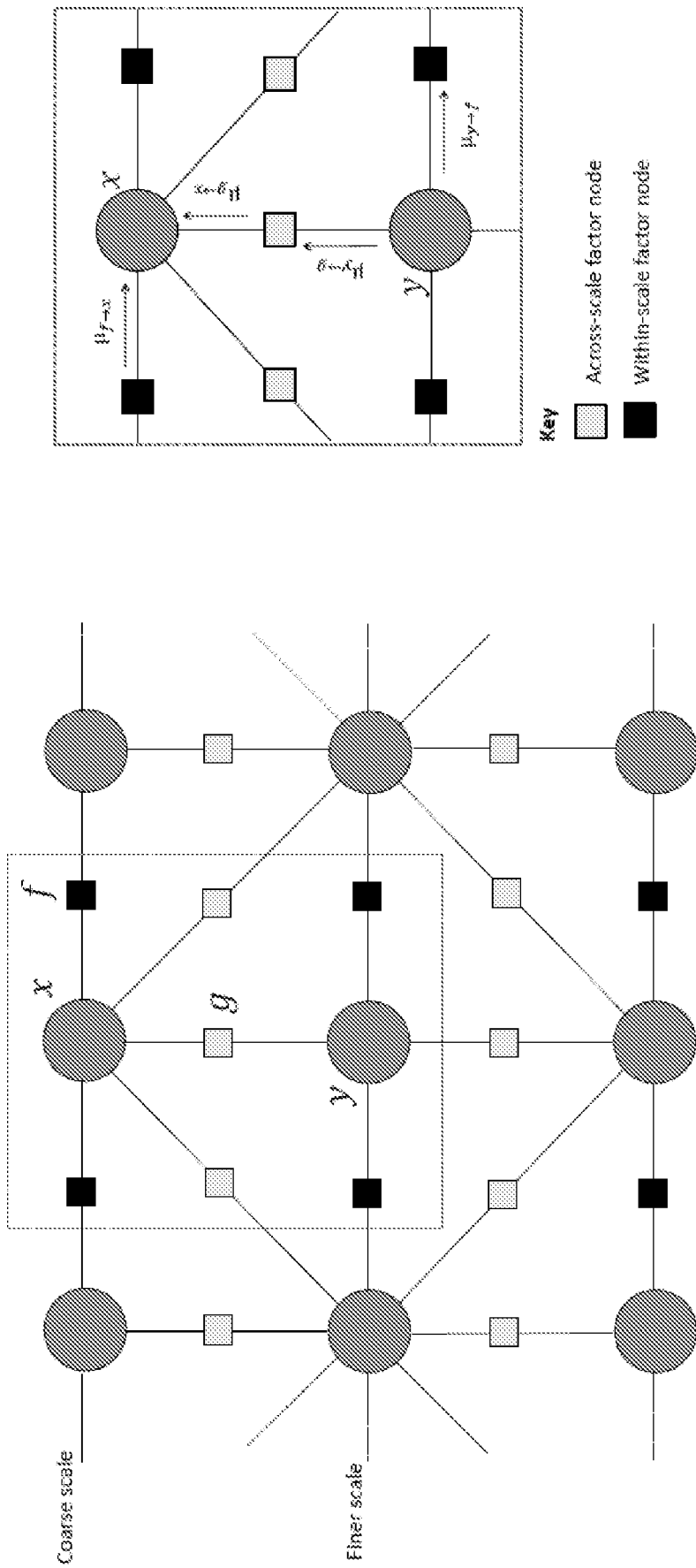


Fig. 8a

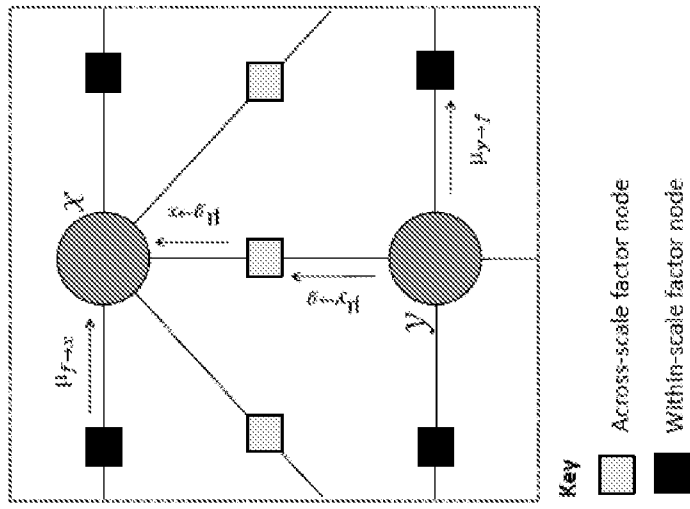


Fig. 8b



Fig. 9a



Fig. 9b



Fig. 10a



Fig. 11a



Fig. 10b



Fig. 11b

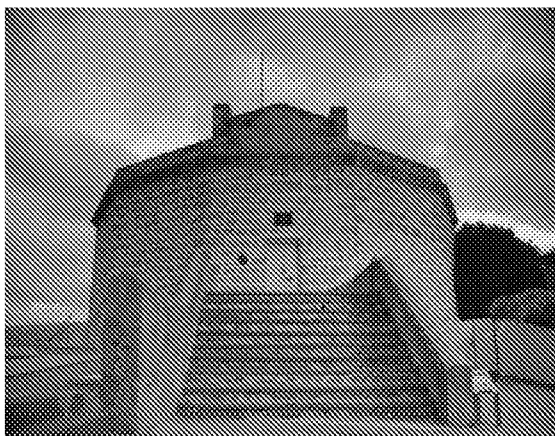


Fig. 10c



Fig. 11c

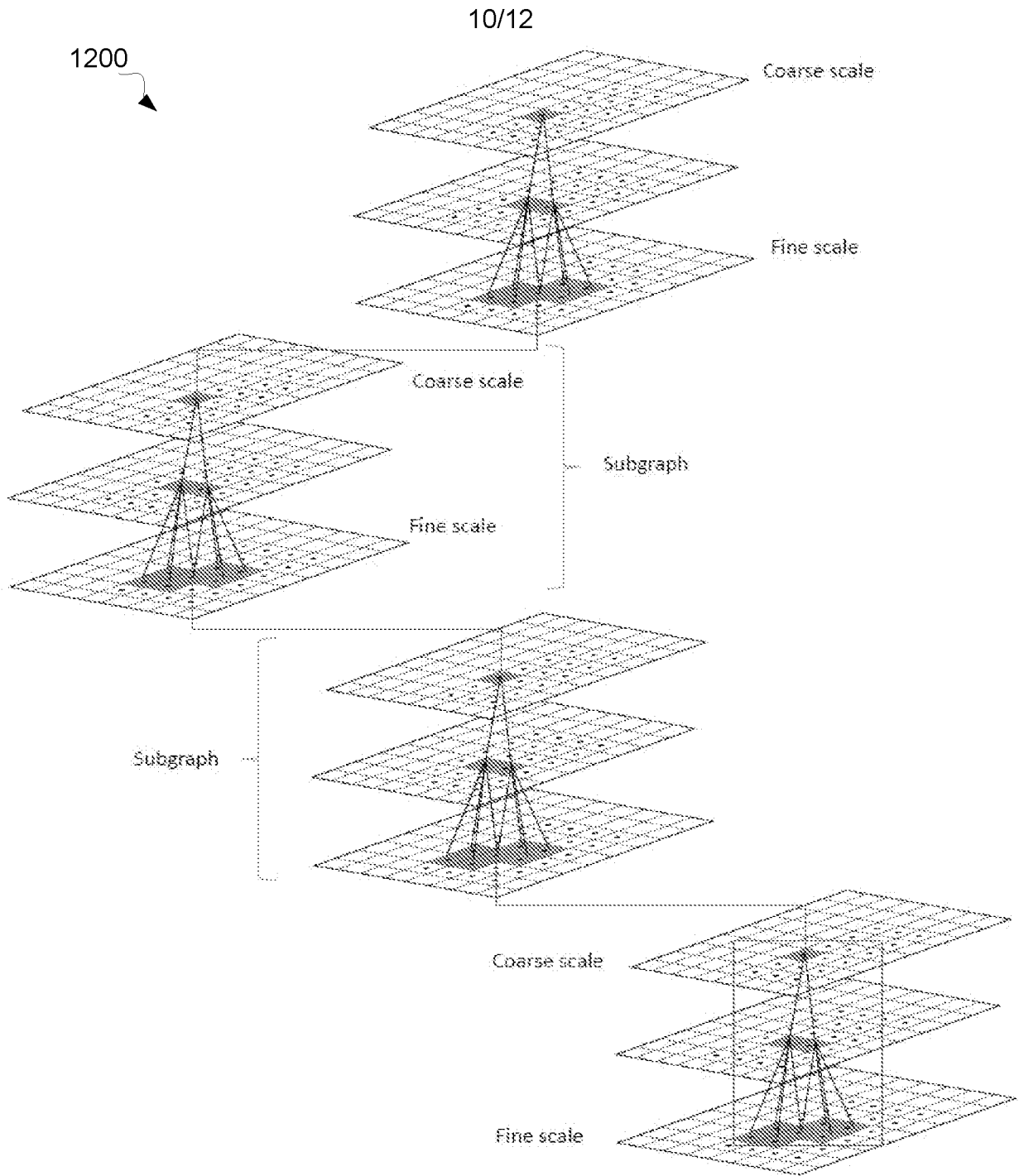


Fig. 12

Algorithm 1 - Pixelwise Illuminant Recovery

Require: $\mathbf{I}(u)$: image data; $\sigma \in \Gamma$: set of Gaussian scales; b : bandwidth parameter for the quick-shift method [Ve ; τ dald
kernel cut-off variable i08]

Ensure: Illuminant estimate $L(u)$ i08]

- 1: **for all** $k \in \mathcal{K}$ **do**
- 2: **for all** $\sigma \in \Gamma$ **do**
- 3: Initialise illuminant estimate according to Algorithm 2 at scale σ
- 4: **end for**
- 5: Construct graph \mathcal{G}_k corresponding to the k^{th} colour constancy method under consideration
- 6: Triangulate graph \mathcal{G}_k using the convex hull technique in [Zhang94]
- 7: **for all** Edges $e \in \mathcal{E}$ connecting variables in \mathcal{V} and factors in \mathcal{F} **do**
- 8: Perform message passing using Equations (12) and (11)
- 9: **end for**
- 10: Find $L_{\sigma,k}(u)$ using the probabilities $P(\ell_i | L_{\sigma,k}(u))$ at the finest scale as shown in Equation (14).
- 11: **end for**
- 12: **return** $L(u)$ given by the geometric mean in Equation (15).

Fig. 13

Algorithm 2 - Compute illuminant prototypes ℓ_i , $\forall k$ and initial illuminant estimates

Require: $\mathbf{I}(u)$: input image; $k \in \mathcal{K}$: illuminant yielded by the \mathcal{K} methods under consideration; $\sigma \in \Gamma$: set of Gaussian scales; b : bandwidth parameter for the method in [Vedald

Ensure: Illuminant prototypes $\forall k, \ell_i \in \Upsilon_k$ i08]

```

1: for all  $k \in \mathcal{K}$  do
2:   for all  $\sigma \in \Gamma$  do
3:     Filter image using the Gaussian kernel in Equation
       (16) at scale  $\sigma$ ,  $\mathcal{I}_\sigma \leftarrow \mathcal{I}$ 
4:     Partition  $\mathcal{I}_\sigma$  into patches,  $\bigcup_m \mathcal{I}_{\sigma,m} \leftarrow \mathcal{I}_\sigma$ 
5:     for all  $\mathcal{I}_{\sigma,m} \in \mathcal{I}_\sigma$  do
6:       Compute uniform illuminant using the  $k^{th}$  algo-
         rithm in  $\mathcal{K}$ 
7:     end for
8:      $L_{\sigma,k}^{(0)} \leftarrow \mathcal{I}_\sigma$ 
9:   end for
10:  Apply the method in [Vedald] to all  $\mathcal{I}_\sigma$  to find  $\ell_i \in \Upsilon_k$ 
11: end for
12: return  $\bigcup_k \{\Upsilon_k\}, \bigcup_{\sigma,k} \{L_{\sigma,k}^{(0)}\}$ 

```

Fig. 14